IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 26, 2018, accepted December 19, 2018, date of publication December 28, 2018,
date of current version January 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2890096

An Encoding Scheme Capturing Generic Priors and Properties of Amino Acids Improves Protein Classification

XINRUI ZHOU[1], RUI YIN[1], JIE ZHENG[2], AND CHEE-KEONG KWOH[1]
[1]School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798
[2]School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China

Corresponding author: Chee-Keong Kwoh (asckkwoh@ntu.edu.sg)

This work was supported by the Singapore Ministry of Education through AcRF Tier 2 under Grant MOE2014-T2-2-023 and through RG21/15 Tier 1 under Grant 2015-T1-001-169-11.

ABSTRACT Feature engineering aims at representing non-numeric data with numeric features that keep the essential information of the underlying problem, and it is a non-trivial process in building a predictive model. In bioinformatics, there is a profound scale of DNA and protein sequences available, but far from being fully utilized. Computational models can facilitate the analyses of large-scale data. However, most computational models require a numeric representation as input. Expert knowledge can help design features to cast the raw symbolic data effectively. But generally, the features vary from case to case and have to be redesigned for a problem. Automated feature engineering, i.e., an encoding scheme automating the construction of features, saves the redesigning process and allows the researchers to try different representations with minimal effort. This is more in line with the explosion of data and the goal of building an intelligent system. In this paper, we introduce an encoding scheme for protein sequences, which encodes the representative sequence dataset into a numeric matrix that can be fed into a downstream learning model. The method, Context-Free Encoding Scheme (CFreeEnS), was proposed for a dataset with labels for pairwise sequences. Here, we improve the method by making it applicable to a batch of protein sequences, requiring no sequence alignment beforehand. The improved method is applied to protein classification at the functional level, including identifying antimicrobial peptides, screening tumor homing peptides, and detecting hemolytic peptides and phage virion proteins. Compared with the traditional methods using task-specific designed features, CFreeEnS improves the predicting accuracy, with an increase ranging from 5.54% to 14.14%. The results indicate that the improved CFreeEnS, free from dependence on carefully designed features, is promising in capturing generic priors and essential properties of amino acids, thereby serving as an automated feature engineering method for protein sequences.

INDEX TERMS Encoding scheme, feature engineering, information representation, machine learning.

I. INTRODUCTION

Representing non-numeric raw data with numeric features that profile raw data from different angles, namely feature engineering, is generally the first process in a machine learning pipeline. Most machine learning, especially deep learning, algorithms require a numeric data representation with equal length as the input [1]. The quality of data representation can profoundly affect the performance of the downstream learning methods. Therefore, studies in many fields (e.g., speech recognition, text mining, bioinformatics, etc.) have endeavored to design an effective data representation supporting and improving the subsequent learning process [2]–[5]. Unfortunately, features usually vary from problems, especially when expert knowledge is involved in the design, making them only useful in the context of specific tasks and models [6]. In the era of big data, automated feature engineering that is less sensitive to the context is more desired. Studies have justified that knowledge learned from one task can be applied to another via transfer learning [7], [8]. A good representation that can benefit large-scale learning should maintain the intrinsic structure of data, be task non-specific but keep the most relevant information about the task at hand [1].

In bioinformatics, the high-throughput sequencing techniques have made scads of DNA and protein sequences available, launching the field into a new era of big data. However,

deciphering the sequences, e.g. to uncover the relationship between genotype and phenotype, remains a challenge. As a complement to the costly wet-lab experiments, computational models provide an alternative perspective to disentangle the hidden patterns in protein sequences. Casting the symbolic sequence dataset into a numeric representation usually serves as the upstream stage in a pipeline for analyses. Expert knowledge about the problem may facilitate the analyses, but this process is also tedious and limited by human subjectivity [9]. An encoding scheme of protein sequences capturing generic priors of amino acids is more likely to be free from the context (i.e., the task, data, and model) so that the bioinformaticians can save the effort of designing features for different problems. This is more in line with the eruption of sequence data and the goal of automating sequence annotation.

Protein classification is a fundamental problem in analyzing the protein sequences, referring to multifaceted tasks. Proteins can be classified regarding different aspects (e.g., family, structure, localization, function, protein-protein interaction, etc.) and different levels of a classification hierarchy (e.g., subfamilies, families, superfamilies, etc.) [10]. Classifying a protein sequence into a well-characterized group is a preliminary but non-trivial analysis, helpful for annotating its properties. For characterizing more specific functions or phenotype, usually experimental assays are designed and conducted to test the properties of a targeted entity [11]. For example, the hemagglutination inhibition (HI) assay is designed to quantify the antigenic similarity between the hemagglutinin proteins [12].

Traditional methods for protein classification include the composition-based methods (e.g., the amino acid composition, pseudo-amino acid composition, atomic composition, etc.) [13]–[15] and motif-based methods (e.g., *n-grams*, active motifs, conserved motifs, etc.) [16]–[18]. Although they have been applied to many classification tasks and yielded moderate accuracy of 75%–85%, generally serving as benchmarks for task-specific methods, there is still room to improve the performance. Recently, there have been many representations of biological sequences inspired by natural language processing (NLP), treating the sequences from a text mining perspective. Asgari and Mofrad [19] proposed *ProtVec*, a continuously distributed representation for protein sequences using the *n-grams* with a *skip-gram* model. Islam *et al.* [20] extended their work by a modified *n-gram* and *skip-gram* model, named *m-NGSG*, where the optimal parameters are obtained through grid search. It is a state-of-the-art automated feature generation method for protein sequences that has been applied to protein classification, promising to accelerate large-scale characterization of protein sequences. However, the methods do not give a solution to lower levels of the classification hierarchy and distinguishing protein sequences that are within the same family but bearing a few phenotype-related mutations.

Previously, we proposed an encoding scheme for protein sequence pairs, named *CFreeEnS*, to predict the antigenic similarity between the hemagglutinin proteins of influenza viruses, which was effective across multiple subtypes of influenza [21]. We hypothesized that the method captured intrinsic distinctions between amino acid pairs and was promising to be applied to other problems with aligned protein sequence pairs as the input. It was only applicable to a dataset with labels for pairwise sequences, measuring the phenotype distinctions. In this paper, we extended the encoding scheme *CFreeEnS* to make it applicable to a batch of protein sequences where the labels annotate each sequence instead of measuring pairwise distinctions. The improved *CFreeEnS* requires no sequence alignment beforehand. Each protein sequence is profiled by the average value of known amino acid properties. This encoding scheme keeps conserved known properties of amino acids as much as possible, thereby allowing the downstream learning method to disentangle features relevant to the target task. Together with the module dealing with pairwise protein sequences, the improved *CFreeEnS* is able to profile any protein sequence dataset with a numeric representation, maintaining generic properties of amino acid, and the distinctions between pairs of amino acids. With the two modules, *CFreeEnS* is more matched with the name "a **C**ontext-**Free** **E**ncoding **S**cheme for protein sequences". Thus, the manuscript would present *CFreeEnS* as an independent encoding scheme with two modules, taking protein sequences and sequence pairs as the input respectively, followed by several cases of application. The manuscript is structured as follows:

1) Section II describes how *CFreeEnS* encodes the protein sequences and protein sequences pairs. The module dealing with protein sequence pairs has been presented in our previous work [21] so that we would only briefly introduce the framework.
2) Section III presents the applications of *CFreeEnS*. Regarding the module of encoding protein sequences, *CFreeEnS* is applied to several protein classification problems, including identifying the antimicrobial peptides, tumor homing peptides, hemolytic peptides, and phage virion proteins. In terms of encoding protein sequence pairs, *CFreeEnS* is used to predict the antigenic similarity between the hemagglutinin protein of influenza viruses. The results of *CFreeEnS* on the mentioned cases are compared with traditional methods using designed features for each case, as well as the state-of-the-art methods that are less specific to the task.
3) Section IV discusses the strength and weakness of *CFreeEnS* and other methods for protein representations.

II. METHODS

A typical pipeline of computational modeling consists of four modules, as shown in Figure 1. The pipeline begins with a data retrieval module and followed by an iterative process including feature engineering, modeling, and evaluation. With a satisfying model performance, the trained model is promising to be deployed for applications. In the

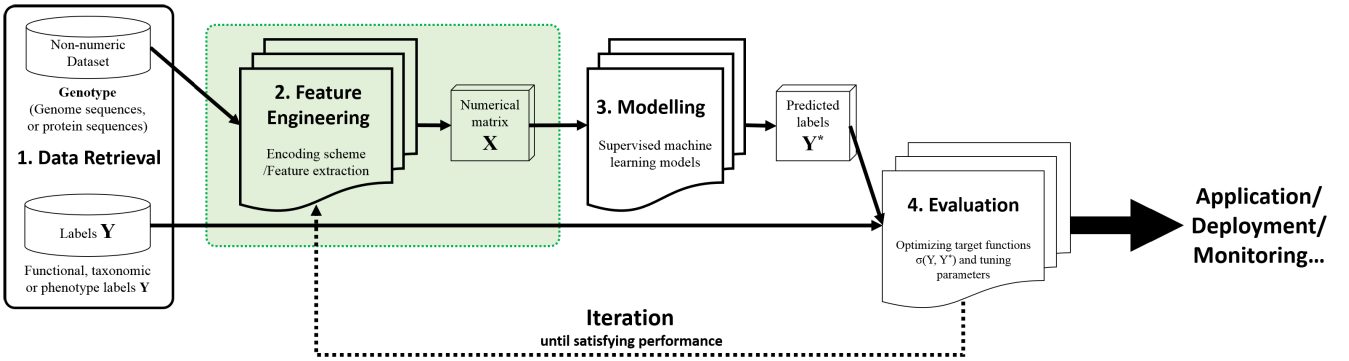

FIGURE 1. A typical pipeline of supervised machine learning models in bioinformatics. 1. Data retrieval. Preparing the genotype dataset and the corresponding labels. 2. Feature engineering. Representing the non-numeric raw dataset with numeric features that can be fed into the downstream modeling module. As CFreeEnS contributes to this module, it is highlighted with green shadow. 3. Modelling. Using supervised learning algorithms to predict the labels. 4. Evaluation. Comparing the predicted labels with the true labels to measure the performance of the model, typically by optimizing an error function $\sigma(Y, Y^*)$. Iterating the process from feature engineering to evaluation and tuning parameters if necessary until the model achieves a satisfying performance, and then the model is promising to be deployed for applications.

feature engineering module, the raw non-numeric dataset is encoded by a numeric matrix so that it can be fed into the downstream learning algorithms, i.e. the modeling module. The performance of a computational model mainly relies on the cooperation of the two parts, i.e. the upstream encoding scheme and the downstream learning algorithm. The effectiveness of a learning algorithm is largely dependent on the quality of the input, which is the data representation cast by an upstream encoding scheme. Different representations can entangle and hide variant explanatory factors of the data.

As for the application in bioinformatics, usually, the dataset includes genotype information represented by symbolic genome sequences or protein sequences, and phenotype labels about the function or taxonomic name, denoted as Y. However, in the modeling part, most downstream learning algorithms need an input of numeric vectors with equal-length. Thus, an encoding scheme is needed to cast the non-numeric sequence dataset into a numeric matrix X, which can be fed into a downstream supervised learning model to predict the target labels. The predictions are denoted as Y^*. In the evaluation module, the true labels Y and the predicted labels Y^* are compared to measure the performance of the computational model.

To measure the effectiveness of encoding schemes, one way is to compare the overall performances using the same downstream learning method. A good encoding scheme should return a representation keeping the most relevant information about the predicting target and the least noise, which will benefit the predicting accuracy of the downstream learning methods. Implementing expert domain knowledge into the input dataset usually would help improve the design of a suitable encoding scheme, but an encoding scheme with more generic priors instead is more in line with the goal of automating data-driven learning.

Herein, we improved our proposed method in [21] so that the encoding scheme can be applied to both protein sequences with varying lengths and protein sequence pairs, which covers the most situation of sequence analyses in bioinformatics. The method, *CFreeEnS*, is based on the AAindex database [22], which is the collection of amino acid indexes and mutation matrices from published work, representing physiochemic and biochemical properties related to the specificity and diversity of protein structures and functions. Currently, the AAindex contains 566 amino acid indexes in AAindex1, 94 substitution matrices in AAindex2 and 47 matrices derived from the statistical pairwise contact potential between amino acids in AAindex3. For encoding protein sequences with varying length to roughly group the proteins, the *CFreeEnS* encodes the sequences with AAindex1. Likewise, for characterizing more subtle distinctions between proteins, the substitution matrices in AAindex2 are utilized in *CFreeEnS*.

Figure 2 presents how the improved *CFreeEnS* works. When taking a sequence batch $\boldsymbol{S}$ of m sequences with varying lengths as the input, *CFreeEnS* encodes each sequence $\boldsymbol{s_i}$ using k amino acid indexes in AAindex1, which represent generic physicochemical and biochemical properties, α-helix, β-strand and turn propensities of amino acids. For the sequence $\boldsymbol{s_i}$ encoded by index j, the outputting numeric vector is denoted as $\boldsymbol{s_i^j}$. The average value v_{ij} is calculated, representing the value of s_i with the property j. After encoded by the k indexes, the sequence s_i is represented by a vector $\boldsymbol{v_i} = [v_{i1}, v_{i2}, ...v_{ij}, ...v_{ik}]$. After stacking the vectors for m sequences, the symbolic dataset is encoded by the numeric matrix $\boldsymbol{X}$ with dimension $m \times k$, which can be fed into a downstream machine learning algorithm together with the label vector $\boldsymbol{Y}$ of length m. When analyzing the substitutions of two protein sequences, pairwise alignment is required before inputting into the encoding module. Taking a batch aligned protein sequence pairs, each sequence pair $\boldsymbol{p_i}$ of length l is encoded with k substitution matrices in AAindex2. For m sequence pairs, *CFreeEnS* outputs a numeric matrix $\boldsymbol{X}$ with dimension $m \times k \times l$.

Algorithm 1 clarifies how the *CFreeEnS* encodes a protein sequence or a pairwise protein sequence alignment using k indexes in detail. For a protein sequence s, each residue is

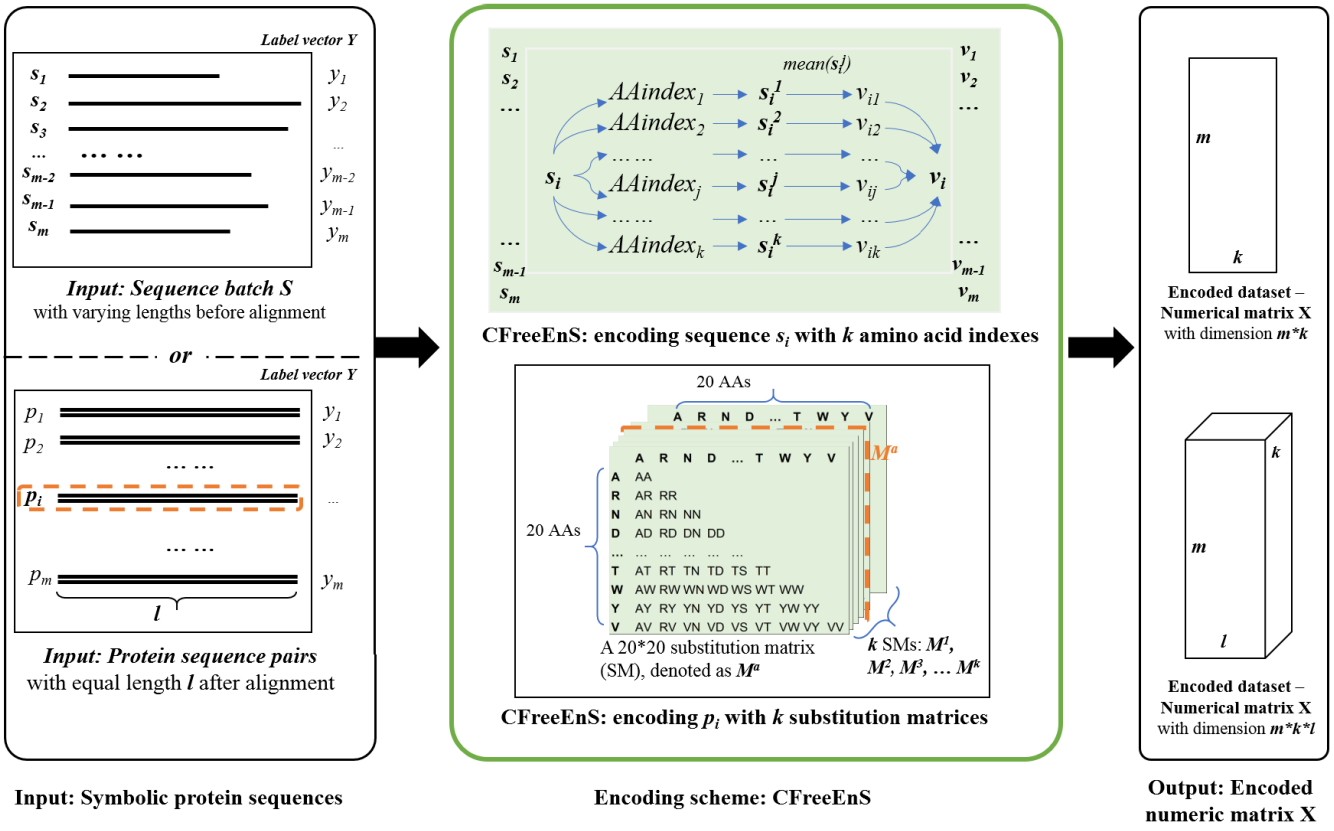

FIGURE 2. The diagram of *CFreeEnS* for m protein sequences or protein sequence pairs. For sequences with varying lengths, each sequence s_i can be casted to a numeric vector using an amino acid index $AAindex_j$ in AAindex1. The average score v_{ij} is calculated. Using k amino acid indexes, the protein sequence will be represented by a vector v_i with length k. For aligned sequence pairs with length l, each pair can be casted to a numeric vector using one amino acid substitution matrix in AAindex2. Using k substitution matrices, each sequence will be represented by a matrix with dimension $l \times k$. For m sequence pairs, *CFreeEnS* outputs a $m \times k \times l$ matrix.

replaced by the scores evaluated in the amino acid index idx. The average value v_{idx} of the encoded vector v_s is taken as the evaluation of the sequence. Using k amino acid indexes, the output is saved in a dictionary v where the idx is taken as the key for the v_{idx}. For a protein sequence pair with sequence s_1 and s_2 using a substitution matrix idx, the distance for each pairwise residue a_1 and a_2 is calculated as:

$$d(a_1, a_2) = idx(a_1, a_1) + idx(a_2, a_2) - 2 \times idx(a1, a2)$$

where the $idx(a_i, a_j)$ represents the score in substitution matrix idx for residue a_i and a_j. A penalty is λ is added for a gap. Similarly, using k substitution matrices, the distance vectors are saved in a dictionary v with idx as the keys. As mentioned, there are $k = 94$ substitution matrices in the AAindex database, with subtle distinctions between residues available [23], which provides an opportunity to systematically check all substitution scoring matrices. The most effective ones casting the dataset into different space can be selected for the scenario we need to analyze. Traversing each instance in the dataset and stacking the value vectors, as illustrated in Figure 2, *CFreeEnS* outputs a numeric matrix of dimension $m \times k$ for m protein sequences, or a matrix of dimension $m \times k \times l$ for m protein sequence pairs. In this way, *CFreeEnS* can convert symbolic protein sequences and protein sequence pairs into numeric representations that can be fed into downstream learning machine learning models.

The encoding scheme has been applied to different tasks of protein classification, as well as measuring the phenotype similarity between proteins, resulting in better performance than other traditional schemes.

III. APPLICATIONS AND RESULTS

A. PROTEIN CLASSIFICATION

To test the effectiveness of *CFreeEnS* on casting the protein sequences to numeric representations, we conducted protein classification under different scenarios. The classification results are compared with other traditional methods using handcrafted features specially designed for each dataset. Also, we compared the *CFreeEnS* with a state-of-the-art protein classification method named *m-NGSG*, which is inspired by natural language processing [20].

An overview of the datasets for protein classification is presented in Table 1. The abbreviations of datasets are identical with the methods proposed explicitly for them.

1) **iAMP.** The iAMP dataset, abbreviated for identifying antimicrobial peptides, includes antibacterial peptides,

Algorithm 1 *CFreeEnS* for a Protein Sequence or a Pairwise Sequence Alignment

function CFreeEnS(s, *idxList*) ▷ Input: s is either a protein sequence or a pairwise sequence alignment; *idxList* is a list with k index IDs
  flag = *checkType*(s) ▷ *checkType* is a function returning 0 if the input s is a protein sequence while 1 is the input is a pairwise sequence alignment.
  declare $v = \{\}$ ▷ v is a dictionary where the keys are the IDs of amino acid indexes for encoding
  if flag == 0 **then** ▷ CFreeEnS for a protein sequence
    for *idx* in *idxList* **do**
      $v_s = []$
      for j = 1 to len(s) **do**
        v_s.append(*idx*.get(s[j])) ▷ Get the score of each residue $s[j]$ from the amino acid index *idx*
      $v_{idx} = v_s$.mean()
    $v[idx] = v_{idx}$
  else ▷ CFreeEnS for a pairwise sequence alignment
    $s_1 = s[0]$; $s_2 = s[1]$
    assert len(s_1) == len(s_2)
    for *idx* in *idxList* **do**
      $v_s = []$
      for j = 1 to len(s_1) **do**
        $a_1 = s_1[j]$; $a_2 = s_2[j]$
        if a_1 == "−" or a_2 == "−" **then**
          v_s.append(λ) ▷ Add penalty for gaps in pairwise alignment
        else
          $dist = idx.\text{get}(a_1, a_1) + idx.\text{get}(a_2, a_2) - 2\times idx.\text{get}(a_1, a_2)$ ▷ Get the distance score of pairwise amino acids
          v_s.append($dist$)
      $v[idx] = v_s$
  return v

TABLE 1. An overview of datasets for protein classification.

Datasets (Methods)	Description	Sequence Lengths	# Sequences
iAMP	Antimicrobial peptides data involves anti-bacteria, anti-cancer, anti-fungal and anti-viral sequences. The task is to classify antimicrobial peptides from non-antimicrobial peptides.	10–255; Median: 26	6214
TumorHPD	TumorHPD classifies tumor homing peptides, helping to design analogs of tumor homing ability	4–31; Median: 10	2240
HemoPI	Identifying the hemolytic peptides from non-hemolytic peptides.	4–98; Median: 18	1104
PVPred	Phage virion proteins are classified from other non-phage virion proteins	23–1825; Median: 213	337

antiviral peptides, and antifungal peptides. The antimicrobial peptides are important host defense molecules in the innate immune system against pathogens. Computational identification of AMPs saves the researchers from expensive *in vitro* wet-lab experiments. Previous analyses tried incorporating several designed features, including the distribution patterns of amino acids [24], pseudo amino acid composition and some selected physiochemical features [25], [26]. The benchmark dataset of iAMP includes 3107 positive samples and an equal number of negative samples generated from UniProt. Sequence lengths of antimicrobial peptides in the dataset vary from 10 to 255, with a median of 26.

2) **TumorHPD.** The TumorHPD is a web server for recognizing tumor homing peptides, which are able to recognize tumor cells [27]. Amino acid composition profile, dipeptide composition, and binary profile are generated in the TumorHPD to capture the features of input sequences. The benchmark dataset includes 651 and 469 positive samples obtained from the TumorHoPe database as the training set and validation set respectively. An equal number of negative samples are generated from Swiss-Prot database. The median of lengths of the tumor homing peptides is 10.
3) **HemoPI.** The HemoPI, short for hemolytic peptide identification, is to screen hemolytic peptides from the non-hemolytic, where quantitative matrices are developed for measuring the hemotoxicity [28]. Motifs observed in hemolytic peptides are utilized as features to differentiate them from the non-hemolytic ones. There are 552 positive samples which are experimentally validated highly hemolytic peptides from the Hemolytik Database. The same amount of negative samples are generated from SwissProt.

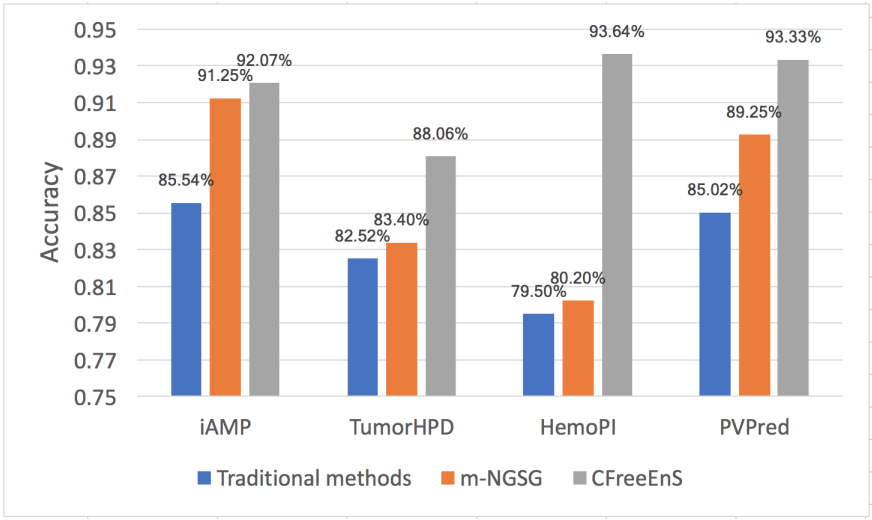

FIGURE 3. Predicting accuracy of *CFreeEnS* on protein classification compared with traditional methods and *m-NGSG*.

4) **PVPred.** PVPred predicts the phage virion proteins by analyzing the variance and optimizing the *g*-gap dipeptide [29]. Most phage virion proteins in the dataset have long primary sequences with several hundred residues. The sequence lengths vary from 23 to 1825, but with a median of 213. There are 99 positive samples and 208 negative samples in the training set; 11 positive samples and 19 negative samples in the validation set.

The four datasets are encoded by *CFreeEnS*, using all the available 566 amino acid indexes in the AAindex database. Sequences with varying length are represented by vectors with length 566. Columns with high correlation are dropped before inputting into a downstream learning method. To compare the effectiveness of data representation, we keep the same downstream learning procedure as those traditional methods using designed features for each dataset. Besides, the *m-NGSG*, a state-of-the-art method treating protein sequences as normal text and generating features from a text mining perspective, has been applied to the four datasets [20].

Table 2 shows the classification results of *CFreeEnS*, taking 0.95 as the dropout threshold. There are 190, 146, 170 and 211 features dropped for *iAMP*, *TumorHPD*, *HemoPI* and *PVPred* respectively. The performance of *CFreeEnS* on each dataset is evaluated with accuracy, precision, recall, F-score, AUC, geometric-mean (g-mean) and Matthews correlation coefficient (MCC). The *CFreeEnS* works best on the *HemoPI* database with the highest AUC (0.936) and MCC (0.874), while worst on the *TumorHPD* database with a moderate AUC of 0.881.

When comparing with other methods, as presented in Figure 3, we can observe that *CFreeEnS* outperforms the state-of-the-art method *m-NGSG* and traditional methods using designed features. The predicting accuracy scores of the four datasets are improved. The increases of accuracy

TABLE 2. The classification results of *CFreeEnS* applied to iAMP, TumorHPD, HemoPI and PVPred datasets.

Datasets	Accuracy	Precision	Recall	F-score	AUC	G-mean	MCC
iAMP	0.9207	0.9261	0.9203	0.9201	0.9203	0.9185	0.8464
TumorHPD	0.8806	0.8987	0.8806	0.8792	0.8806	0.8741	0.7791
HemoPI	0.9364	0.9377	0.9364	0.9363	0.9364	0.9360	0.8740
PVPred	0.9333	0.9397	0.9333	0.9317	0.9091	0.9045	0.8604

range from 5.54% to 14.14% when compared with the traditional method, from 0.82% to 13.44% when compared with *m-NGSG*. Although the accuracy of predicting tumor homing peptides seems not high enough, it has been improved by 5.54% compared with the traditional method using several designed profiles. Even compared with the *m-NGSG*, the accuracy has been increased by 4.66%. The fact that tumor homing peptides are generally short with a median of 10 may partially contribute to the difficulty in accurate prediction. Both *m-NGSG* and *CFreeEnS* work well on the *iAMP* dataset, which may benefit from the large amount and balanced training samples. It is worth noting that the *CFreeEnS* also works well on the *PVPred* dataset with a small number of training samples.

B. SUBTLE DISTINCTIONS BETWEEN PROTEINS WITHIN THE SAME FAMILY

Our previous work has demonstrated that *CFreeEnS* is effective in predicting the antigenic similarity between the hemagglutinin protein of influenza viral strains [21], indicating that *CFreeEnS* for protein sequence pairs can distinguish subtle differences between proteins within the same family.

Quantifying the antigenic similarity between viral strains is an essential step in selecting and manufacturing vaccine

TABLE 3. Datasets and accuracy of predicting the antigenicity of influenza viruses using different encoding schemes.

Datasets	# SeqPairs	MutCounts	RegionBand	CFreeEnS
A/H1N1	355	0.824	0.706	0.859
A/H3N2	791	0.843	0.790	0.885
A/H5N1	293	0.863	0.858	0.915
A/H9N2	118	0.775	0.804	0.850
Combined	1557	0.698	0.751	0.846

#SeqPairs: Number of sequence pairs;

strains. But the traditional hemagglutination inhibition (HI) assays are costly and require high biosafety facilities for high pathogenic subtypes, resulting in limited HI assays data. We collected the HI assays data and the corresponding protein sequence pairs of four flu subtypes, namely influenza A/H1N1, A/H3N2, A/H5N1 and A/H9N2, forming a combined dataset [30]. For each subtype, all substitution matrices were evaluated. Subsequently, the ones resulted in the best performance in each dataset were selected to encode the combined dataset. Thus, we used four substitution matrices to encode the combined dataset of various influenza subtypes. The random forest with a maximum depth of 9 was used as the downstream classifier.

Most works analyzing the antigenicity of influenza design subtype-specific features, which could be applicable to other subtypes but not work so well [31]–[33]. For comparison, We adapted a mutation-counts-based method proposed by Liao *et al.* [31] to all subtypes. Also, we compared the *CFreeEnS* with a universal model proposed by Peng *et al.* [34], which is based on regional bands cross subtypes. The two methods for comparison are shorted for *MutCounts* and *RegionBand* respectively. Using the same downstream learning method, the performances of *MutCounts*, *RegionBand* and *CFreeEnS* on each subtype and the combined dataset with four subtypes(A/H1N1, A/H3N2, A/H5N1 and A/H9N2) are listed in Table 3. *CFreeEnS* always achieves the highest predicting accuracy, not only on the datasets with one subtype, but also on the combined dataset with diverse subtypes. We also analyzed the performance of *CFreeEnS* in transfer learning models, i.e. training the model on one subtype but testing on another subtype with fewer samples. *CFreeEnS* always achieves higher predicting accuracy than *MutCounts* and *RegionBand* [21]. The results indicated that *CFreeEnS* could capture the cross-subtype features of influenza viruses.

IV. DISCUSSION AND CONCLUSION

A dilemma in feature engineering is that domain-specific knowledge can benefit the design of an effective data representation for a specific dataset, but it can be tedious, time-consuming and limited by human subjectivity. A representation with more generic priors can help automate the design of features and facilitate large-scale analyses of different datasets. The explosion of protein sequence data and the increasing availability of computing power make the latter more urgent and promising. When it comes to the protein classification problem, existing methods of protein representation have achieved moderate performance, and can compete with the design incorporated with domain-specific knowledge especially classifying proteins into families [35].

To compare the effectiveness of *CFreeEnS*, we chose four protein classification problems, detecting peptides with different functions, namely antimicrobial peptides, tumor homing peptides, hemolytic peptides and phage virion proteins. Sequences in the four selected datasets not only have different biological backgrounds, but also vary in the data size, sample distribution and sequence length distribution. The number of training samples range from hundreds to thousands, and the lengths of peptides range from several residues to a few hundred. The predicting accuracy of *CFreeEnS* exceeds 88% on each dataset, regardless of being balanced or not for the positive and negative classes, with short or long peptides. The results show the robustness of *CFreeEnS*, suggesting that *CFreeEnS* encodes the generic priors of protein sequences into features representative enough for distinguishing them at the functional level. For distinguishing subtle differences among protein sequences with only several mutations, but showing different phenotypes, we applied *CFreeEnS* to predicting the antigenic similarity between hemagglutinin proteins of influenza viruses, showing its ability to capture cross-subtype antigenic features of influenza viruses [21].

CFreeEnS is a protein representation heavily depends on the AAindex database, i.e. the known properties of amino acids. Therefore, features selected by the downstream learning models are easy to interpret through the analysis of variable importance. It has been demonstrated that with features created by *CFreeEnS*, protein functions can be predicted with high accuracy. However, *CFreeEnS* is not good at disentangling more abstract features, or providing a new angle to explain the relationship between genotype and phenotype. The *m-NGSG*, inspired from NLP, although not as good as *CFreeEnS* on the mentioned tasks, provides a novel perspective to treat the biological sequences. The abstract features generated by *m-NGSG* can be taken as new properties of a group of amino acids. Graphic representations of protein sequences are also interesting, providing visual qualitative inspection of sequences, but they are not efficient in describing long protein sequences [36]. Different representations of protein sequences may disentangle or hide different aspects. An encoding scheme capturing the known generic properties of amino acids can help automate the process of constructing features and facilitate annotating protein functions. For profiling other aspects of proteins, e.g. predicting the protein folds, computational predictions using other novel representations may give more insights.

ACKNOWLEDGMENT

The authors would like to thank Dr.Fransiskus Xaverius Ivan and Dr.Shamima Banu Binte SM Rashid for sharing their pearls of wisdom with us during the course of this research. They are also immensely grateful to the reviewers for their comments on an earlier version of the manuscript,

although any errors are our own and should not tarnish the reputations of these esteemed persons.

REFERENCES

[1] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[2] S.-Y. Kung and M.-W. Mak, "Feature selection for genomic and proteomic data mining," in *Machine Learning in Bioinformatics*. Hoboken, NJ, USA: Wiley, 2009, pp. 1–45.

[3] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand. (ASRU)*, Dec. 2011, pp. 24–29.

[4] A. S. Martinez-Vernon *et al.*, "An improved machine learning pipeline for urinary volatiles disease detection: Diagnosing diabetes," *PLoS ONE*, vol. 13, no. 9, p. e0204425, 2018.

[5] F. Wang, T. Xu, T. Tang, M. Zhou, and H. Wang, "Bilevel feature extraction-based text mining for fault diagnosis of railway systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 1, pp. 49–58, Jan. 2017.

[6] A. Zheng and A. Casari, *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. Sebastopol, CA, USA: O'Reilly Media, 2018.

[7] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[8] N. Zou, Y. Zhu, J. Zhu, M. Baydogan, W. Wang, and J. Li, "A transfer learning approach for predictive modeling of degenerate biological systems," *Technometrics*, vol. 57, no. 3, pp. 362–373, 2015.

[9] P. Domingos, "A few useful things to know about machine learning," *Commun. ACM*, vol. 55, no. 10, pp. 78–87, 2012.

[10] D. Petrey and B. Honig, "Is protein classification necessary? Toward alternative approaches to function annotation," *Current Opinion Structural Biol.*, vol. 19, no. 3, pp. 363–368, 2009.

[11] C. B. Kennedy, "Multiplexed microfluidic devices and systems," U.S. Patent 6 086 740 A, Jul. 11, 2000.

[12] G. K. Hirst, "The quantitative determination of influenza virus and antibodies by means of red cell agglutination," *J. Exp. Med.*, vol. 75, no. 1, pp. 49–64, 1942.

[13] M. H. Smith, "The amino acid composition of proteins," *J. Theor. Biol.*, vol. 13, pp. 261–282, Dec. 1966.

[14] K.-C. Chou, "Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes," *Bioinformatics*, vol. 21, no. 1, pp. 10–19, 2004.

[15] B. S. Cherian and A. S. Nair, "Protein location prediction using atomic composition and global features of the amino acid sequence," *Biochem. Biophys. Res. Commun.*, vol. 391, no. 4, pp. 1670–1674, 2010.

[16] K. Blekas, D. I. Fotiadis, and A. Likas, "Motif-based protein sequence classification using neural networks," *J. Comput. Biol.*, vol. 12, no. 1, pp. 64–82, 2005.

[17] A. Ben-Hur and D. Brutlag, "Sequence motifs: highly predictive features of protein function," in *Feature Extraction*. Berlin, Germany: Springer, 2006, pp. 625–645.

[18] I. Vujaklija, A. Bielen, T. Paradžik, S. Biin, P. Goldstein, and D. Vujaklija, "An effective approach for annotation of protein families with low sequence similarity and conserved motifs: Identifying gdsl hydrolases across the plant kingdom," *BMC Bioinf.*, vol. 17, no. 1, p. 91, 2016.

[19] E. Asgari and M. R. K. Mofrad, "Continuous distributed representation of biological sequences for deep proteomics and genomics," *PLoS ONE*, vol. 10, no. 11, p. e0141287, 2015.

[20] S. M. A. Islam, B. J. Heil, C. M. Kearney, and E. J. Baker, "Protein classification using modified *n-grams and skip-grams*," *Bioinformatics*, vol. 34, no. 9, pp. 1481–1487, 2017.

[21] X. Zhou, R. Yin, C.-K. Kwoh, and J. Zheng, "A context-free encoding scheme of protein sequences for predicting antigenicity of diverse influenza A viruses," *BMC Genomics*, vol. 19, no. 10, p. 936, 2018.

[22] S. Kawashima, P. Pokarowski, M. Pokarowska, A. Kolinski, T. Katayama, and M. Kanehisa, "AAindex: Amino acid index database, progress report 2008," *Nucleic Acids Res.*, vol. 36, pp. D202–D205, Jan. 2007.

[23] K. Tomii and M. Kanehisa, "Analysis of amino acid indices and mutation matrices for sequence comparison and structure prediction of proteins," *Protein Eng., Des. Select.*, vol. 9, no. 1, pp. 27–36, 1996.

[24] P. Bhadra, J. Yan, J. Li, S. Fong, and S. W. I. Siu, "AmPEP: Sequence-based prediction of antimicrobial peptides using distribution patterns of amino acid properties and random forest," *Sci. Rep.*, vol. 8, no. 1, 2018, Art. no. 1697.

[25] X. Xiao, P. Wang, W.-Z. Lin, J.-H. Jia, and K.-C. Chou, "iAMP-2L: A two-level multi-label classifier for identifying antimicrobial peptides and their functional types," *Anal. Biochem.*, vol. 436, no. 2, pp. 168–177, 2013.

[26] P. K. Meher, T. K. Sahu, V. Saini, and A. R. Rao, "Predicting antimicrobial peptides with improved accuracy by incorporating the compositional, physico-chemical and structural features into Chou's general PseAAC," *Sci. Rep.*, vol. 7, p. 42362, Feb. 2017.

[27] A. Sharma *et al.*, "Computational approach for designing tumor homing peptides," *Sci. Rep.*, vol. 3, p. 1607, Apr. 2013.

[28] K. Chaudhary *et al.*, "A Web server and mobile app for computing hemolytic potency of peptides," *Sci. Rep.*, vol. 6, Mar. 2016, Art. no. 22843.

[29] H. Ding, P.-M. Feng, W. Chen, and H. Lin, "Identification of bacteriophage virion proteins by the ANOVA feature selection and analysis," *Mol. BioSyst.*, vol. 10, no. 8, pp. 2229–2235, 2014.

[30] R. B. Squires *et al.*, "Influenza research database: An integrated bioinformatics resource for influenza research and surveillance," *Influenza Other Respiratory Viruses*, vol. 6, no. 6, pp. 404–416, 2012.

[31] Y.-C. Liao, M.-S. Lee, C.-Y. Ko, and C. A. Hsiung, "Bioinformatics models for predicting antigenic variants of influenza A/H3N2 virus," *Bioinformatics*, vol. 24, no. 4, pp. 505–512, 2008.

[32] X. Du *et al.*, "Mapping of H3N2 influenza antigenic evolution in China reveals a strategy for vaccine strain recommendation," *Nature Commun.*, vol. 3, Feb. 2012, Art. no. 709.

[33] J. Qiu, T. Qiu, Y. Yang, D. Wu, and Z. Cao, "Incorporating structure context of HA protein to improve antigenicity calculation for influenza virus A/H3N2," *Sci. Rep.*, vol. 6, p. 31156, Aug. 2016.

[34] Y. Peng *et al.*, "A universal computational model for predicting antigenic variants of influenza A virus based on conserved antigenic structures," *Sci. Rep.*, vol. 7, Feb. 2017, Art. no. 42051.

[35] T. Frickey and A. Lupas, "CLANS: A Java application for visualizing protein families based on pairwise similarity," *Bioinformatics*, vol. 20, no. 18, pp. 3702–3704, 2004.

[36] J. Wen and Y. Zhang, "A 2D graphical representation of protein sequence and its numerical characterization," *Chem. Phys. Lett.*, vol. 476, nos. 4–6, pp. 281–286, 2009.

XINRUI ZHOU received the B.Eng. degree in software and engineering and the secondary degree in finance from Wuhan University, China, in 2015. She is currently pursuing the Ph.D. degree with the Biomedical Informatics Laboratory, School of Computer Science and Engineering, Nanyang Technological University.

RUI YIN received the B.S. degree in automation from Shandong University, China, in 2013, and the M.Sc. degree in control engineering from Central South University, China, in 2016. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Nanyang Technological University, Singapore.

His research interests include data mining and pattern recognition to make sense of big heterogeneous data for real application in engineering and biomedical science.

JIE ZHENG received the B.Eng. degree (Hons.) in computer science from Zhejiang University, in 2000, and the Ph.D. degree in computer science from the University of California at Riverside, Riverside, in 2006. From 2006 to 2011, he was a Postdoctoral Visiting Fellow and a Research Scientist with the National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, USA. He was an Assistant Professor with the School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore. Since 2012, he has been an Adjunct Senior Research Scientist with the Genome Institute of Singapore, A*STAR, Singapore. In 2018, he joined ShanghaiTech University as an Associate Professor. He has published more than 50 journal papers (14 of which have impact factors higher than 5) and over 40 conference papers. He has served as a PC Member and a Reviewer for a number of international conferences and as a Program Co-Chair for two conferences. Moreover, he reviews many papers for top-tier journals (e.g., *Bioinformatics* and *Nucleic Acids Research*) each year. In Singapore, he was a PI of three competitive external research grants at national level (with total funding more than 2 million Singapore dollars), and he participated in more than 10 other projects as a Co-PI or a Collaborator. In the International Genetically Engineered Machine Competition 2015, he was the Coach of mathematical modeling for the NTU Team of students, who received the Gold Medal. At the International Conference on Bioinformatics 2017, he received the Best Paper Award (Gold Medal) in BMC Track. In 2016 and 2017, he was nominated for the Nanyang Education Award at NTU.

CHEE-KEONG KWOH received the bachelor's degree (Hons.) in electrical engineering and the master's degree in industrial system engineering from the National University of Singapore, Singapore, in 1987 and 1991, respectively, and the Ph.D. degree from the Imperial College of Science, Technology and Medicine, University of London, in 1995. He has been with the School of Computer Engineering, Nanyang Technological University, since 1993, where he is currently the Program Director of the M.Sc. degree in Bioinformatics Program.

His research interests include data mining, soft computing, graph-based inference, and their applications in bioinformatics and biomedical engineering. He has done significant research work in his research areas and has published many quality international conferences and journal papers. He has often been invited as an Organizing Member or a Referee and a Reviewer for a number of premier conferences and journals, including GIW, the IEEE BIBM, RECOMB, PRIB, BIBE, ICDM, and iCBBE. He is a member of the Association for Medical and Bioinformatics and the Imperial College Alumni Association of Singapore. He has provided many services to professional bodies in Singapore and was conferred the Public Service Medal by the President of Singapore, in 2008. He is an Editorial Board Member of the *International Journal of Data Mining and Bioinformatics*, *The Scientific World Journal*, *Network Modeling and Analysis in Health Informatics and Bioinformatics*, *Theoretical Biology Insights*, and *Bioinformation*. He has been a Guest Editor for many journals, such as the *Journal of Mechanics in Medicine and Biology*, and the *International Journal on Biomedical and Pharmaceutical Engineering*.