Nguyen et al. BMC Medical Genomics 2020, 13(Suppl 10):155
https://doi.org/10.1186/s12920-020-00779-w

BMC Medical Genomics

RESEARCH Open Access

TNFPred: identifying tumor necrosis factors using hybrid features based on word embeddings

Trinh-Trung-Duong Nguyen[1], Nguyen-Quoc-Khanh Le[2,3], Quang-Thai Ho[1], Dinh-Van Phan[4] and Yu-Yen Ou[1*]

From The 18th Asia Pacific Bioinformatics Conference
Seoul, Korea. 18-20 August 2020

Abstract

Background: Cytokines are a class of small proteins that act as chemical messengers and play a significant role in essential cellular processes including immunity regulation, hematopoiesis, and inflammation. As one important family of cytokines, tumor necrosis factors have association with the regulation of a various biological processes such as proliferation and differentiation of cells, apoptosis, lipid metabolism, and coagulation. The implication of these cytokines can also be seen in various diseases such as insulin resistance, autoimmune diseases, and cancer. Considering the interdependence between this kind of cytokine and others, classifying tumor necrosis factors from other cytokines is a challenge for biological scientists.

Methods: In this research, we employed a word embedding technique to create hybrid features which was proved to efficiently identify tumor necrosis factors given cytokine sequences. We segmented each protein sequence into protein words and created corresponding word embedding for each word. Then, word embedding-based vector for each sequence was created and input into machine learning classification models. When extracting feature sets, we not only diversified segmentation sizes of protein sequence but also conducted different combinations among split grams to find the best features which generated the optimal prediction. Furthermore, our methodology follows a well-defined procedure to build a reliable classification tool.

Results: With our proposed hybrid features, prediction models obtain more promising performance compared to seven prominent sequenced-based feature kinds. Results from 10 independent runs on the surveyed dataset show that on an average, our optimal models obtain an area under the curve of 0.984 and 0.998 on 5-fold cross-validation and independent test, respectively.

Conclusions: These results show that biologists can use our model to identify tumor necrosis factors from other cytokines efficiently. Moreover, this study proves that natural language processing techniques can be applied reasonably to help biologists solve bioinformatics problems efficiently.

Keywords: Machine learning, Binary classification, Natural language processing, Feature extraction

* Correspondence: yienou@gmail.com
[1]Department of Computer Science and Engineering, Yuan Ze University, Taoyuan 32003, Taiwan
Full list of author information is available at the end of the article

Background

Cytokines is a varied group of polypeptides, usually linked to inflammation and cell differentiation or death. Among major families of cytokines (interleukins (IL), interferons (IFNs), tumor necrosis factors (TNFs), chemokine and various growth factors, comprised of transforminggrowth factor b (TGF-b), fibroblast growth factor (FGF), heparin binding growth factor (HBGF) and neuron growth factor (NGF)) [1], tumor necrosis factors are versatile cytokines with a wide range of functions that attracts abundant of biological researchers (see, e.g. [2–6]). TNFs can take part in pathological reactions as well as involve in a variety of processes, such as inflammation, tumor growth, transplant rejection, etc. [3, 6]. TNFs act through their receptors at the cellular level to activate separate signals that control cell survival, proliferation or death. Furthermore, TNFs play two opposite roles in regard to cancer. On the positive side, activity in the suppression of cancer is supposed to be limited, primarily due to system toxicity of TNFs. On the negative side, TNFs might act as a promoter of the endogenous tumor through their intervention to the proliferation, invasion and tumor cell metastasis thus contributing to tumor provenance. Such TNFs' effect on cancer cell death makes them a probable therapeutic for cancer [3]. Moreover, in the United States and other nations, patients with TNF-linked autoimmune diseases have been authorized to be treated with TNF blockers [2]. In cytokine network, TNFs and other factors such as interleukins, interferons form an extremely complicated interactions generally mirroring cytokine cascades which begin with one cytokine causing one or additional different cytokines to express that successively trigger the expression of other factors and generate complex feedback regulatory circuits. Abnormalities in these cytokines, their receptors, and the signaling pathways that they initiate involve a broad range of illnesses [7–12]. Interdependence between TNFs and other cytokines accounts for such diseases. For instance, TNFs and interleukin-1 administers TNF-dependent control of *Mycobacterium tuberculosis* infection [12]. Another example is the TNF and type I interferons interactions in inflammation process which involve rheumatoid arthritis and systemic lupus erythematosus [13]. For the above reasons, identification of TNFs from other cytokines presents a challenge for many biologists.

To date, in bioinformatics, several research teams have built machine learning models to predict cytokines and achieved high performance [14–20]. Bases on these research, it is noted that Support Vector Machine (SVM) [21] classifier is a solid foundation for building prediction models. Regarding the feature extraction method used, considerable efforts were made to create hybrid features through useful characteristics extracted from sequences such as compositions of amino acids and amino acid pairs, physicochemical properties, secondary structures, and evolutionary information to improve predictive ability. It therefore has been verified that a primary concern for biological scientists is a discriminatory and effective feature set. Additionally, among these groups, some have further classified cytokines into tumor necrosis factor family and other subfamilies. For example, Huang et al. utilized dipeptide composition to developed CTKPred, a tool for identifying cytokines families and subfamilies [15]. This classification method accomplished an accuracy of 92.5% on 7-fold cross-validation when predicting cytokines, and also enabled 7 significant cytokine classes to be predicted with 94.7% accuracy, overall. Although the results seem promising, we noted that the authors set cutoff threshold value as very high as 90% for excluding homologous sequences within the dataset. Three years later, Lata et al. constructed CytoPred using the combination of support vector machine and Psi Blast to classify cytokines into 4 families and 7 subfamilies [16]. Despite the fact that CytoPred outperforms CTKPred, Lata's group reused the dataset created by Huang et al. which poses the similar concern about the high value for cutoff threshold. We believe that if the cutoff threshold is lower, the more the homology bias will be excluded from the surveyed dataset in a stricter manner and thus increasing reliability of the prediction model [22]. Considering the important roles of TNFs and the imperfection of such models like CTKPred and CytoPred, a fresh approach to classifying TNF among cytokines is required, so our study seeks to discover a solution to this issue.

From the interpretation of the genetic alphabet order, scientists have found that in terms of in composition, biological sequences, particularly protein sequences, are comparable to human language [23]. A growing variety of scientists are relating these molecule sequences as a special textual data and examining them using available text mining techniques. Initial stage of this transformation is to convert (biological) words to real numbers like those in vectors. This means that every word is encoded by one or more values that locate it in a discretionary space. Regarding this issue, NLP researchers have seen some landmarks. They are one-hot encoding, co-occurrence matrix, and word embedding techniques. A one-hot vector represents a word in text and encompass a dimension up to a vocabulary size, where a sole entry resembling the word is a one and all opposite entries are zeros. This presents a significant drawback because this method does not group commonly co-occurring items together in the representation space. Furthermore, scalability is another limitation as the representation size increases with the corpus size. Later,

the concept of co-occurrence matrix arrived. Co-occurrence matrix represents a word while considering its adjacent words. The underlying reasoning for co-occurrence matrix follows an assumption that "context is king". This technique has been believed to take a radical change within the field of NLP. During this methodology, a window size s is selected to formulate a co-occurrence matrix where words appearing along in the chunk with length s are going to be counted. The linguistics and grammar relationships made by this method are compelling yet this idea encounters some limitations such as high-dimensional space and thus computational expensive. These barriers resulted in the arrival of word embeddings, or continuous word vectors that are illustrations of words that also considers the context via neighboring words yet keep the number of dimensions greatly lower. A typical word embedding vector accumulates additional characteristics with reduced dimensions, and therefore more efficient. Recently, its utilization have been seen as underlying principle of various word embedding-related research such as sentiment classification, bilingual word translation, information retrieval, etc. with appreciable accomplishment [24–27].

Motivated by remarkable achievements in NLP using word embeddings, in this study, we tried to use NLP technique for extracting features. We transferred the protein sequences into "protein sentences" comprised of composing biological words from which word vectors were created. Next, we trained the Fast Text model to created word embeddings on which final word embedding-based features were generated. Finally, we employed some advanced machine learning algorithms for classification.

Many studies in bioinformatics show that [28–37] to build a helpful statistical predictor from primary sequences to help biologists solve a certain problem, researchers ought to obey the rules from 5-step rule which is restated here for clarity: (1) the accurate procedure to compose training and independent dataset to build and evaluate the predictive model; (2) the way to represent amino acid chain samples in mathematical forms that can genuinely mirror their characteristic interconnection with the predicted target; (3) the best approach to present or build up an effective and robust predictive algorithm; (4) the method to correctly conduct cross-validation tests to justly evaluate the predictor's unseen accuracy; (5) the way to set up a convenient and accessible web-server for the predictor that is helpful to general users. Herein, we strictly follow the the first four steps recommended by Chou et al. [22]. For the last step, we deposited our prediction model on a public repository which can also assist general users.

Results

Number of features with n-gram sizes

It is necessary to figure out the quality as well as the quantity of extracted features for the learning process. Therefore, we calculated the number of features of word embedding-based vectors that were input in our binary classifiers (see Table 1). As we randomly divided the data into the training part and testing part and repeated this process for 10 times resulting 10 datasets for experiments, the numbers of n-grams vary from dataset to dataset. We found that when biological word length was equal to 3 ($n = 3$), the numbers of features were highest (from 1736 to 1915) compared to those with length 1, 2, 4, or 5. In this group, the numbers of features when the length of biological words is equal to 2 is the second highest (from 395 to 398). Therefore, when we combined between 2-g and 3-g features, the total number of hybrid features is the highest one (from 2131 to 2313).

Amino acid composition analysis of TNFs and non-TNFs

We analyzed the amino acid composition of TNFs and non-TNFs by calculating the frequency of comprised amino acid. We used the entire dataset for this analysis. Figure 1 illustrates the amino acids that contributed unevenly in two different groups and Fig. 2 displays the variance of the composition. We noticed that the amino acid K, G, L, I occurred at the significant frequency difference. Therefore, in recognizing TNF proteins, these amino acids definitely play an important role. According to these amino acid contributions, the model can evaluate these unique characteristics to define tumor necrosis factors.

We further carried out statistical tests on the entire dataset to evaluate the distinction between TNFs and non-TNFs amino acid composition variance. We performed the tests on 3 types of composition: single amino acid, dipeptide and tripeptide. First, F-test has been used to test the null hypothesis that two datasets' variances are equal. We presumed that there are equal variances between TNFs and non-TNFs. After running the test, in all three cases, we acknowledged $F < F\text{-critical}$, (the F

Table 1 Number of features input in our binary classifiers

Feature types	Number of features
1-g	20
2-g	395➔398
3-g	1736 ➔ 1915
4-g	60 ➔ 83
5-g	6 ➔ 11
1-g and 2-g combined	415 ➔ 418
1-g and 3-g combined	1756 ➔ 1935
2-g and 3-g combined	2131 ➔ 2313

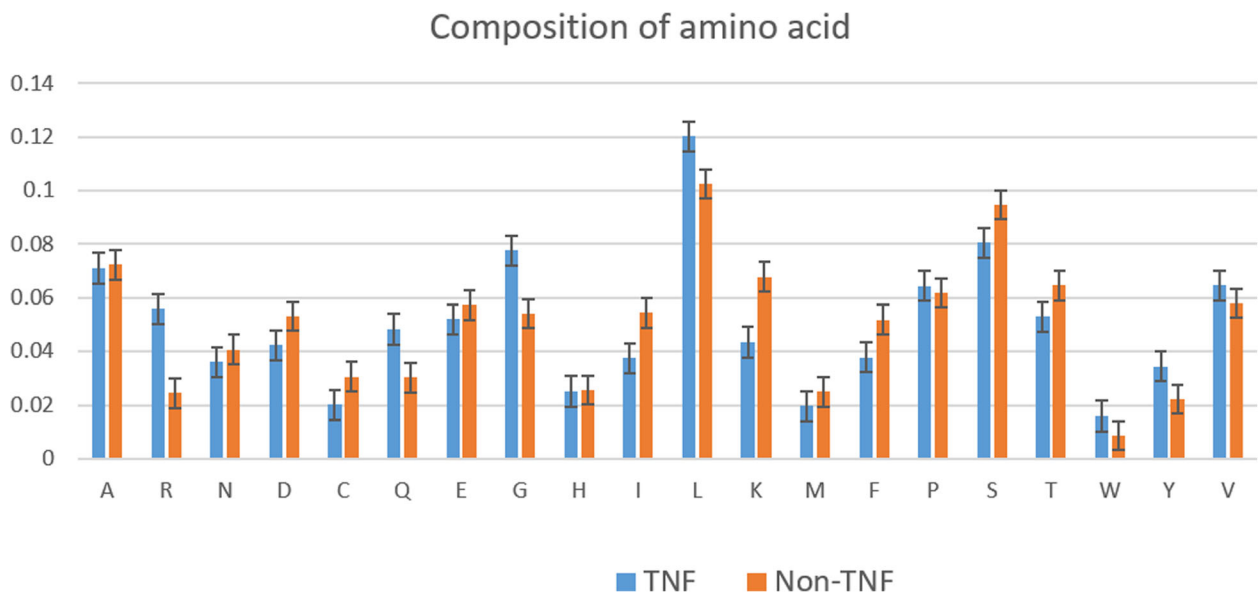

Fig. 1 Amino acid composition of the surveyed TNF and non-TNF proteins. The amino acid K, G, L, I occurred at the significant frequency difference

and F-critical values of 3 cases are provided in the Table S1, Additional file 1), so we accepted the null hypothesis. It implies that the variances of two datasets in terms of single amino acid, dipeptide and tripeptide composition are equal or the amino acid structures of the dataset are not substantially different. Therefore, an efficient approach for extracting useful features is critically needed to identify TNFs with high performance.

Next, we performed the unpaired t-test with values of amino acid, dipeptide and tripeptide composition on positive group (18 TNF sequences) and negative group (non-TNF sequences). The Table S2, S3, S4 in the displays the *p*-values of the tests.

From the Table S2, we can see that the *p*-values are generally high which means that there are not much difference in amino acid compostion between TNF and non-TNF sequences. However, at the common criterion (*p*-value of 0.05), six amino acids C, G, K, W, Y, and V shows significant frequency difference. In addition, from the Table S3, it can be seen that GL, LY, FG are amino acid pairs with lowest *p*-values which means that they may contribute more for distinguishing positive and negative data. It is also noted from the Table S4 that these patterns can be found in tripeptides with lowest *p*-values (GLY, FFG, FGA, LYY). The other tripeptides with lowest *p*-values are QDG, TFF and VYV.

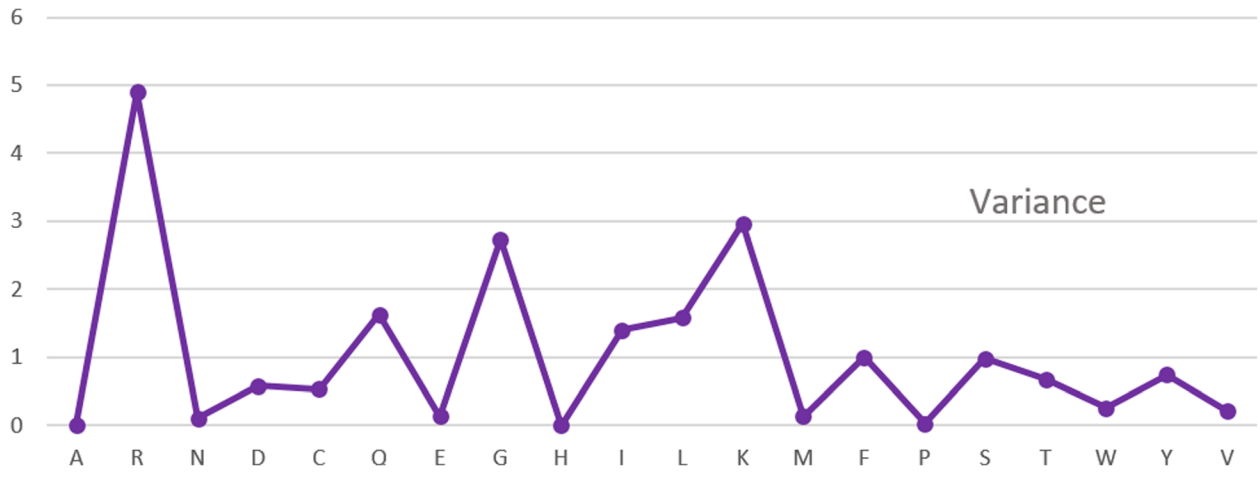

Fig. 2 The variance of amino acid composition of the surveyed TNF and non-TNF proteins

Visualization of word embedding-based features in representation space

In this analysis, we would like to envisage the surveyed protein feature vectors, which have a high number of dimensions and are very hard to see in the raw form. We performed t-Distributed Stochastic Neighbor Embedding [38] (t-SNE), an unsupervised, non-linear method mainly used for data exploration and high-dimensional visualization, on the whole dataset. t-SNE allows us to reduce data with hundreds or even thousands of dimensions to just two (forming a point in two-dimensional space) and thus being suitable to plot our data on a two-dimensional space. We used the word embedding-based vectors generated from extraction method described in section 2.2 as input into t-SNE algorithm to create two-dimensional "maps" for TNF and non-TNF sequences. We also noted that t-SNE contains a tunable parameter, "perplexity", which indicated balance attention between local and global aspects of visualized data. The perplexity value has a complex effect on the resulting pictures and typical values are between 5 and 50. In order to get a multiple view about our data, we performed our visualization with perplexity values of 25 and 50. Fig. S1, S2, S3, S4, S5, S6, S7, S8 in Additional file 2 visualize our dataset corresponding to 5 different word lengths (length = 1, 2, 3, 4, 5) and 3 ways we combined n-gram features, respectively. From these figures, it is interesting that the two-dimensional "maps" for TNF and non-TNF sequences have different shapes. Moreover, features with n-gram, where n takes values of 1, 2, 4, and 5 are points that tend to be farther from each other compared to those of the rest. In addition, when we changed the perplexity from 25 to 50, the shape comprised from the points changed differently among feature types.

The influence of n-gram sizes on the performance of the models

In this prior research, we intended to compare the efficacy of distinct segmentation sizes and the mixture of n-gram used. The feature sets were produced as outlined in section 2.2. We used SVM as the binary classifier. (We used scikit-learn library (version 0.19.1) for building all classifiers in this study). In both 5-fold cross-validation and independent test, we assessed the general performance of each experiment using the AUC scores. The outcomes with highlighted best values in bold are shown in Table 2. We found that among 5 segmentation sizes, feature type corresponding to biological words of length = 3 helped yielded the highest average performance. This may come from the larger number of features because when the biological words length is equal to 3 ($n = 3$), the numbers of features are highest (from 1736 to 1915, see Table 1). Furthermore, model with the 2-g and 3-g combined feature set achieved the best average AUC of 0.984 ± 0.298 and 0.998 ± 0.277 on cross-validation and independent test, respectively. From these outcomes, we chose this feature type for further experiments.

Table 2 AUC performance of SVM classifier on embedding features with different biological word lengths

Feature	AUC	
	5-fold cross-validation	Independent
1-g	0.856 ± 0.485	0.848 ± 0.525
2-g	0.901 ± 0.416	0.883 ± 0.599
3-g	0.934 ± 0.423	1 ± 0
4-g	0.617 ± 0.63	0.563 ± 0.751
5-g	0.543 ± 0.539	0.574 ± 0.686
1-g and 2-g combined	0.952 ± 0.416	0.934 ± 0.48
1-g and 3-g combined	0.96 ± 0.42	0.921 ± 0.497
2-g and 3-g combined	**0.984** ± 0.298	**0.998** ± 0.277

(Each result is reported in format: $m \pm d$, where m is the mean and d is the standard deviation across the ten runs)

Comparison between proposed features and other sequence-based feature types

We used SVM to evaluate the efficiency of our suggested features against 7 frequently used feature types i.e. amino acid composition (AAC), amino acid pair composition (DPC), position-specific scoring matrix (PSSM) features and the combination of these types. After a search for the ideal parameters, we conducted these experiments on both cross-validation and independent data. We found that with our suggested features, the SVM models reached the best performance (95.82, 97.59, 83.67%, 0.83 on the validation data, 96.49, 98**,** 85%, 0.86 on the independent data in terms of accuracy, specificity, sensitivity, and MCC, respectively, see Table 3). These values demonstrate that, compared to other sequence-based feature kinds, our word embedding-based features are more discriminatory and efficient. For long time, it is well-known that evolutionary information or PSSM is an efficient feature type to solve numerous bioinformatics problem yet it takes a lot of time to be created. These results show that, with our approach for generating features, we can save much time on feature extraction phase.

Comparison of different advanced classifiers using embedding features

We used the feature kind selected from the results of the experiments described in section 3.4 as inputs of 5 commonly used machine learning algorithms namely Support Vector Machine [39], k Nearest Neighbor (kNN) [40], RandomForest (RF) [41], Naïve Bayes [42]

Table 3 Performance comparison of proposed features with AAC, DPC, PSSM, and the combined features with highest performance values for each class highlighted in bold

	Cross-validation data			
Feature types	Acc (%)	Spec (%)	Sen (%)	MCC
AAC	60.69 ± 12.13	59.61 ± 15.82	68.01 ± 19.77	0.24 ± 0.09
DPC	70.42 ± 13.97	72.89 ± 17.91	51.67 ± 20.14	0.24 ± 0.13
AAC-DPC	86.48 ± 5.82	88.83 ± 8.04	69.34 ± 16.62	0.53 ± 0.08
PSSM	89.57 ± 6.68	91.88 ± 5.73	73.34 ± 18.33	0.61 ± 0.16
PSSM-AAC	91.17 ± 3.18	93.19 ± 4.11	76.34 ± 12.91	0.66 ± 0.1
PSSM-DPC	91.25 ± 3.29	93.85 ± 4.58	72.34 ± 12.58	0.64 ± 0.09
PSSM-DPC-AAC	91.25 ± 2.84	93.67 ± 3.85	73.67 ± 14.45	0.64 ± 0.1
Proposed features	**95.82** ± 1.67	**97.59** ± 2.15	**83.67** ± 7.45	**0.83** ± 0.06
	Independent data			
Feature types	Acc (%)	Spec (%)	Sen (%)	MCC
AAC	46.92 ± 25.59	42.37 ± 33.15	81.25 ± 34.49	0.20 ± 0.07
DPC	82.05 ± 23.71	84.63 ± 29.06	62.75 ± 39.92	0.46 ± 0.28
AAC-DPC	93.65 ± 2.66	94.95 ± 3.26	83.75 ± 9.59	0.73 ± 0.09
PSSM	94.85 ± 2.82	97.56 ± 2.85	74.5 ± 28.35	0.72 ± 0.26
PSSM-AAC	95.77 ± 1.45	97.42 ± 2.14	83.25 ± 11.31	0.81 ± 0.06
PSSM-DPC	95.94 ± 1.33	97.81 ± 1.49	82 ± 8.8	0.81 ± 0.07
PSSM-DPC-AAC	95.14 ± 2.02	96.77 ± 2.48	83 ± 9.7	0.78 ± 0.08
Proposed features	**96.49** ± 4.34	**98** ± 5.27	**85** ± 17.48	**0.86** ± 0.13

(Each result is reported in format: $m \pm d$, where m is the mean and d is the standard deviation across the ten runs)

and QuickRBF [43–46]. We used cross-validation data and independent data for these experiments. We also searched for the best parameters for each algorithm. The aim of this assessment is to find out which classifier obtains the greatest outputs given this kind of feature. Table 5 shows the outcomes with the greatest average performance values highlighted in bold. We discovered that, the SVM classifier outperformed the other classifiers on both cross-validation and independent data on the same suggested dataset in terms of AUC (see Table 4). Accordingly, we employed SVM to build the final prediction model.

Source codes for the replication of our model

To assist future research replication, we provide all the surveyed datasets and source codes at https://github.com/khucnam/TNFPred. From the best feature types based on the 5-fold cross-validation results in the development process, to create a simple and publicly accessible model called TNFPred to demonstrate our research. TNFPred was implemented using Python programming language with sklearn library. The repository also includes a two-step guide. The users who are unfamiliar with programming and machine learning can easily use the model and evaluate our technique.

Discussions

In this work, we provided a computational model for classifying tumor necrosis factors from cytokines. We supported biologists with data for their experiment replications and scholarly work with reliable cytokine and

Table 4 Performance comparison of five commonly used binary classifiers on proposed features

Classifier	Cross-validation data			
	Acc (%)	Spec (%)	Sen (%)	MCC
SVM	**95.82** ± 1.67	**97.59** ± 2.15	**83.67** ± 7.45	**0.83** ± 0.06
kNN	77.33 ± 3.7	75.41 ± 3.98	100 ± 0	0.47 ± 0.03
RandomForest	94.22 ± 2.3	94.20 ± 2.9	94 ± 8.43	0.75 ± 0.05
Naïve Bayes	21.59 ± 10.62	14.76 ± 11.45	100 ± 0	0.09 ± 0.06
QuickRBF	94.80 ± 1.52	99.81 ± 0.4	57.99 ± 14.25	0.72 ± 0.09
	Independent data			
	Acc (%)	Spec (%)	Sen (%)	MCC
SVM	**96.49** ± 4.34	**98** ± 5.27	**85** ± 17.48	**0.86** ± 0.13
kNN	79.39 ± 8.9	78.01 ± 10.57	93.34 ± 14.04	0.47 ± 0.09
RandomForest	97.28 ± 2.25	99 ± 2.26	80.01 ± 23.31	0.84 ± 0.14
Naïve Bayes	19.09 ± 23.76	10.99 ± 26.15	100 ± 0	0.08 ± 0.17
QuickRBF	94.12 ± 1.97	100 ± 0	50 ± 16.7	0.68 ± 0.13

(Each result is reported in format: $m \pm d$, where m is the mean and d is the standard deviation across the ten runs)

non-cytokine sequence information. Additionally, our experiments was carefully-designed to ensure the reliability of the predictive model. Specifically, we also carried the experiments with 10 runs, each with the same data but different training and testing data partitions to get more insight about the dataset. Although we kept the same sequence number distributions over these two parts, it is interesting to find that the performance is influenced by the random partitioning of data into training and testing data parts. This is reflected in the standard deviation across 10 runs. In this regard, sensitivity scores are the most unpredictable ones. We think a reason for this instability may come from the much insufficiency of positive data samples (see Table 1). Additionally, with our data, in the case of 3-g, 4-g and 5-g, the number of biological words generated (see Table 2) are far from reaching the maximum possible number following this formulae: $(20)^{length}$, where 20 is the number of standard amino acids and length is the size for sequence segmentation. This means that we did not have abundant samples to train the FastText model so that it can reach its best potential. Fortunately, when we accumulated 2-g and 3-g features to create a new feature set, we obtained the best average AUC and lowest standard deviation (see Table 2). In order to improve the performance as well as the stability of predictive models in future research, we suggest 2 tentative methods. First, for extracting useful features, scientists may try with another word representation approach such as contextualized word embeddings of which the same word (motif) will have different embeddings depending on its contextual use [47–49]. Second, for tackling the insufficiencies of positive data samples (e.g., TNF sequences in this study), transfer learning approach can be a good attempt [50]. As part of our future work, we are currently evaluating the effect of an optimal features based on contextualized word embeddings for comparison with the scheme used in this study. Moreover, we are also considering transfer learning approach for protein classification tasks with source data from proteins having general characteristics of cytokines, e.g., cell-signaling, and target data from specific kind of cytokines such as TNFs.

Conclusions

In this research, we have consistently applied word embedding techniques for identifying tumor necrosis factors from other cytokines. We assessed our performance on 5-fold cross-validation and independent testing dataset with support vector machine classifier and optimal features generated from word embeddings. Our technique showed an average five-fold cross-validation accuracy of 95.82% and MCC of 0.83 for predicting TNFs among cytokines. The average accuracy of independent datasets is 96.49% and MCC is 0.96, respectively. In addition, this research strictly adheres to the guidelines of 5-step rule with a slight modification on the last step. This makes our predictor reliable compared to other published works. Our suggested approach is also simpler, less laborious and much quicker to generate feature sets. In addition, our work could provide a foundation for future studies that can utilize natural language processing tactics in bioinformatics and computational biology.

Methods

We present a new method utilizing word embedding vectors to efficiently identify tumor necrosis factors from cytokines. Figure 3 displays a flowchart of our study consisting two main sub-processes: using FastText to train vector model and support vector machine classifier to train supervised learning classification. Each sub-process is explained further in the sections below.

Data collection

From UniProt database [51] (release 2019_05), we first collected positive data which are tumor necrosis factors by using the query as "family:"tumor necrosis factor family" AND reviewed:yes". From this step, we obtained 106 protein sequences. After that, we collected negative data which are 1023 cytokine sequences from other major cytokine families including 347 interleukins, 205 chemokines, 227 TGF-betas, 138 interferons and 106 others. Using PSI Blast [52], we dismissed sequences with similarity higher than 20% which ensures our study strictly follows the first principle in the 5-step rule. We also removed protein sequences that contain uncommon amino acid (BJOUXZ). After this step, we were left with 18 TNF proteins and 133 non-TNF proteins. These are proteins from various organisms such as human, mouse, rat, bovine and fruit fly, etc. As the number of protein sequences left for survey is small (151 sequences), the prediction model might be biased toward the way we separated the data parts used for training and testing. Accordingly, we randomly divided 151 surveyed sequences into cross-validation data and independent data for building and testing the models, respectively. We repeated this with process for 10 times when keeping the same sequence number distributions over these two parts. This means that all our later experiments were carried out on 10 different datasets and the results were averaged. Table 5 shows the detailed number of sequences for each part in each of 10 datasets.

Feature extraction for identifying tumor necrosis factors

According to the fundamental concepts, in our research, every protein sequence is divided into segments with

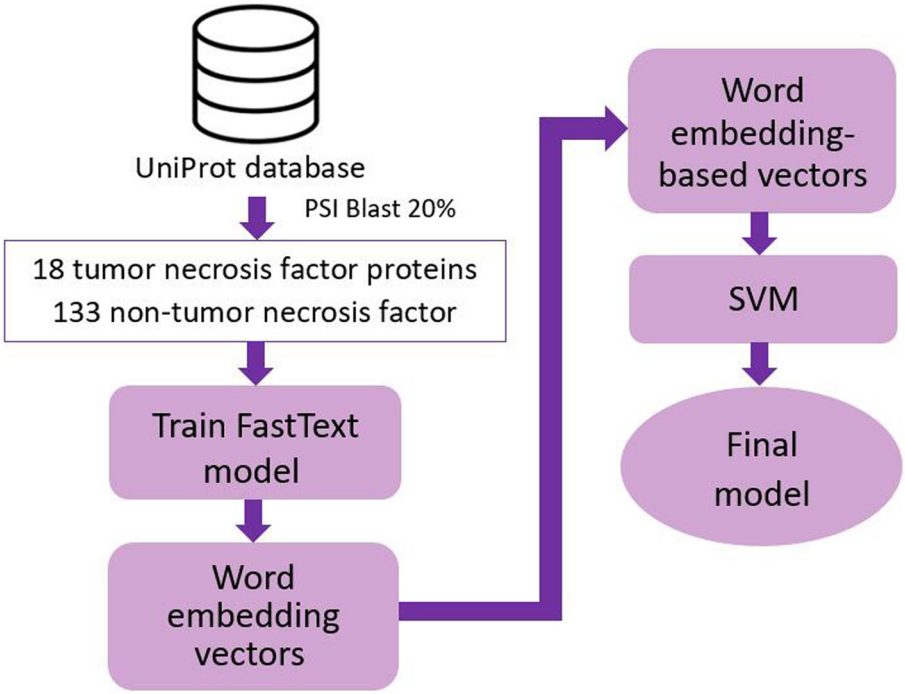

Fig. 3 The flowchart of this study. First, surveyed dataset was used to train a FastText model and then we used this trained model to generate word embedding vectors. Next, word embedding-based vectors were created. In the end, support vector machine classifier was used for classification

same length and allowed overlapping. We then trained the neural network via FastText [53, 54] to generate the word embedding vector corresponding to each word. During this method, rather than employing a distinct vector illustration for every word, we considered inner arrangement in every word: each word w was described as a bag of character n-grams. In this manner, particular boundary characters "<" and ">" are supplementary at the start and finish of words. If taking QIGEF an instance, the following character n-grams: "<QI", "QIG", "IGE", "GEF", "EF>" and the particular sequence "<QIGEF >" will be counted when $n = 3$. For this reason, content of shorter words may be retained, which may seems as as n-grams of different words. This conjoint permits the meanings of suffixes/prefixes to be taken. This enable us to make the most of characteristics from rare words, that contributes considerably to the potency of the tactic [53]. It should be noted that we unbroken all the initial parameters of FastText and specify the embedding vector dimension as one. This implies that every biological word is delineated by only 1 real number.

Since surveyed sequence lengths are changing, the number of split words also varies. Machine learning algorithms, however, involve an equal amount of variables in the data samples. We addressed this difficulty using two steps: (1) First, we built L, an ordered list containing all protein words from the training data. (We v to denote the number of words in L), (2) Second, we used v embedding vectors appended one after another to describe each data sample. In this representation, the i^{th} vector is the embedding vector ensemble to the i^{th} word within the ordered list. During this step, it is important to note that if the i^{th} word is not present in the protein sequence, its corresponding embedding vector is adequate to zero. Likewise, if the i^{th} word emerges m times within the sequence, its corresponding embedding vector was increased m times. In this fashion, we would quantify the prevalence or contribution of the every biological word to a full feature vector. These biological words can be correlated to protein sequence motifs, that show their valuable characteristics for discriminating protein function [55]. The use of these characteristics

Table 5 Statistics of the surveyed TNF and non-TNF sequences

	Original	After 20% similarity check	Cross-validation	Independent
TNF	106	18	14	4
Non-TNF	1023	133	103	30

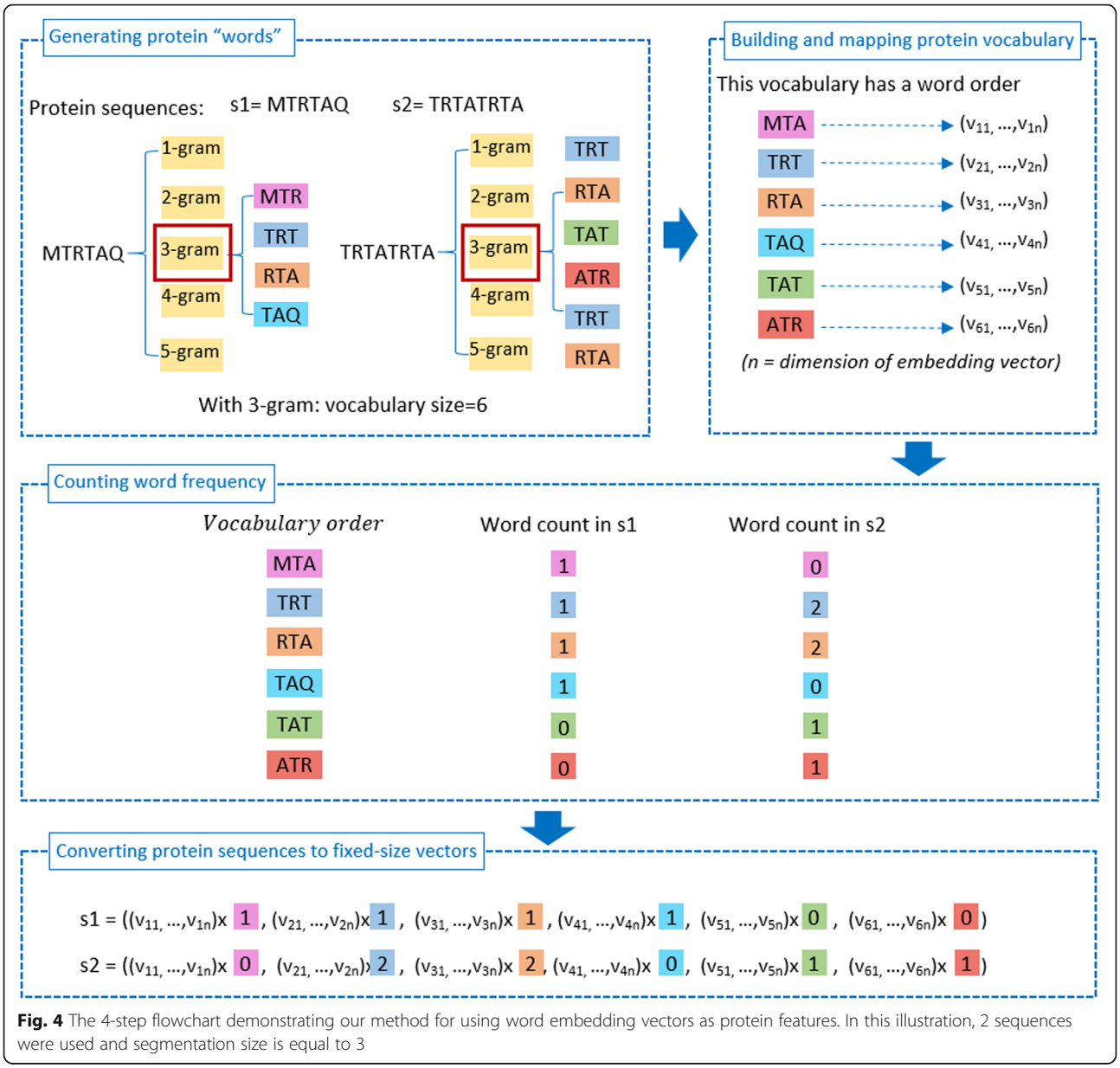

Fig. 4 The 4-step flowchart demonstrating our method for using word embedding vectors as protein features. In this illustration, 2 sequences were used and segmentation size is equal to 3

enabled us to gain a lot of discriminatory features for more efficient prediction. The flow chart in Fig. 4 depicts our extraction technique for the case with a pair of sequences and biological word lengths equal to 3. In Additional file 3, we present a more detailed step-by-step demonstration of our feature extraction method with more sample sequences and selected n-gram length of 3. Moreover, rather than using fixed n-grams, with n are segmentation lengths of 1, 2, 3, 4 and 5, we conducted several mixtures among these grams, where we expected to find the best merge among different feature types to produce truly helpful features. The total number of these hybrid features is accumulated by surveyed grams.

Assessment of predictive ability

Our study is a binary classification problem. First, we employed a five-fold cross-validation technique to develop and evaluate our models during the training process. Next, we assessed the capacity of our models in predicting unseen data using the independent dataset. We used four widely-used metrics sensitivity (Sen), specificity (Spec), accuracy (Acc) and Matthews's correlation coefficient (MCC) (see, e.g. [56, 57]) to measure the predictive performance. In Additional file 4, the formulae of these metrics are presented. When a single metric is required to assess a predictive model globally, we used the area under the receiver operating characteristic (ROC) curves (AUC) score [58, 59].

Supplementary information

Supplementary information accompanies this paper at https://doi.org/10.1186/s12920-020-00779-w.

Additional file 1 Statistical test results. **Table S1**: F and F-critical values from F-test to test the null hypothesis that two datasets' variances are equal in terms of single amino acid, dipeptide and tripeptide composition. **Table S2**: *p*-values from the unpaired student T-test on single amino acid composition on positive group (18 TNF sequences) and negative group (non-TNF sequences) with the assumption that the variance are equal. **Table S3**: Top 20 dipeptides with the lowest *p*-values from the unpaired student T-test on dipeptide composition on positive group (18 TNF sequences) and negative group (non-TNF sequences) with the assumption that the variance are equal. **Table S4**: Top 20 tripeptides with the lowest *p*-values from the unpaired student T-test on tripeptide composition on positive group (18 TNF sequences) and negative group (non-TNF sequences) with the assumption that the variance are equal.

Additional file 2 Visualization of feature sets. **Fig. S1a**: Visualization corresponding to protein feature vectors comprised from 1-g embedding vectors with perplexity equal to 25. **Fig. S1b**: Visualization corresponding to protein feature vectors comprised from 1-g embedding vectors with perplexity equal to 50. **Fig. S2a**: Visualization corresponding to protein feature vectors comprised from 2-g embedding vectors with perplexity equal to 25. **Fig. S2b**: Visualization corresponding to protein feature vectors comprised from 2-g embedding vectors with perplexity equal to 50. **Fig. S3a**: Visualization corresponding to protein feature vectors comprised from 3-g embedding vectors with perplexity equal to 25. **Fig. S3b**: Visualization corresponding to protein feature vectors comprised from 3-g embedding vectors with perplexity equal to 50. **Fig. S4a**: Visualization corresponding to protein feature vectors comprised from 4-g embedding vectors with perplexity equal to 25. **Fig. S4b**: Visualization corresponding to protein feature vectors comprised from 4-g embedding vectors with perplexity equal to 50. **Fig. S5a**: Visualization corresponding to protein feature vectors comprised from 5-g embedding vectors with perplexity equal to 25. **Fig. S5b**: Visualization corresponding to protein feature vectors comprised from 5-g embedding vectors with perplexity equal to 50. **Fig. S6a**: Visualization corresponding to protein feature vectors comprised from 1-g and 2-g embedding combined vectors with perplexity equal to 25. **Fig. S6b**: Visualization corresponding to protein feature vectors comprised from 1-g and 2-g embedding combined vectors with perplexity equal to 50. **Fig. S7a**: Visualization corresponding to protein feature vectors comprised from 1-g and 3-g embedding combined vectors with perplexity equal to 25. **Fig. S7b**: Visualization corresponding to protein feature vectors comprised from the combination of 1-g and 3-g embedding combined vectors with perplexity equal to 50. **Fig. S8a**: Visualization corresponding to protein feature vectors comprised from 2-g and 3-g embedding combined vectors with perplexity equal to 25. **Fig. S8b**: Visualization corresponding to protein feature vectors comprised from the combination of 2-g and 3-g embedding combined vectors with perplexity equal to 50.

Additional file 3 The details of the feature extraction method with examples.

Additional file 4 Formulae of assessment metrics.

Additional file 5 The surveyed dataset.

Abbreviations

AAC: Amino acid composition; Acc: Accuracy; AUC: Area under the curve; DPC: Dipeptide composition; IFN: Interferon; IL: Interleukin; kNN: K-nearest neighbor; MCC: Matthew's correlation coefficient; NGF: Neuron growth factor; NLP: Natural language processing; PSSM: Position specific scoring matrix; RF: Random forest; ROC: Receiver operating characteristic; Sen: Sensitivity; Spec: Specificity; SVM: Support vector machine; TGF-b: Transforminggrowth factor b; TNF: Tumor necrosis factor; TPC: Tripeptide composition; t-SNE: T-distributed stochastic neighbor embedding

Acknowledgements

We are very grateful to the Ministry of Science and Technology, Taiwan, R.O.C. for their support and for providing the funding for this publication.

Preprint posting details

doi: https://doi.org/10.1101/860791

About this supplement

This article has been published as part of *BMC Medical Genomics Volume 13 Supplement 10, 2020: Selected articles from the 18th Asia Pacific Bioinformatics Conference (APBC 2020): medical genomics.* The full contents of the supplement are available online at https://bmcmedgenomics.biomedcentral.com/articles/supplements/volume-13-supplement-10 .

Authors' contributions

Conceived and designed the study: TTDN LNQK YYO Analyzed the data: TTDN DVP. Designed and performed the experiments: TTDN HQT. Wrote the paper: TTDN NQKL HQT. All authors have read and approved this manuscript.

Funding

Publication costs are funded by Ministry of Science and Technology, Taiwan, R.O.C. under Grant no. MOST 108–2221-E-155-040. The funding agency played no part in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Availability of data and materials

The dataset supporting the conclusions of this article is included in the additional file 5.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Department of Computer Science and Engineering, Yuan Ze University, Taoyuan 32003, Taiwan. [2]Professional Master Program in Artificial Intelligence in Medicine, College of Medicine, Taipei Medical University, Taipei City 106, Taiwan. [3]Research Center for Artificial Intelligence in Medicine, Taipei Medical University, Taipei City 106, Taiwan. [4]University of Economics, The University of Danang, Danang 550000, Vietnam.

Published: 22 October 2020

References

1. Benveniste EN. Cytokine actions in the central nervous system. Cytokine Growth Factor Rev. 1998;9(3–4):259–75.
2. Aggarwal BB. Signalling pathways of the TNF superfamily: a double-edged sword. Nat Rev Immunol. 2003;3(9):745.
3. Wang X, Lin Y. Tumor necrosis factor and cancer, buddies or foes? 1. Acta Pharmacol Sin. 2008;29(11):1275–88.
4. Yi F, Frazzette N, Cruz AC, Klebanoff CA, Siegel RM. Beyond cell death: new functions for TNF family cytokines in autoimmunity and tumor immunotherapy. Trends Mol Med. 2018;24(7):642–53.
5. Guerrini MM, Okamoto K, Komatsu N, Sawa S, Danks L, Penninger JM, Nakashima T, Takayanagi H. Inhibition of the TNF family cytokine RANKL prevents autoimmune inflammation in the central nervous system. Immunity. 2015;43(6):1174–85.
6. Aggarwal BB, Shishodia S, Ashikawa K, Bharti AC. The role of TNF and its family members in inflammation and cancer: lessons from gene deletion. Curr Drug Targets-Inflamm Allergy. 2002;1(4):327–41.
7. Brennan FM, McInnes IB. Evidence that cytokines play a role in rheumatoid arthritis. J Clin Invest. 2008;118(11):3537–45.
8. Smith KA, Griffin JD. Following the cytokine signaling pathway to leukemogenesis: a chronology. J Clin Invest. 2008;118(11):3564–73.
9. Feldmann M. Many cytokines are very useful therapeutic targets in disease. J Clin Invest. 2008;118(11):3533–6.
10. Steinman L. Nuanced roles of cytokines in three major human brain disorders. J Clin Invest. 2008;118(11):3557–63.

11. Barnes PJ. The cytokine network in asthma and chronic obstructive pulmonary disease. J Clin Invest. 2008;118(11):3546–56.
12. Di Paolo NC, Shafiani S, Day T, Papayannopoulou T, Russell DW, Iwakura Y, Sherman D, Urdahl K, Shayakhmetov DM. Interdependence between interleukin-1 and tumor necrosis factor regulates TNF-dependent control of mycobacterium tuberculosis infection. Immunity. 2015;43(6): 1125–36.
13. Yarilina A, Ivashkiv LB. Type I interferon: a new player in TNF signaling, TNF Pathophysiology, vol. 11. Basel: Karger Publishers; 2010. p. 94–104.
14. Zou Q, et al. An approach for identifying cytokines based on a novel ensemble classifier. Biomed Res Int. 2013;2013:686090.
15. Huang N, Chen H, Sun Z. CTKPred: an SVM-based method for the prediction and classification of the cytokine superfamily. Protein Eng Des Sel. 2005; 18(8):365–8.
16. Lata S, Raghava G. CytoPred: a server for prediction and classification of cytokines. Protein Eng Des Sel. 2008;21(4):279–82.
17. Zeng X, Yuan S, Huang X, Zou Q. Identification of cytokine via an improved genetic algorithm. Front Comput Sci. 2015;9(4):643–51.
18. Yang Z, Wang J, Zheng Z, Bai X. A new method for recognizing cytokines based on feature combination and a support vector machine classifier. Molecules. 2018;23(8):2008.
19. He W, Jiang Z, Li Z. Predicting cytokines based on dipeptide and length feature. In: International Conference on Intelligent Computing: 2008. Basel: Springer; 2008. p. 86–91.
20. Jiang L, Liao Z, Su R, Wei L. Improved identification of cytokines using feature selection techniques. Lett Org Chem. 2017;14(9):632–41.
21. Joachims T. Text categorization with support vector machines: Learning with many relevant features. In: European conference on machine learning: 1998. Basel: Springer; 1998.p. 137–42.
22. Chou K-C. Some remarks on protein attribute prediction and pseudo amino acid composition. J Theor Biol. 2011;273(1):236–47.
23. Zeng Z, Shi H, Wu Y, Hong Z. Survey of natural language processing techniques in bioinformatics. Comput Mathl Methods Med. 2015;2015.
24. Ganguly D, Roy D, Mitra M, Jones GJ. Word embedding based generalized language model for information retrieval. In: Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval: 2015. Basel: ACM; 2015. p. 795–8.
25. Zhou G, He T, Zhao J, Hu P: **Learning continuous word embedding with metadata for question retrieval in community question answering**. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers): 2015*; 2015: 250–259.
26. Tang D, Wei F, Yang N, Zhou M, Liu T, Qin B: **Learning sentiment-specific word embedding for twitter sentiment classification**. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers): 2014*; 2014: 1555–1565.
27. Xing C, Wang D, Liu C, Lin Y: **Normalized word embedding and orthogonal transform for bilingual word translation**. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: 2015*; 2015: 1006–1011.
28. Le NQK. iN6-methylat (5-step): identifying DNA N 6-methyladenine sites in rice genome using continuous bag of nucleobases via Chou's 5-step rule. Mol Gen Genomics. 2019:1–10.
29. Song J, Li F, Takemoto K, Haffari G, Akutsu T, Chou K-C, Webb GI. PREvalL, an integrative approach for inferring catalytic residues using sequence, structural, and network features in a machine-learning framework. J Theor Biol. 2018;443:125–37.
30. Butt AH, Rasool N, Khan YD. Predicting membrane proteins and their types by extracting various sequence features into Chou's general PseAAC. Mol Biol Rep. 2018;45(6):2295–306.
31. Cheng X, Xiao X, Chou K-C. pLoc-mEuk: predict subcellular localization of multi-label eukaryotic proteins by extracting the key GO information into general PseAAC. Genomics. 2018;110(1):50–8.
32. Qiu W-R, Jiang S-Y, Xu Z-C, Xiao X, Chou K-C. iRNAm5C-PseDNC: identifying RNA 5-methylcytosine sites by incorporating physical-chemical properties into pseudo dinucleotide composition. Oncotarget. 2017;8(25):41178.
33. Jia J, Li X, Qiu W, Xiao X, Chou K-C. iPPI-PseAAC (CGR): identify protein-protein interactions by incorporating chaos game representation into PseAAC. J Theor Biol. 2019;460:195–203.
34. Cai L, Huang T, Su J, Zhang X, Chen W, Zhang F, He L, Chou K-C. Implications of newly identified brain eQTL genes and their interactors in schizophrenia. Mol Ther-Nucleic Acids. 2018;12:433–42.
35. Le NQK, Yapp EKY, Ou Y-Y, Yeh H-Y. iMotor-CNN: identifying molecular functions of cytoskeleton motor proteins using 2D convolutional neural network via Chou's 5-step rule. Anal Biochem. 2019;575:17–26.
36. Le NQK, Yapp EKY, Ho Q-T, Nagasundaram N, Ou Y-Y, Yeh H-Y. iEnhancer-5Step: identifying enhancers using hidden information of DNA sequences via Chou's 5-step rule and word embedding. Anal Biochem. 2019;571:53–61.
37. Kusuma RMI, Ou Y-Y. Prediction of ATP-binding sites in membrane proteins using a two-dimensional convolutional neural network. J Mol Graph Model. 2019.
38. Lvd M, Hinton G. Visualizing data using t-SNE. J Mach Learn Res. 2008; 9(Nov):2579–605.
39. Scholkopf B, Smola AJ. Learning with kernels: support vector machines, regularization, optimization, and beyond. Basel: MIT press; 2001.
40. Weinberger KQ, Saul LK. Distance metric learning for large margin nearest neighbor classification. J Mach Learn Res. 2009;10(2).
41. Liaw A, Wiener M. Classification and regression by randomForest. R news. 2002;2(3):18–22.
42. McCallum A, Nigam K. A comparison of event models for naive bayes text classification. In: AAAI-98 workshop on learning for text categorization: 1998. Basel: Citeseer; 1998. p. 41–8.
43. Ou Y: QuickRBF: a package for efficient radial basis function networks. QuickRBF software. In.; 2005.
44. Ou Y-Y. Prediction of FAD binding sites in electron transport proteins according to efficient radial basis function networks and significant amino acid pairs. BMC Bioinformatics. 2016;17(1):298.
45. Ou Y-Y. Identifying the molecular functions of electron transport proteins using radial basis function networks and biochemical properties. J Mol Graph Model. 2017;73:166–78.
46. Ou Y-Y. Incorporating efficient radial basis function networks and significant amino acid pairs for predicting GTP binding sites in transport proteins. BMC Bioinformatics. 2016;17(19):501.
47. Akbik A, Blythe D, Vollgraf R: **Contextual string embeddings for sequence labeling**. In: *Proceedings of the 27th International Conference on Computational Linguistics: 2018*; 2018: 1638-1649.
48. Salant S, Berant J: **Contextualized word representations for reading comprehension**. *arXiv preprint arXiv:171203609* 2017.
49. Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L: **Deep contextualized word representations**. *arXiv preprint arXiv:180205365* 2018.
50. Pan SJ, Yang Q. A survey on transfer learning. IEEE Trans Knowl Data Eng. 2009;22(10):1345–59.
51. Apweiler R, Bairoch A, Wu CH, Barker WC, Boeckmann B, Ferro S, Gasteiger E, Huang H, Lopez R, Magrane M. UniProt: the universal protein knowledgebase. Nucleic Acids Res. 2004;32(suppl_1):D115–9.
52. Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 1997;25(17):3389–402.
53. Bojanowski P, Grave E, Joulin A, Mikolov T: **Enriching word vectors with subword information**. *arXiv preprint arXiv:160704606* 2016.
54. Joulin A, Grave E, Bojanowski P, Mikolov T: **Bag of tricks for efficient text classification**. *arXiv preprint arXiv:160701759* 2016.
55. Ben-Hur A, Brutlag D. Sequence motifs: highly predictive features of protein function. In: Feature extraction. Basel: Springer; 2006. p. 625–45.
56. Taju SW, Nguyen TTD, Le NQK, Kusuma RMI, Ou YY. DeepEfflux: a 2D convolutional neural network model for identifying families of efflux proteins in transporters. Bioinformatics. 2018;34(18):3111–7.
57. Ho Q-T, Phan D-V, Ou Y-Y. Using word embedding technique to efficiently represent protein sequences for identifying substrate specificities of transporters. Anal Biochem. 2019;577:73–81.
58. Fawcett T. An introduction to ROC analysis. Pattern Recogn Lett. 2006;27(8): 861–74.
59. Davis J, Goadrich M. The relationship between Precision-Recall and ROC curves. In: Proceedings of the 23rd international conference on Machine learning: 2006. Basel: ACM; 2006. p. 233–40.

Publisher's Note