Identifying SNAREs by Incorporating Deep Learning Architecture and Amino Acid Embedding Representation

Nguyen Quoc Khanh Le[1*†] *and Tuan-Tu Huynh*[2,3*]

[1] Professional Master Program in Artificial Intelligence in Medicine, Taipei Medical University, Taipei, Taiwan, [2] Department of Electrical Electronic and Mechanical Engineering, Lac Hong University, Bien Hoa, Vietnam, [3] Department of Electrical Engineering, Yuan Ze University, Taoyuan, Taiwan

OPEN ACCESS

Edited by:
Panayiotis V. Benos,
University of Pittsburgh, United States

Reviewed by:
Litao Sun,
The Scripps Research Institute,
United States
Alexey Goltsov,
Abertay University, United Kingdom

***Correspondence:**
Nguyen Quoc Khanh Le
khanhlee@tmu.edu.tw;
khanhlee87@gmail.com
Tuan-Tu Huynh
huynhtuantu@lhu.edu.vn

†*Present address:*
Nguyen Quoc Khanh Le,
Research Center for Artificial
Intelligence in Medicine, Taipei
Medical University, Taipei, Taiwan

Specialty section:
This article was submitted to
Systems Biology,
a section of the journal
Frontiers in Physiology

Received: *28 March 2019*
Accepted: *26 November 2019*
Published: *10 December 2019*

Citation:
Le NQK and Huynh T-T (2019)
Identifying SNAREs by Incorporating
Deep Learning Architecture
and Amino Acid Embedding
Representation.
Front. Physiol. 10:1501.
doi: 10.3389/fphys.2019.01501

SNAREs (soluble N-ethylmaleimide-sensitive factor activating protein receptors) are a group of proteins that are crucial for membrane fusion and exocytosis of neurotransmitters from the cell. They play an important role in a broad range of cell processes, including cell growth, cytokinesis, and synaptic transmission, to promote cell membrane integration in eukaryotes. Many studies determined that SNARE proteins have been associated with a lot of human diseases, especially in cancer. Therefore, identifying their functions is a challenging problem for scientists to better understand the cancer disease as well as design the drug targets for treatment. We described each protein sequence based on the amino acid embeddings using fastText, which is a natural language processing model performing well in its field. Because each protein sequence is similar to a sentence with different words, applying language model into protein sequence is challenging and promising. After generating, the amino acid embedding features were fed into a deep learning algorithm for prediction. Our model which combines fastText model and deep convolutional neural networks could identify SNARE proteins with an independent test accuracy of 92.8%, sensitivity of 88.5%, specificity of 97%, and Matthews correlation coefficient (MCC) of 0.86. Our performance results were superior to the state-of-the-art predictor (SNARE-CNN). We suggest this study as a reliable method for biologists for SNARE identification and it serves a basis for applying fastText word embedding model into bioinformatics, especially in protein sequencing prediction.

Keywords: SNARE proteins, deep learning, convolutional neural networks, word embedding, skip-gram

INTRODUCTION

Soluble N-ethylmaleimide-sensitive factor activating protein receptors (SNAREs) are the most important and broadly studied proteins in membrane fusion, trafficking, and docking. They are membrane-associated proteins that consist of distinguishing SNARE domains: heptad restates ∼60 amino acids in length that are predicted to assemble coiled-coils (Duman and Forte, 2003). Most SNAREs consist of only one SNARE motif adjacent to a single C-terminal membrane (e.g., syntaxin 1 and synaptobrevin 2). **Figure 1** shows the domain architecture of some example

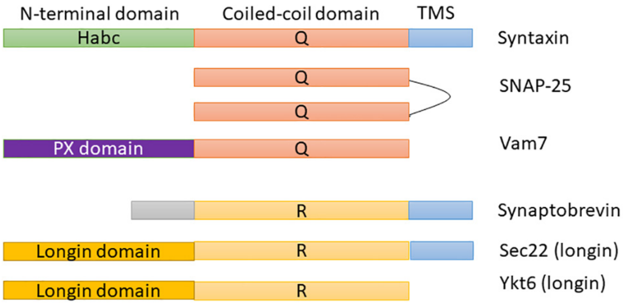

FIGURE 1 | Domain architecture model of SNARE proteins.

SNAREs (e.g., syntaxin, SNAP-25, or Vam 7). As shown in these proteins, SNAREs generally consist of a central "SNARE domain" that is flanked by a variable N-terminal domain and a C-terminal single α-helical transmembrane anchor (Ungermann and Langosch, 2005). SNARE proteins are crucial for a broad range of cell processes, e.g., cytokinesis, synaptic transmission, and cell growth, to promote cell membrane integration in eukaryotes (Jahn and Scheller, 2006; Wickner and Schekman, 2008). There are two categories of SNARE: v-SNAREs incorporated into the membranes of transport vesicles during budding, and t-SNAREs associated with nerve terminal membranes. Researchers have recently identified a lot of SNARE proteins in human and they demonstrated that there is a crucial link between SNARE proteins and numerous diseases [e.g., neurodegenerative (Hou et al., 2017), mental illness (Dwork et al., 2002), and especially cancer (Meng and Wang, 2015; Sun et al., 2016)]. As a detail, a 1 bp deletion in SNAP-29 causes a novel neurocutaneous syndrome (Sprecher et al., 2005), mutation in the b-isoform of neuronal SNARE synaptosomal-associated protein of 25 kDa (SNAP-25) results in both diabetes and psychiatric disease (Jeans et al., 2007), mutations in VPS33B cause arthrogryposis–renal dysfunction–cholestasis (ARC) syndrome (Gissen et al., 2004), and so on.

Because SNARE proteins play an essential molecular function in cell biology, a wide variety of techniques were presented and used to investigate them. One of the best studies on SNAREs is molecular docking of synaptic vesicles with the presynaptic membrane in neurons. Another solution is to identify SNAREs from unknown sequence according to their motif information. In order to address it, Kloepper team is a first group that used bioinformatics techniques in this kind of problem. In their research, they have already built a database for retrieving and classifying SNARE proteins (Kloepper et al., 2007, 2008; Kienle et al., 2009). Furthermore, SNARE functions in sub-Golgi localization had also been predicted using bioinformatics techniques (van Dijk et al., 2008). Yoshizawa et al. (2006) identified SNAREs in membrane trafficking via extracting sequence motifs and the phylogenetic features. In the latest work, Le and Nguyen (2019) identified SNAREs by treating position-specific scoring matrices as images to feed into 2D convolutional neural network (CNN).

To our knowledge, only the study from Le and Nguyen (2019) conducted the SNARE protein prediction in membrane fusion by using machine learning techniques. However, their performance results need a lot of improvements, and we therefore motivate to create a better model for this. To address this, we transform the protein sequences into a continuous bag of nucleobases using fastText model (Bojanowski et al., 2017) and then carry out to identify them with the use of deep neural networks. Releasing by Facebook Research, fastText is a natural language processing (NLP) model for word embedding and text classification. It uses neural network for learning text representations and since its discovery, it has been used in a lot of different NLP problems (Joulin et al., 2017). It has been also used in interpreting biological sequences such as DNA sequences (Le, 2019; Le et al., 2019b) and protein sequences (Asgari et al., 2019), and here we provide a different application with a more in-depth analysis.

The idea is to treat protein sequence as a sentence and amino acids as words, we used fastText to train the language model on all sequences. Subsequently, this language model will be used to generate vectors for protein sequences. At the latest stage, we used a deep neural network to learn these vectors as features and perform supervised learning for classification. The rest of this paper is organized as follows: our materials and methods are introduced in the section "Methods"; some of our relevant experiments and results are introduced in the section "Results"; discussions of the model performance as well as limitations are given in the section "Discussion."

METHODS

Figure 2 illustrates our flowchart which consists of three major processes: data collection, training fastText model and 1D CNN model. We describe the detailed description of our approach in the following paragraphs.

Data Collection

The dataset retrieved from the National Center for Biotechnology Information (NCBI) (by 4-2-2019) (Coordinators, 2015), which is a large suite of online resources for biological information and data. Moreover, on-line resource conserved domain database (CDD) (Zheng et al., 2014) suggested that "SNARE superfamily" members could be identified using the SNARE motif "cl22856," therefore, we used this information to generate non-redundant (annotated) SNARE proteins. This step ensures that we collected all corrected SNARE proteins including SNARE motif. There are many protein sources in NCBI, and we chose to collect all protein sequences from RefSeq (Pruitt et al., 2006). Next, to prevent overfitting problem, we used CD-HIT (Fu et al., 2012) to eliminate the redundant sequences with similarity greater than 30%, and the rest of proteins reaches 26,789 SNAREs. We used full sequences of proteins, thus it includes typical coiled coil as well as other motifs.

In the next step, we collected a negative set to treat our problem as a binary classification between positive (SNAREs) and negative set. To perform this, we retrieved all general proteins without the SNARE motif and with similarity more

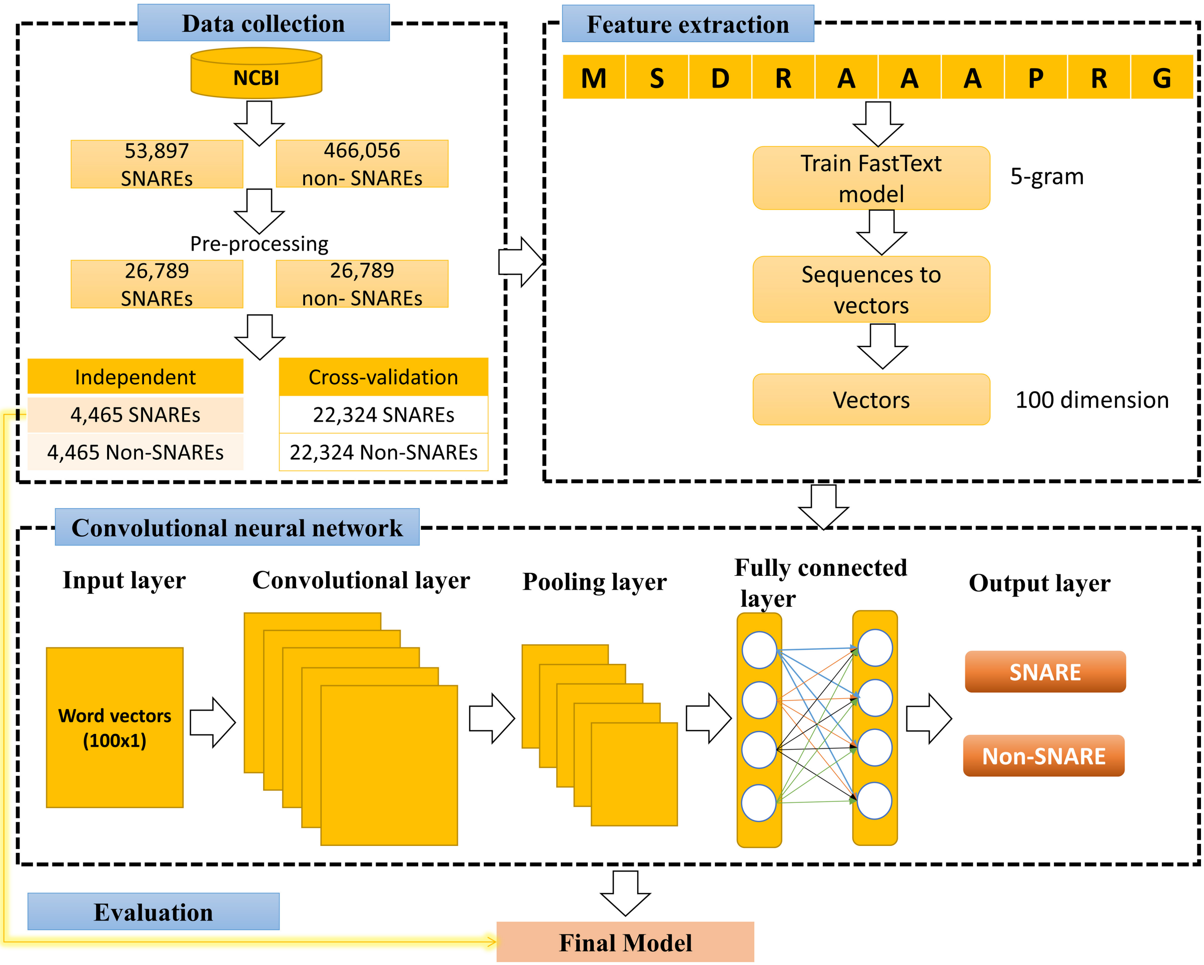

FIGURE 2 | Flow chart of this study.

than 30%. Because the number of negative data was much higher than the number of positive data, it will cause difficulties in machine learning problem. Therefore, we randomly selected 26,789 negative samples to give balance training in our problem.

Amino Acid Embedding Representation

Encouraged by the high performance of word embedding in many NLP tasks, we presented a similar feature set called "amino acid embedding." The objective is to apply recent NLP models into biological sequences. It was first proposed by Asgari and Mofrad (2015) and successfully used to solve the latter biological problems related to sequence information (Habibi et al., 2017; Vang and Xie, 2017; Öztürk et al., 2018). Nevertheless, with the use of Word2Vector to describe the biological sequences, these findings had some disadvantages such as out-of-vocabulary cases for unknown words as well as not taking care of the inner structure of words. Accordingly, a critical issue therefore needs to be resolved is that instead of using an single specific vector representation for the protein word, the internal structure of each word needs to be taken into account. Facebook suggested fastText, which is a Word2vec extension that can handle the word as a continuous bag of character n-grams (Bojanowski et al., 2017), to perform this task. The vector for a word therefore consists of the number of n-grams of this type. It has been shown that fastText was more accurate than using Word2vec in a variety of fields (Joulin et al., 2017). Inspired by its accomplishments, previous researchers used it to describe biological sequences such as DNA enhancer sequence (Le et al., 2019b), DNA N6-methyladenine sites (Le, 2019) and protein sequence (Asgari et al., 2019).

The goal of this step is to encode nucleotides by establishing their vector space distribution, enabling them to be adopted by supervised learning algorithms. To perform a supervised learning classification, we need a set of features having the same dimension. Nonetheless, our protein sequences are of different lengths, so to address this issue, we set the embedding vector dimension to 100. This means that each protein sequence is represented as real numerical values of 100 and can be fed directly without pre-processing into any machine learning classifier. We have more special features for a good prediction by bringing this information into the dataset.

Convolutional Neural Network

Convolutional neural network generally consists of multiple layers with each layer performing a particular function of translating its output into a functional representation. All layers are combined to form the architecture of our CNN system using a specific order. Similar to many published works in this field (Le et al., 2018, 2019a,c; Nguyen et al., 2019), different layers used in CNN for the current study include:

(1) Input layer of our CNN is a 1D vector, which is a vector of size 1 × 100 (created by fastText model).

(2) Convolutional layers were used as convolution operations to extract features embedded in the 1D input vector. These layers took a sliding window with specific stride shifting across all the input shapes. After sliding, the input shapes will be transformed into representative values. The spatial relationship between numeric values in the vectors has been preserved in this convolutional process. It will help this layer learn the important features using small slides of input data. Since the input of our CNN model is a vector of small size, we used the kernel size of 3 to deduce more information. This number of kernel has been used in previous works on CNN (Le et al., 2017, 2018, 2019a).

(3) Activation layer was performed after convolutional layers. It is an additional non-linear operation, called ReLU (Rectified Linear Unit) and is calculated as follows:

$$f(x) = \max(0, x) \tag{1}$$

Where x is the number of inputs in a neural network. The purpose of ReLU is to introduce non-linearity in our CNN and help our model learn better from the data.

(4) Pooling layer was applied in convolutional layers to reduce the computational size for the next layers. There are three types of pooling layers, and we selected max pooling in our architecture to select the maximum value over a window of 2.

(5) Dropout layer was applied aiming to reduce the overfitting of our model and also to improve the performance results in some cases (Srivastava et al., 2014).

(6) Flatten layer was used to transform the input matrix into a vector. It always stand before fully connected layers.

(7) Fully connected layer was usually applied in the last stages of neural network architectures. In this layer, each node is fully connected with all the nodes of the previous layers. Two fully connected layers have been included in the current model. The first one connected all the input nodes to the flatten layer to help our model to gain more knowledge and perform better. This one was then connected to the output layer by the second layer. The number of nodes in the output layer is equal to 2 as identifying SNARE proteins was as a binary classification problem.

(8) Softmax was an evaluation function standing at the output of the model to determine the probability of each possible output. Its function could be calculated by the formula:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}} \tag{2}$$

where z indicates the input vector with K-dimensional vector, σ(z)i is real values in the range (0, 1) and ith class is the predicted probability from sample vector x.

Assessment of Predictive Ability

We firstly trained the model on the entire training set using 5-fold cross-validation technique. Since every 5-fold cross-validation produces different results each time, we performed ten times 5-fold cross-validation to achieve more reliable results. Thereafter, we reported the cross-validation performance by averaging all the ten times cross-validation tests. In the training process, hyper-parameter optimization has been used to identify the best parameters for each dataset. Finally, an independent test was applied to evaluate the performance and to ensure preventing any systematic bias in the cross-validation set.

Moreover, to evaluate the performance of our method, we applied Chou's criterion (Chou, 2001) used in many bioinformatics studies. With this criterion, some standard metrics sensitivity, specificity, accuracy and Matthews correlation coefficient (MCC) are as follows:

$$Sensitivity = 1 - \frac{N_{-}^{+}}{N^{+}}, \quad 0 \le Sen \le 1 \tag{3}$$

$$Specificity = 1 - \frac{N_{+}^{-}}{N^{-}}, \quad 0 \le Spec \le 1 \tag{4}$$

$$Accuracy = 1 - \frac{N_{-}^{+} + N_{+}^{-}}{N^{+} + N^{-}}, \quad 0 \le Acc \le 1 \tag{5}$$

$$MCC = \frac{1 - \left(\frac{N_{-}^{+}}{N^{+}} + \frac{N_{+}^{-}}{N^{-}}\right)}{\sqrt{\left(1 + \frac{N_{+}^{-} - N_{-}^{+}}{N^{+}}\right)\left(1 + \frac{N_{-}^{+} - N_{+}^{-}}{N^{-}}\right)}},$$

$$-1 \le MCC \le 1 \tag{6}$$

The relations between these symbols and the symbols in Eqs. (3–6) are given by:

$$\begin{cases} N_{+}^{-} = FP \\ N_{-}^{+} = FN \\ N^{+} = TP + N_{-}^{+} \\ N^{-} = TN + N_{+}^{-} \end{cases} \tag{7}$$

Where TP, FP, TN, FN are true positive, false positive, true negative, and false negative values, respectively.

RESULTS

Composition of Amino Acid Representation in SNAREs and Non-SNAREs

In this section, we would like to analyze the differences between SNARE and non-SNARE sequences in our dataset by computing the composition of amino acid representation between them. The amino acids which had the highest frequency in the positive and negative set are shown in **Figure 3**. It is easy to point out some of the differences between the two types of dataset. For instance, we were aware of the higher frequency of amino acid L, and F, and R in the SNARE proteins but lower in the non-SNAREs. Otherwise, the amino acids that appeared a lot in non-SNARE sequences are G, T, N, and D. Besides, we plotted the standard error bars at each column to statistically see the differences among amino acid compositions. These error bars aim to calculate confidence intervals, or margins of error to quantify uncertainty. As shown in **Figure 3**, there are some amino acids had significantly differences (with no overlap error bars) such as N, D, G, L, F, and T. Therefore, these amino acids might play a crucial role in identifying SNARE sequences and they can be special features that help our model predict SNAREs with high accuracy. This finding also plays an important role in further research that aims to analyze the motif information in SNARE proteins.

Hyperparameters Optimization

Hyper-parameters are architecture-level parameters and are different from parameters of a model trained via backpropagation. To tune hyperparameters, we used the approach to choose a set of hyperparameters for speeding up the training process as well as preventing overfitting. As suggested by Chollet (2015), each step of the above hyper-parameter-tuning approach was integrated into the hyper-parameter-tuning process as follows:

- Selecting a specific set of hyper-parameters.
- Creating the model according to the specific set.
- Evaluating the performance results using testing dataset.
- Moving to the next set of hyper-parameters.
- Repeating.
- Measuring performance results on an independent dataset.

Keras framework library (Chollet, 2015) with a TensorFlow backend (Abadi et al., 2016) was used as a deep learning framework to build the 1D CNN architecture. We performed grid search on training set and used accuracy to select the next set of hyperparameters. Furthermore among the six optimizers

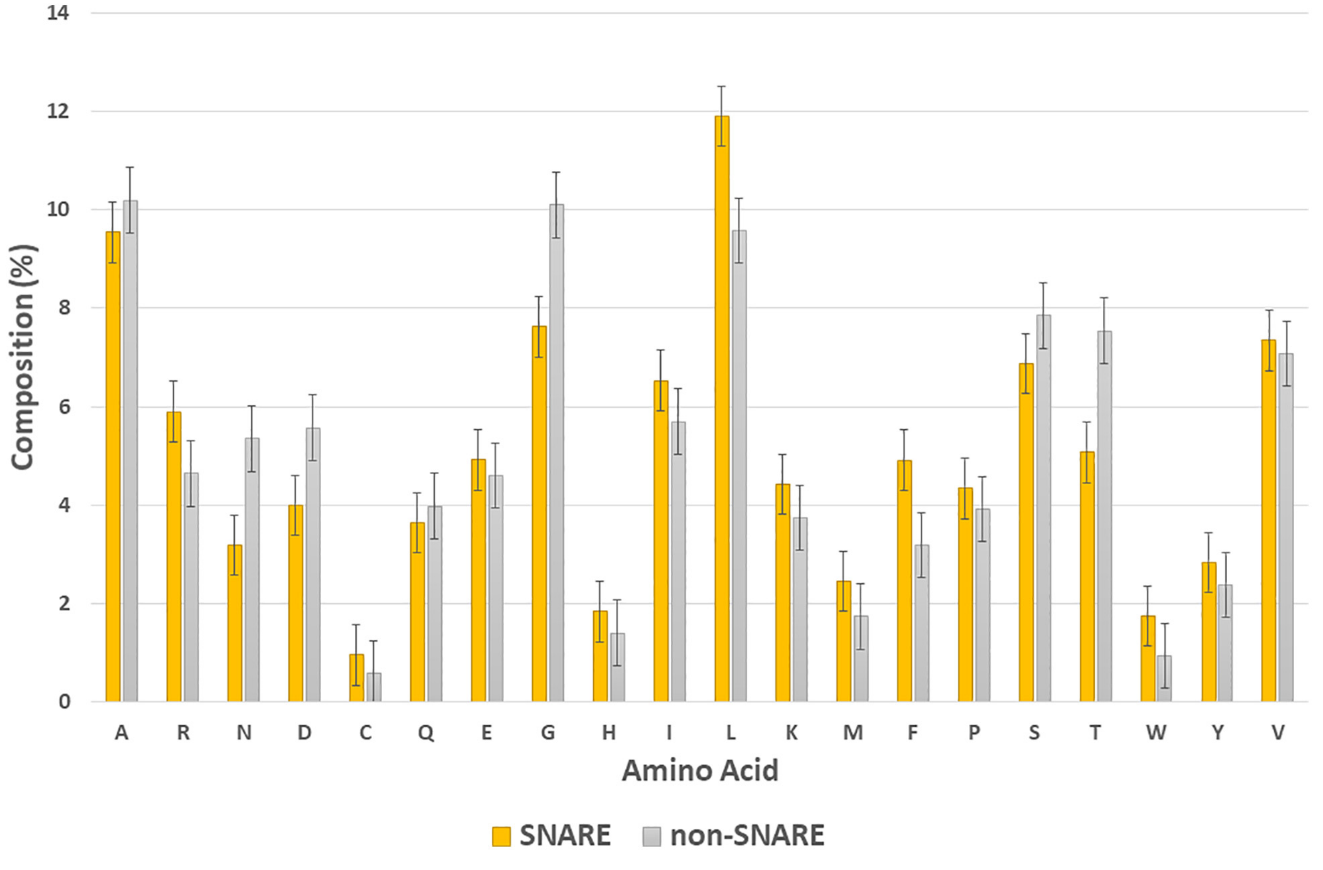

FIGURE 3 | Composition of amino acid in SNAREs and non-SNAREs.

in Keras [e.g., Adam, Adadelta, Adagrad, Stochastic Gradient Descent (SGD), RMSprop, and Adamax], Adadelta has given a superior performance. Therefore, we used Adadelta in our model to achieve an optimal result. This point is also proven in the previous protein function prediction using CNN (Le et al., 2017; Nguyen et al., 2019).

SNARE Identification With Different n-Gram Levels

After tuning the optimal parameters for 1D CNN model, we evaluated the performance of this architecture on the datasets of different n-gram levels (from 1 to 5). In this step, all the measurement metrics were used to evaluate the comparative performance in both cross-validation and independent test. The result is displayed in **Table 1**. **Table 1** shows that the performance results of n-gram levels are proportional. We were not able to achieve the best performance unless we used high levels of n-gram values. To maximize the performance of our models, we should choose the n-gram levels from 4 (accuracy of more than 97%). This means that the model only captures the special information in a high level of n-gram, increasing high level of n-gram will help to increase much in the results. In this study, we chose n-gram = 5 with the best metrics (accuracy of 97.5 and 92.8% in the cross-validation and independent test, respectively) to perform further experiments.

In most of the supervised learning problems, our model can perform well during training test, but worse in another invisible data. This is called overfitting and our study, no exception also included in this issue. Therefore, an independent test was used in our study to ensure that our model also works well in a blind dataset with unseen data. As described in the previous part, our independent dataset contained 4,465 SNAREs and 4,465 non-SNAREs. None of these samples occur in the training set. As shown in **Table 1**, our independent testing results also comply with cross-validation results in most metrics. To detail, our independent testing performance achieved the accuracy of 92.8%, sensitivity of 88.5%, specificity of 97%, and MCC of 0.86. There is a very few overfitting in our model and it can demonstrate that our model has been well done in this type of dataset. Another reason is the use of dropout inside CNN structure and it helps us prevent overfitting.

Comparative Performance Between Proposed Method and the Existing Methods

From the previous section, we chose the combination of 1D CNN and 5-gram as our optimal model for SNARE identification. In this section, we aim to compare the effectiveness of our proposed features with other research groups studying the same problem. As mentioned in the literature review, there have been some published works on identifying SNARE proteins using computational techniques. However, among of them, there is only one predictor to propose the machine learning techniques on predicting SNARE (Le and Nguyen, 2019). Therefore, we compared our performance with them in both cross-validation and independent test. **Table 2** shows the performance results by highlighting the higher values for each metrics. It is clear that on average, our method outperforms the previous model in all measurement metrics. Therefore, we are able to generate effective features for identifying SNAREs with a better performance than PSSM profiles which had been used in the previous work.

DISCUSSION

Based on the outstanding results of word embeddings in NLP, applying it to protein function prediction is an essential concern for biological researchers. In this study, we have approached a method using word embedding and deep learning for identifying SNARE proteins. Our structure is a combination between fastText (to train vectors model) and 1D CNN (to train deep learning model from the generated vectors). By using fastText, the protein sequences have been interpreted via different

TABLE 1 | Performance results on identifying SNAREs with different n-gram levels.

	Cross validation				Independent			
n-gram	**Sens**	**Spec**	**Acc**	**MCC**	**Sens**	**Spec**	**Acc**	**MCC**
1	83.8	88.7	86.3	0.73	39.4	94.6	67	0.41
2	93.7	91.6	92.6	0.85	83.1	87.4	85.2	0.71
3	95.8	97.6	96.7	0.93	87.4	95	91.2	0.83
4	96.7	98.1	97.4	0.95	88.7	96.4	92.6	0.85
5	96.6	98.4	97.5	0.95	88.5	97	92.8	0.86

TABLE 2 | Comparative performance of predicting SNAREs between the proposed method and the previous published work.

	Cross validation				Independent			
Predictor	**Sens**	**Spec**	**Acc**	**MCC**	**Sens**	**Spec**	**Acc**	**MCC**
SNARE-CNN	76.6	93.5	89.7	0.7	65.8	90.3	87.9	0.46
Ours	**96.6**	**98.4**	**97.5**	**0.95**	**88.5**	**97**	**92.8**	**0.86**

The bold values is to show the significant values for each metric.

representations and we could generate the hidden information of them. While the other NLP models do not have subword information, it is an advantage of fastText that can help to improve this problem. Benefits of fastText when comparing to the other features have been also proven in the previous works based on their results (Do and Khanh Le, 2019; Le, 2019; Le et al., 2019b). We used 5-fold cross-validation set to train our model and an independent set to examine the performance results. Compared to the state-of-the-art predictor, our method produced superior performance in all the typical measurement metrics. Through this study, biologists can use our model to identify SNARE proteins with high accuracy and use them as necessary information for drug development. In addition, we contribute a method to interpret the information of protein sequences and further research is able to apply in bioinformatics research, especially in protein function prediction.

Furthermore, we provided our source codes and datasets at https://github.com/khanhlee/fastSNARE. The readers and biologists are able to reproduce our results as well as perform their classifications according our method. We also hope that our future research would be able to provide a web-server for the method of prediction as presented in this paper. Moreover, a limitation of using language model is that it could not consider mutations and SNPs in SNARE sequence. Therefore, further studies could integrate these information into fastText model to improve the predictive performance.

DATA AVAILABILITY STATEMENT

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

AUTHOR CONTRIBUTIONS

Both authors conceived the ideas, designed the study, participated in the discussion of the results and writing of the manuscript, and read and approved the final version of the manuscript. NL conducted the experiments and analyzed the results.

FUNDING

This work was supported in part by the Ministry of Science and Technology of the Republic of China under Grant MOST 109-2811-E-155-501.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of Nvidia Corporation with the donation of the Titan Xp GPU used for this research.

REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (eds) (2016). "Tensorflow: a system for large-scale machine learning," in *Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation*, Savannah, GA.

Asgari, E., McHardy, A. C., and Mofrad, M. R. K. (2019). Probabilistic variable-length segmentation of protein sequences for discriminative motif discovery (DiMotif) and sequence embedding (ProtVecX). *Sci. Rep.* 9:3577. doi: 10.1038/s41598-019-38746-w

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

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching Word Vectors with subword information. *Trans. Assoc. Comput. Linguist.* 5, 135–146. doi: 10.1162/tacl_a_00051

Chollet, F., (2015). *Keras*. Available at: https://github.com/fchollet/keras (accessed November 20, 2018).

Chou, K.-C. (2001). Using subsite coupling to predict signal peptides. *Protein Eng.* 14, 75–79. doi: 10.1093/protein/14.2.75

Coordinators, N. R. (2015). Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.* 44, D7–D19. doi: 10.1093/nar/gkv1290

Do, D. T., and Khanh Le, N. Q. (2019). A sequence-based approach for identifying recombination spots in *Saccharomyces cerevisiae* by using hyper-parameter optimization in fastText and support vector machine. *Chemometr. Intell. Lab. Syst.* 194:103855. doi: 10.1016/j.chemolab.2019.103855

Duman, J. G., and Forte, J. G. (2003). What is the role of SNARE proteins in membrane fusion? *Am. J. Physiol. Cell Physiol.* 285, C237–C249. doi: 10.1152/ajpcell.00091.2003

Dwork, A. J., Li, H.-Y., Mann, J. J., Xie, J., Hu, L., Falkai, P., et al. (2002). Abnormalities of SNARE mechanism proteins in anterior frontal cortex in severe mental illness. *Cereb. Cortex* 12, 349–356. doi: 10.1093/cercor/12.4.349

Fu, L., Niu, B., Zhu, Z., Wu, S., and Li, W. (2012). CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics* 28, 3150–3152. doi: 10.1093/bioinformatics/bts565

Gissen, P., Johnson, C. A., Morgan, N. V., Stapelbroek, J. M., Forshew, T., Cooper, W. N., et al. (2004). Mutations in VPS33B, encoding a regulator of SNARE-dependent membrane fusion, cause arthrogryposis–renal dysfunction–cholestasis (ARC) syndrome. *Nat. Genet.* 36, 400–404. doi: 10.1038/ng1325

Habibi, M., Weber, L., Neves, M., Wiegandt, D. L., and Leser, U. (2017). Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics* 33, i37–i48. doi: 10.1093/bioinformatics/btx228

Hou, C., Wang, Y., Liu, J., Wang, C., and Long, J. (2017). Neurodegenerative disease related proteins have negative effects on SNARE-Mediated membrane fusion in pathological confirmation. *Front. Mol. Neurosci.* 10:66. doi: 10.3389/fnmol.2017.00066

Jahn, R., and Scheller, R. H. (2006). SNAREs — engines for membrane fusion. *Nat. Rev. Mol. Cell Biol.* 7, 631–643. doi: 10.1038/nrm2002

Jeans, A. F., Oliver, P. L., Johnson, R., Capogna, M., Vikman, J., Molnár, Z., et al. (2007). A dominant mutation in Snap25 causes impaired vesicle trafficking, sensorimotor gating, and ataxia in the blind-drunk mouse. *Proc. Natl. Acad. Sci.U.S.A.* 104, 2431–2436. doi: 10.1073/pnas.0610222104

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (eds) (2017). "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2*, Valencia.

Kienle, N., Kloepper, T. H., and Fasshauer, D. (2009). Phylogeny of the SNARE vesicle fusion machinery yields insights into the conservation of the secretory pathway in fungi. *BMC Evol. Biol.* 9:19. doi: 10.1186/1471-2148-9-19

Kloepper, T. H., Kienle, C. N., and Fasshauer, D. (2008). SNAREing the basis of multicellularity: consequences of protein family expansion during evolution. *Mol. Biol. Evol.* 25, 2055–2068. doi: 10.1093/molbev/msn151

Kloepper, T. H., Kienle, C. N., Fasshauer, D., and Munro, S. (2007). An elaborate classification of SNARE proteins sheds light on the conservation of the eukaryotic endomembrane system. *Mol. Biol. Cell* 18, 3463–3471. doi: 10.1091/mbc.e07-03-0193

Le, N. Q. K. (2019). iN6-methylat (5-step): identifying DNA N6-methyladenine sites in rice genome using continuous bag of nucleobases via Chou's 5-step rule. *Mol. Genet. Genomics* 294, 1173–1182. doi: 10.1007/s00438-019-01570-y

Le, N. Q. K., Ho, Q. T., and Ou, Y. Y. (2017). Incorporating deep learning with convolutional neural networks and position specific scoring matrices for identifying electron transport proteins. *J. Comput. Chem.* 38, 2000–2006. doi: 10.1002/jcc.24842

Le, N. Q. K., Ho, Q.-T., and Ou, Y.-Y. (2018). Classifying the molecular functions of Rab GTPases in membrane trafficking using deep convolutional neural networks. *Anal. Biochem.* 555, 33–41. doi: 10.1016/j.ab.2018.06.011

Le, N. Q. K., Huynh, T.-T., Yapp, E. K. Y., and Yeh, H.-Y. (2019a). Identification of clathrin proteins by incorporating hyperparameter optimization in deep learning and PSSM profiles. *Comput. Methods Prog. Biomed.* 177, 81–88. doi: 10.1016/j.cmpb.2019.05.016

Le, N. Q. K., Yapp, E. K. Y., Ho, Q.-T., Nagasundaram, N., Ou, Y.-Y., and Yeh, H.-Y. (2019b). iEnhancer-5Step: identifying enhancers using hidden information of DNA sequences via Chou's 5-step rule and word embedding. *Anal. Biochem.* 571, 53–61. doi: 10.1016/j.ab.2019.02.017

Le, N. Q. K., Yapp, E. K. Y., Ou, Y.-Y., and Yeh, H.-Y. (2019c). iMotor-CNN: identifying molecular functions of cytoskeleton motor proteins using 2D convolutional neural network via Chou's 5-step rule. *Anal. Biochem.* 575, 17–26. doi: 10.1016/j.ab.2019.03.017

Le, N. Q. K., and Nguyen, V. N. (2019). SNARE-CNN: a 2D convolutional neural network architecture to identify SNARE proteins from high-throughput sequencing data. *PeerJ Comput. Sci.* 5:e177. doi: 10.7717/peerj-cs.177

Meng, J., and Wang, J. (2015). Role of SNARE proteins in tumourigenesis and their potential as targets for novel anti-cancer therapeutics. *Biochim. Biophys. Acta* 1856, 1–12. doi: 10.1016/j.bbcan.2015.04.002

Nguyen, T.-T.-D., Le, N.-Q.-K., Kusuma, R. M. I., and Ou, Y.-Y. (2019). Prediction of ATP-binding sites in membrane proteins using a two-dimensional convolutional neural network. *J. Mol. Graph. Model.* 92, 86–93. doi: 10.1016/j.jmgm.2019.07.003

Öztürk, H., Ozkirimli, E., and Özgür, A. (2018). A novel methodology on distributed representations of proteins using their interacting ligands. *Bioinformatics* 34, i295–i303. doi: 10.1093/bioinformatics/bty287

Pruitt, K. D., Tatusova, T., and Maglott, D. R. (2006). NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Res.* 35(Suppl._1), D61–D65. doi: 10.1093/nar/gkl842

Sprecher, E., Ishida-Yamamoto, A., Mizrahi-Koren, M., Rapaport, D., Goldsher, D., Indelman, M., et al. (2005). A mutation in SNAP29, coding for a SNARE protein involved in intracellular trafficking, causes a novel neurocutaneous syndrome characterized by cerebral dysgenesis, neuropathy, ichthyosis, and palmoplantar keratoderma. *Am. J. Hum. Genet.* 77, 242–251. doi: 10.1086/432556

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15, 1929–1958.

Sun, Q., Huang, X., Zhang, Q., Qu, J., Shen, Y., Wang, X., et al. (2016). SNAP23 promotes the malignant process of ovarian cancer. *J. Ovarian Res.* 9:80. doi: 10.1186/s13048-016-0289-289

Ungermann, C., and Langosch, D. (2005). Functions of SNAREs in intracellular membrane fusion and lipid bilayer mixing. *J. Cell Sci.* 118, 3819–3828. doi: 10.1242/jcs.02561

van Dijk, A. D. J., van der Krol, A. R., ter Braak, C. J. F., Bosch, D., and van Ham, R. C. H. J. (2008). Predicting sub-Golgi localization of type II membrane proteins. *Bioinformatics* 24, 1779–1786. doi: 10.1093/bioinformatics/btn309

Vang, Y. S., and Xie, X. (2017). HLA class I binding prediction via convolutional neural networks. *Bioinformatics* 33, 2658–2665. doi: 10.1093/bioinformatics/btx264

Wickner, W., and Schekman, R. (2008). Membrane fusion. *Nat. Struct. Mol. Biol.* 15, 658–664. doi: 10.1038/nsmb.1451

Yoshizawa, A. C., Kawashima, S., Okuda, S., Fujita, M., Itoh, M., Moriya, Y., et al. (2006). Extracting sequence motifs and the phylogenetic features of SNARE-Dependent membrane traffic. *Traffic* 7, 1104–1118. doi: 10.1111/j.1600-0854.2006.00451.x

Zheng, C., Lanczycki, C. J., Zhang, D., Hurwitz, D. I., Chitsaz, F., Lu, F., et al. (2014). CDD: NCBI's conserved domain database. *Nucleic Acids Res.* 43, D222–D226. doi: 10.1093/nar/gku1221

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

Copyright © 2019 Le and Huynh. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.