Identification of clathrin proteins by incorporating hyperparameter optimization in deep learning and PSSM profiles

Le, Nguyen Quoc Khanh; Huynh, Tuan-Tu; Yapp, Edward Kien Yee; Yeh, Hui-Yuan

2019

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

<https://hdl.handle.net/10356/143870>

<https://doi.org/10.1016/j.cmpb.2019.05.016>

© 2019 Elsevier B.V. All rights reserved. This paper was published in *Computer Methods and Programs in Biomedicine* and is made available with permission of Elsevier B.V.

Downloaded on 26 Apr 2023 23:08:21 SGT

Identification of Clathrin proteins by incorporating hyperparameter optimization in deep learning and PSSM profiles

Nguyen Quoc Khanh Le^{1,*}, Tuan-Tu Huynh², Edward Kien Yee Yapp³, and Hui-Yuan Yeh^{1,*}

¹Medical Humanities Research Cluster, School of Humanities, Nanyang Technological University, 48 Nanyang Ave, Singapore 639798

²Department of Electrical Electronic and Mechanical Engineering, Lac Hong University, No. 10 Huynh Van Nghe Road, Bien Hoa, Dong Nai, Vietnam

³Singapore Institute of Manufacturing Technology, 2 Fusionopolis Way, #08-04, Innovis, 138634, Singapore

*corresponding author: khanhle@ntu.edu.sg, hyyeh@ntu.edu.sg

Abstract

Background and Objectives: Clathrin is an adaptor protein that serves as the principal element of the vesicle-coating complex and is important for the membrane cleavage to dispense the invaginated vesicle from the plasma membrane. The functional loss of clathrins has been tied to a lot of human diseases, i.e., neurodegenerative disorders, cancer, Alzheimer's diseases, and so on. Therefore, creating a precise model to identify its functions is a crucial step towards understanding human diseases and designing drug targets.

Methods: We present a deep learning model using a two-dimensional convolutional neural network (CNN) and position-specific scoring matrix (PSSM) profiles to identify clathrin proteins from high throughput sequences. Traditionally, the 2D CNNs take images as an input so we treated the PSSM profile with a 20x20 matrix as an image of 20x20 pixels. The input PSSM

profile was then connected to our 2D CNN in which we set a variety of parameters to improve the performance of the model. Based on the 10-fold cross-validation results, hyper-parameter optimization process was employed to find the best model for our dataset. Finally, an independent dataset was used to assess the predictive ability of the current model.

Results: Our model could identify clathrin proteins with sensitivity of 92.2%, specificity of 91.2%, accuracy of 91.8%, and MCC of 0.83 in the independent dataset. Compared to state-of-the-art traditional neural networks, our method achieved a significant improvement in all typical measurement metrics.

Conclusions: Throughout the proposed study, we provide an effective tool for investigating clathrin proteins and our achievement could promote the use of deep learning in biomedical research. We also provide source codes and dataset freely at <https://www.github.com/khanhlee/deep-clathrin/>.

Keywords: clathrin coated pits; convolutional neural network; vesicular transport; molecular function; adaptor protein complex; position specific scoring matrix

1. Introduction

Clathrin is an adaptor protein that serves as the principal element of the vesicle-coating complex and is important for the membrane cleavage to dispense the invaginated vesicle from the plasma membrane [1]. It forms a triskelion shape including three clathrin heavy chains at their C-terminals and three light chains. Clathrin plays a vital role in deciding the function of vesicular transport in the cytoplasm for membrane trafficking [2, 3]. Clathrin-coated vesicles selectively separate cargo at the cell membrane, trans-Golgi network, and endosomal compartments for diversified membrane traffic pathways. After vesicle sprouts into the cytoplasm, the coat immediately disassembles, admitting the clathrin to reuse since the vesicle are transported to a variety of sub-locations. Some of the extracellular molecules, e.g., proteins, membrane receptors, and ion channels use clathrin as a specific uptake. Clathrin is also the main scaffold protein that is present in cellular uptake of DNA–chitosan nanoparticles. It includes a variety of cholesterol-rich pathways, e.g. the caveolae-mediated pathway [4]. Many studies determined that the functional missing of clathrins in cell systems would affect a variety of human diseases, e.g. cancer, Alzheimer, neurodegenerative, and so on [5-7].

Due to their essential role in human diseases, clathrin proteins attracted various researchers who conducted their research on them. Over the past decade, many research groups have applied different biological techniques to identify clathrin proteins, i.e., by using Tom1–Tollip complex [8], partial amino acid sequence [9], or agarose gel electrophoresis [10]. James et al [11] identified the clathrin-binding domain by proteolytically cleaving AP-2 into 2 discrete moieties, termed light and heavy mero-AP (LM-AP and HM-AP). Biological researchers also conducted experiments to discover new clathrins, such as assembly protein AP180 [12], γ 2-adaptin [13], TACC3/ch- TOG/clathrin complex [14], myelin basic protein [15], and so on.

Most published works on clathrin proteins achieved high performance, but to our knowledge, no researcher conducted the identification of clathrin proteins by using machine learning techniques. It is challenging and motivates us to create a precise model for this. In earlier years, researchers used shallow neural networks for solving problems in computational biology. For instance, Ou constructed the QuickRBF package [16] for training radial basis function (RBF) networks and applied it on several biological problems including classifying membrane transport proteins [17], transporters [17], and FAD binding sites [18]. Next, (who?) [19] introduced LibSVM to help biologists implement biological models by using support vector machines. Recently, as deep learning had been successfully applied in various fields, researchers started to use it in biomedical research, e.g., medical imaging prediction [20], cancer prediction [21] and protein secondary structures prediction [22]. Although those studies achieved very good performances, we believe that we can obtain superior results by using 2D CNN in some biological applications. Based on the advantages of deep learning, this study consequently proposes the use of a 2D convolutional neural network (CNN) constructed from position-specific scoring matrix (PSSM) profiles to identify clathrin proteins. The basic principle has already been successfully applied to identify electron transport proteins [23], Rab GTPases [24], and motor proteins [25]. Thus, in this paper, we extend this approach with in-depth analysis to identify the molecular functions of clathrin proteins. The main achievements, including contributions to the field, are presented as follows: (i) development of a deep learning framework to identify clathrins' functions from protein sequences, in which our model exhibited a significant improvement beyond traditional machine learning algorithms; (ii) first computational study to identify clathrin proteins and provide useful information to biologists to discover clathrins' molecular functions; (iii) valid benchmark dataset to train and test clathrin proteins with high accuracy, which forms a basis for

future research on clathrin proteins; and (iv) source codes and models for further researches in applying 2D CNN architecture in protein function prediction.

2. Materials and Methods

We implemented an efficient framework for identifying clathrin proteins by using a 2D CNN and PSSM profiles. The framework consists of four procedures: data collection, feature extraction, CNN generation, and model evaluation. Fig. 1 presents the flowchart of our framework, and its details are described as follows.

2.1. Data collection

The dataset was retrieved from the NCBI database [26], which is one of the comprehensive resources for biotechnology information. First of all, we collected clathrin proteins using the keyword ‘clathrin’ from the NCBI. There are many database sources in NCBI, and in this study, we chose the protein sequence from UniProt [27]. The proposed problem was the binary classification between clathrin proteins and general proteins, thus we collected a set of general proteins as negative data. In order to create a precise model, there is a need to collect negative dataset which has a similar function and structure with the positive dataset. From there it is challenging to build a precise model but it increases our contribution to the predictor. Therefore, we chose vesicular transport protein, which is a general protein including clathrin protein.

In most bioinformatics problems, authors should remove redundant sequences with similarity more than 30-40%. However, to fully utilize the superiority of deep learning, this study chose to remove the redundant sequences with similarity of 100%. With this cut-off level, we were able to retrieve enough data for deep neural networks and generate hidden information inside each sequence. To perform this step, we used BLAST [28], which is a common tool for clustering

biological sequences, and the rest of the proteins reached 1546 clathrins and 1360 non-clathrins. We then divided the data into the cross-validation and independent datasets. In the cross-validation set, there were 1288 clathrins and 1133 non-clathrins while the corresponding numbers in the independent set were 258 and 227, respectively. The details of the dataset used in this study are listed in Table 1.

2.2. Feature extraction for identifying clathrin proteins

In order to convert the protein secondary structure into feature sets, we applied the PSSM matrices for FASTA sequences. A PSSM profile is a matrix represented by all motifs in biological sequences in general and protein sequences in particular. It is created by rendering two sequences having similar structures with different amino acid compositions. Therefore, PSSM profiles have been adopted and used in a number of biological researches, e.g., prediction of protein secondary structure [29], RNA-binding sites [30], and dual-tropic HIV-1 [31], with significant improvements.

Since the retrieved dataset was in FASTA format, it needed to be converted into PSSM profiles. To perform this task, we used PSI-BLAST [28] to search all the sequence alignments of proteins in the non-redundant (NR) database. The feature extraction part of Fig. 1 indicates the information of generating the 400 PSSM capabilities from original PSSM profiles. Each element of the 400D input vector was divided by the sequence length and then inserted into neural networks.

2.3. 2D convolutional neural network architecture

We conducted this study using a 2D convolutional neural network, which is a conventional neural network for deep learning. 2D CNN had been applied in bioinformatics to identify the

protein functions with remarkable results, i.e., electron transport chains [23], Rab GTPases [24], cytoskeleton motor proteins [25], and SNARE proteins [32]. The bottom part of Fig. 1 illustrates the layer structure of the simplified convolutional neural network. Our deep learning architecture was carried out using the Keras library with TensorFlow backend [33]. GPU computing and CUDA kernel were also applied to accelerate the performance more efficiently. In general, CNN is composed of multiple layers with each layer performing a specific function of transforming its input into a useful representation. All layers were combined using a specific ordering to form the architecture of our CNN model. As shown in a sequence of studies on this field [23, 24], to create an effective model, one should follow the hyperparameter optimization step to find the good architecture and parameters. Different problems and datasets need a different set of layers and parameters. According to this rule, we performed this process in this study and reported as follows.

2.3.1. Layers

(1) Input layer: In this study, the input layer parameters were from PSSM profiles converted into 20x20 matrices. By using these matrices as the input data, we aim to propose a method to identify clathrin proteins from a set of general proteins. We assumed the 20x20 matrix to be an image of 20x20 pixels so that we could train the 2D CNN model with different weights and biases to enhance its predictive performance. The purpose of using a 2D CNN model is to capture the hidden features inside the PSSM profiles as opposed to a 1D structure.

(2) Zero padding layer: In the first few layers of deep neural networks, we would like to preserve as many hidden patterns about the original data so we apply the zero padding layers. This layer can add rows and columns with zero values at the top, bottom, left and right side of a PSSM matrix. When we apply 2x2 strides to a 20x20 matrix, the output volume would be 22x22. The

zero padding layer allowed our model to not have different output dimensions after applying filters to the input data.

(3) Convolutional layer: Convolutional layer is the core building block of a CNN to perform most of the computational heavy lifting. A convolutional layer is used to extract features encoded in the 2D input matrix via convolution operations. The convolutional layer takes a sliding window that is moved in stride across the input transforming the values into representative values. During this process, convolution operation preserves the spatial relationship between numeric values in the PSSM profiles by learning useful features using small squares of input data.

When constructing our model, we applied convolution to the 2D matrices with a 3x3 sliding window, the features were learned with the small 3x3 matrices and shifted one unit at a time. Each neuron received inputs with the weights and biases from the previous layer and trained again.

(4) Activation layer: the rectified linear unit (ReLU) plays an important role as an activation function used during the construction of the CNN to classify clathrin proteins. ReLU is the most important activation function for all deep neural networks and became popular in the last few years. The ReLU activation function is defined by the formula:

$$f(x) = \max(0, x) \quad (1)$$

where, x is the number of inputs in a neural network.

(5) Pooling layer: The pooling layer is usually inserted among the convolutional layers with the aim of reducing the size of matrix calculation for the next convolutional layer. The operation performed by this layer is also called “down-sampling” as it removes certain values leading to

less computational operations and overfitting control while still preserving the most relevant representative features. The pooling layer also takes a sliding window or a certain region that is moved in stride across the input matrix transforming the values into representative values. The transformation is either performed by taking the maximum value in the window (max pooling), or by taking the average of the values (average pooling). In this study, we used a generally known design of two pooling strides with 3x3 filters.

(6) Dropout layer was added to enhance the predictive performance of the present model and prevent overfitting [34]. In the dropout layer, the model will randomly deactivate the neurons in a layer with certain probability p . If the dropout value is added in a layer, the neural network will ignore selected neurons during training, and the training time will be faster. In this study, the dropout values ranging from 0 to 1 were used to evaluate our model.

(7) Flatten layer: Because the output layers require the distribution of all classes as probabilities, the flatten layers convert the input matrix into a vector. It is believed that this output can then be used in the following layers to generate information.

(8) Fully connected layer: Subsequently, we see a dense layer which is a regular fully connected neural network. In this layer, the classification will be accomplished on the features from the convolutional layers and the pooling layers. Including a fully connected layer is a typical approach of learning non-linear hybrids of the features.

(9) Loss function: Because our problem is a binary classification, we used the ‘binary_crossentropy’ as a loss function. This loss function has been proven effective in a number of binary classification problems [23, 32].

(10) Softmax (normalized exponential function): The output of the model was evaluated through a softmax function by which the probability for each possible output was determined. Softmax function is a logistic function defined by the formula:

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

where z in the above formula indicates the input vector with K -dimensional vector, $\sigma(z)_j$ is real values in the range $(0, 1)$ and j^{th} class is the predicted probability from sample vector x . In summary, we set a total of 374,658 trainable parameters in the model (Table 2).

2.3.2. Hyperparameters

Hyperparameters are parameters at the architecture level and differ from the parameters of a model trained through backpropagation. The choice of these hyperparameters is governed by a number of factors when building a deep learning model. It has a significant impact on the model's performance. For example, some important hyperparameters that affect the deep learning model are: number of convolutional layers, number of filters in each layer, number of epochs, the dropout rate, and the optimizers.

To tune hyperparameters, we need to choose a set of parameters for speeding up the training process and prevent overfitting. As suggested by Chollet [33], each step of the above hyperparameter-tuning approach was integrated into the hyperparameter-tuning process as follows:

- 1) Choose a set of hyperparameters.
- 2) Build the corresponding model.

- 3) Fit the training data to the model, and measure the final performance on the validation dataset.
- 4) Try the next set of hyperparameters.
- 5) Repeat.
- 6) Eventually, measure performance on an independent dataset.

2.4. Performance evaluation

The most important purpose of the present study was to predict whether or not a sequence is a clathrin protein; therefore, we used “Positive” to define the clathrin protein, and “Negative” to define the non-clathrin protein. For each dataset, we first trained the model by applying a 10-fold cross-validation technique on the training dataset. Based on the 10-fold cross-validation results, hyper-parameter optimization process was employed to find the best model for each dataset. Finally, an independent dataset was used to assess the predictive ability of the current model. The evaluation metrics used to measure the predictive performance of our model include sensitivity, specificity, accuracy, and MCC (Matthews’s correlation coefficient). We denote TP, FP, TN, FN as true positive, false positive, true negative, and false negative, respectively. Then the evaluation metrics are defined as follows [35, 36].

$$Sensitivity = \frac{TP}{TP + FN} \quad (3)$$

$$Specificity = \frac{TN}{TN + FP} \quad (4)$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (5)$$

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (6)$$

3. Results

The quality and reliability of the modeling techniques of research is an important factor in the study. Initially, we designed an experiment by analyzing data, perform calculations and take various comparisons in the results and discussions section.

3.1. Composition of amino acid in clathrin and non-clathrin sequences

We analyzed the composition of amino acid in clathrin sequences and non-clathrin sequences by computing the frequency between them. Fig. 2 illustrates the amino acids which contributed to the significantly highest frequency in two different datasets. We realized that there are not many differences between two types of data, however, there were some points here. The amino acid C and P occurred at the highest frequencies surrounding the clathrin proteins. On the other hand, amino acids L and I occurred at the highest frequencies surrounding the non-clathrin proteins. Therefore, these amino acids certainly had an essential role in identifying clathrin proteins. Thus, our model might predict clathrin proteins accurately via the special features from these amino acids contributions.

3.2. Performance results for identifying clathrin proteins with 2D CNN

We implemented our 2D CNN architecture by using Keras package with Tensorflow backend. First, we tried to find the optimal setup for the hidden layers by doing experiments using four different convolutional layers: 32, 64, 128, and 256. Table 3 demonstrates the performance results from various filter numbers in the cross-validation dataset. We easily observed that during the 10-fold cross-validation to identify clathrins, the model with structure of 32, 64 and 128 filter numbers was prominent, identifying sequences with an average 10-fold cross-validation accuracy of 88.4%. The performance results were higher than the performances from the other

results with other filter numbers. The sensitivity, specificity, and MCC for cross-validation data achieved 90%, 86.6%, and 0.77, respectively. Therefore, we used this convolutional layer structure in the hidden layers to develop our model. We then optimized the neural networks using a variety of optimizers: rmsprop, adam, nadam, sgd, and adadelta. The model was reinitialized, i.e. a new network was built, after each round of optimization so as to provide a fair comparison between the different optimizers. Overall, the performance results are shown in Fig. 3 and we decided to choose Adam, an optimizer with consistent performance, to create our final model. Adam is also an optimizer chosen in similar work in this field [24]. According to Fig. 3, we also realized that after the 80th epoch, our validation accuracy could not increase according to the training accuracy. Therefore, we determined to finish our training process at the 80th epoch to reduce the training time and prevent overfitting. Next, we tuned the other hyperparameters in our model (i.e., learning rate, batch size, or dropout rate) to achieve the best performance results for this dataset. After this step, all of the optimal hyperparameters can be seen in Table 4.

Overfitting is the most important concern out of all machine learning problems, which means our classifier can only perform well in our training set but worse in another unseen dataset. Therefore, we also used an independent test to ensure that our model also performs well in a blind dataset. As described in the previous part, our independent dataset contained 258 clathrins and 227 non-clathrins. None of these samples had a certain occurrence in the training set. Detailed results are shown as two confusion matrices in Fig. 4, in which our independent test result was consistent with the cross-validation result. To detail, our model reached the accuracy of 91.8%, sensitivity of 92.2%, specificity of 91.2%, and MCC of 0.83 in the independent test. Compared with the cross-validation result, the differences are not too much and it can show that

our model did not contain much overfitting. Another reason is the use of dropout inside our CNN structure and it has been proven effective in preventing overfitting [34].

3.3. Additional analysis on the important features of the CNN model

Deep learning methods have black-box nature and the features go from local to abstract hierarchical, therefore, it is hard to detect the important features in our CNN model. However, to provide more useful information for readers and biologists, we aimed to use a technique to capture the special features in this problem. Because we inputted 20x20 PSSM profiles to our CNN architecture, we would like to analyze the significant features in these matrices. We used F-score [37], which is a feature selection technique for the purpose of identifying features that have the greatest contribution towards improving the outcome of the problem. The idea was to find out differences between clathrin and non-clathrin sequences which our model would capitalize on to generate better results. Fig. 5 shows the feature maps of the F-scores of all our PSSM features, and it can be observed that there were differences between the two datasets. The features with the greatest contributions were in amino acids L, M, and P. In addition, some motifs in other amino acids had low contributions and these motifs might play the few essential roles in deciding the functions of clathrins. In summary, we found that our model might identify amino acids L, M, and P as important hidden features and it would be of aid to us to acquire the most important features according to the specific proteins and achieve the best result for each of them.

3.4. Comparative performance for identifying clathrins between 2D CNN and shallow neural networks

We examined the performances of using different machine learning classifiers for identifying clathrin proteins. We used four different classifiers (i.e., nearest neighbor (kNN), Random Forest, support vector machine (SVM), multi-layer perceptron (MLP) and 1D CNN) to evaluate the model and compared our 2D CNN results with their results. For a fair comparison, we used the optimal parameters for all the classifiers in all the experiments. Table 5 shows the performance results between our method and other machine learning algorithms. It can be seen that our 2D CNN exhibited higher performance than those of the other traditional machine learning techniques using the same experimental setup. Especially, our 2D CNN outperformed other algorithms when using the independent dataset.

4. Discussion

In this study, we presented a computational model aimed to classify clathrin proteins' molecular functions. We provide biologists with source codes and data for the reproduction of experiments and their academic work with a well-assembled protein collection and reliable information on clathrin sequences. In addition, our work is important in order to better understand the molecular functions of clathrins in the vesicular transport system. While previous publications only considered the identification of clathrin proteins using biological experiments [9, 11, 38], our study fills a gap in the completion work for clathrin sequences using machine learning techniques. Furthermore, it is the first computational study on this data set, which provides biologists much useful information to understand clathrins' molecular functions and to design drug targets according to their relevance in human diseases.

Furthermore, we were able to serve a profound deep learning architecture that achieves high performance in protein sequences. To select the best parameters for efficient optimization, we validated the performance via hyper-parameters tuning. The way to treat PSSM profiles as an

image and feed into our 2D CNN played a vital role in the classification of clathrins in particular and other proteins in general. Our system can be applied to detect hidden information even if such information is not known directly within sequences. While most traditional machine learning algorithms and previous works in this field [17, 18] could only use PSSM profiles as a vector when entering the networks; our findings serve as a different way to treat PSSM features and make them to be suitable for CNN networks. Moreover, our two-dimensional CNN model uses a number of measurement metrics to outperform alternative approaches [19, 39, 40] at the same dataset and level.

Our approach is totally suitable for real-time systems. For example, we can design a biological information retrieval and analysis system based on protein sequences from the internet. Moreover, as shown in a series of recent publications in the development of new prediction methods [18, 41], user-friendly and publicly accessible web servers will make their contributions significantly better. Furthermore, it can be used for developing a smart device that will serve the purpose of predicting protein functions according to their sequence information. This smart device is able to be developed more to discover human disease variants or mutations based on protein functions. From that, biologists can use that information to design drug targets for pharmaceutical research.

Our contributions take this research a step further and open the doors for further research that will enrich the computational biology field. The way to treat PSSM profiles as an image is a very important advantage that helped CNN performed well. Moreover, the use of GPU computing plays an important role in training a deep network with many hyper-parameter tunings inside. However, it still bears some limitations and there remains possible approaches to improve the proposed methodology in the future. Firstly, a huge number of datasets might increase the

performance of deep learning, therefore investigating and retrieving more data is a necessary task for improving the performance results. Secondly, future studies could investigate how to input all of the PSSM information into CNN networks to avoid as many missing features as possible. Thirdly, we want our future to be able to provide a web server for the method of prediction presented in this paper.

5. Conclusion

In summary, this study approaches an efficient model for identifying clathrin proteins by using deep learning. The idea is to transform PSSM profiles into matrices and use them as the input to 2D CNN architectures. We evaluated the performance of our model, which was developed by using a 2D CNN and PSSM profiles, using 10-fold cross-validation and an independent testing dataset. Our method produced superior performance, and compared to other state-of-the-art neural networks, it achieved a significant improvement in all the typical measurement metrics. Using our model, new clathrin proteins can be accurately identified and used for drug development. Moreover, the contribution of this study can help further promote the use of 2D CNN in biochemical research, especially in protein function prediction.

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

Funding. This research is partially supported by the Nanyang Technological University Start-Up Grant.

Statements of ethical approval

Not applicable.

Conflict of interest

The authors have no conflicts of interest to disclose.

ACCEPTED MANUSCRIPT

References

- [1] M.R. D'Andrea, Chapter 2 - Origin(s) of Intraneuronal Amyloid, in: M.R. D'Andrea (Ed.) *Intracellular Consequences of Amyloid in Alzheimer's Disease*, Academic Press 2016, pp. 15-41.
- [2] M.P. Lisanti, M. Flanagan, S. Puszkin, Clathrin lattice reorganization: theoretical considerations, *Journal of Theoretical Biology*, 108 (1984) 143-157.
- [3] D.N. McKinley, Model for transformations of the clathrin lattice in the coated vesicle pathway, *Journal of Theoretical Biology*, 103 (1983) 405-419.
- [4] Z. Garaiova, S.P. Strand, N.K. Reitan, S. Llu, S.. Strset, K. Berg, J. Malm, O. Folasire, A. Bjrky, C. de L. Davies, Cellular uptake of DNA–chitosan nanoparticles: The role of clathrin- and caveolae-mediated pathways, *International Journal of Biological Macromolecules*, 51 (2012) 1043-1051.
- [5] F. Wu, P.J. Yao, Clathrin-mediated endocytosis and Alzheimer's disease: An update, *Ageing Research Reviews*, 8 (2009) 147-149.
- [6] S.J. Royle, The cellular functions of clathrin, *Cellular and Molecular Life Sciences CMLS*, 63 (2006) 1823-1832.
- [7] Y. Miao, H. Jiang, H. Liu, Y.-d. Yao, An Alzheimers disease related genes identification method based on multiple classifier integration, *Computer Methods and Programs in Biomedicine*, 150 (2017) 107-115.
- [8] Y. Katoh, H. Imakagura, M. Futatsumori, K. Nakayama, Recruitment of clathrin onto endosomes by the Tom1–Tollip complex, *Biochemical and Biophysical Research Communications*, 341 (2006) 143-149.
- [9] S.M. Voglmaier, J.H. Keen, J.-E. Murphy, C.D. Ferris, G.D. Prestwich, S.H. Snyder, A.B. Theibert, Inositol hexakisphosphate receptor identified as the clathrin assembly protein AP-2, *Biochemical and Biophysical Research Communications*, 187 (1992) 158-163.
- [10] M.H. Gottlieb, C.J. Steer, A.C. Steven, A. Chrambach, Applicability of agarose gel electrophoresis to the physical characterization of clathrin-coated vesicles, *Analytical Biochemistry*, 147 (1985) 353-363.
- [11] J.H. Keen, K.A. Beck, Identification of the clathrin-binding domain of assembly protein AP-2, *Biochemical and Biophysical Research Communications*, 158 (1989) 17-23.
- [12] E. Ungewickell, L. Oestergaard, Identification of the clathrin assembly protein AP180 in crude calf brain extracts by two-dimensional sodium dodecyl sulfate-polyacrylamide gel electrophoresis, *Analytical Biochemistry*, 179 (1989) 352-356.
- [13] H. Takatsu, M. Sakurai, H.-W. Shin, K. Murakami, K. Nakayama, Identification and Characterization of Novel Clathrin Adaptor-related Proteins, *Journal of Biological Chemistry*, 273 (1998) 24693-24700.
- [14] D.G. Booth, F.E. Hood, I.A. Prior, S.J. Royle, A TACC3/ch- TOG/clathrin complex stabilises kinetochore fibres by inter- microtubule bridging, *The EMBO Journal*, 30 (2011) 906-919.
- [15] K. Prasad, W. Barouch, B.M. Martin, L.E. Greene, E. Eisenberg, Purification of a New Clathrin Assembly Protein from Bovine Brain Coated Vesicles and Its Identification as Myelin Basic Protein, *Journal of Biological Chemistry*, 270 (1995) 30551-30556.
- [16] Y.-J. Oyang, S.-C. Hwang, Y.-Y. Ou, C.-Y. Chen, Z.-W. Chen, Data classification with radial basis function networks based on a novel kernel density estimation algorithm, *IEEE Transactions on Neural Networks*, 16 (2005) 225-236.

- [17] N.Q.K. Le, G.A. Sandag, Y.-Y. Ou, Incorporating post translational modification information for enhancing the predictive performance of membrane transport proteins, *Computational Biology and Chemistry*, 77 (2018) 251-260.
- [18] N.Q.K. Le, Y.-Y. Ou, Prediction of FAD binding sites in electron transport proteins according to efficient radial basis function networks and significant amino acid pairs, *BMC Bioinformatics*, 17 (2016) 298.
- [19] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2 (2011) 27.
- [20] E. Gibson, W. Li, C. Sudre, L. Fidon, D.I. Shakir, G. Wang, Z. Eaton-Rosen, R. Gray, T. Doel, Y. Hu, T. Whyntie, P. Nachev, M. Modat, D.C. Barratt, S. Ourselin, M.J. Cardoso, T. Vercauteren, NiftyNet: a deep-learning platform for medical imaging, *Computer Methods and Programs in Biomedicine*, 158 (2018) 113-122.
- [21] Y. Xiao, J. Wu, Z. Lin, X. Zhao, A semi-supervised deep learning method based on stacked sparse auto-encoder for cancer prediction using RNA-seq data, *Computer Methods and Programs in Biomedicine*, 166 (2018) 99-105.
- [22] S. Babaei, A. Geranmayeh, S.A. Seyyedsalehi, Protein secondary structure prediction using modular reciprocal bidirectional recurrent neural networks, *Computer Methods and Programs in Biomedicine*, 100 (2010) 237-247.
- [23] N.Q.K. Le, Q.T. Ho, Y.Y. Ou, Incorporating deep learning with convolutional neural networks and position specific scoring matrices for identifying electron transport proteins, *Journal of Computational Chemistry*, 38 (2017) 2000-2006.
- [24] N.Q.K. Le, Q.-T. Ho, Y.-Y. Ou, Classifying the molecular functions of Rab GTPases in membrane trafficking using deep convolutional neural networks, *Analytical Biochemistry*, 555 (2018) 33-41.
- [25] N.Q.K. Le, E.K.Y. Yapp, Y.-Y. Ou, H.-Y. Yeh, iMotor-CNN: Identifying molecular functions of cytoskeleton motor proteins using 2D convolutional neural network via Chou's 5-step rule, *Analytical Biochemistry*, 575 (2019) 17-26.
- [26] N.R. Coordinators, Database resources of the National Center for Biotechnology Information, *Nucleic Acids Research*, 44 (2016) D7-D19.
- [27] U. Consortium, UniProt: a hub for protein information, *Nucleic Acids Research*, (2014) gku989.
- [28] S.F. Altschul, T.L. Madden, A.A. Schäffer, J. Zhang, Z. Zhang, W. Miller, D.J. Lipman, Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, *Nucleic Acids Research*, 25 (1997) 3389-3402.
- [29] D.T. Jones, Protein secondary structure prediction based on position-specific scoring matrices1, *Journal of Molecular Biology*, 292 (1999) 195-202.
- [30] J. Tong, P. Jiang, Z.-h. Lu, RISP: A web-based server for prediction of RNA-binding sites in proteins, *Computer Methods and Programs in Biomedicine*, 90 (2008) 148-153.
- [31] G.B. Fogel, S.L. Lamers, E.S. Liu, M. Salemi, M.S. McGrath, Identification of dual-tropic HIV-1 using evolved neural networks, *Biosystems*, 137 (2015) 12-19.
- [32] N.Q.K. Le, V.-N. Nguyen, SNARE-CNN: a 2D convolutional neural network architecture to identify SNARE proteins from high-throughput sequencing data, *PeerJ Computer Science*, 5 (2019) e177.
- [33] F. Chollet, Keras, 2015.

- [34] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, *Journal of Machine Learning Research*, 15 (2014) 1929-1958.
- [35] M. Abdar, N.Y. Yen, J.C.-S. Hung, Improving the Diagnosis of Liver Disease Using Multilayer Perceptron Neural Network and Boosted Decision Trees, *Journal of Medical and Biological Engineering*, 38 (2018) 953-965.
- [36] B. Wang, Y. Kong, Y. Zhang, D. Liu, L. Ning, Integration of unsupervised and supervised machine learning algorithms for credit risk assessment, *Expert Systems with Applications*, 128 (2019) 301-315.
- [37] Y.-W. Chen, C.-J. Lin, Combining SVMs with Various Feature Selection Strategies, in: I. Guyon, M. Nikravesh, S. Gunn, L.A. Zadeh (Eds.) *Feature Extraction: Foundations and Applications*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 315-324.
- [38] B.C.H. Lam, T.L. Sage, F. Bianchi, E. Blumwald, Role of SH3 Domain-Containing Proteins in Clathrin-Mediated Vesicle Trafficking in Arabidopsis, *The Plant Cell*, 13 (2001) 2499.
- [39] J.M. Keller, M.R. Gray, J.A. Givens, A fuzzy k-nearest neighbor algorithm, *IEEE Transactions on Systems, Man, and Cybernetics*, (1985) 580-585.
- [40] V. Svetnik, A. Liaw, C. Tong, J.C. Culberson, R.P. Sheridan, B.P. Feuston, Random forest: a classification and regression tool for compound classification and QSAR modeling, *Journal of Chemical Information and Computer Sciences*, 43 (2003) 1947-1958.
- [41] N.Q.K. Le, Y.-Y. Ou, Incorporating efficient radial basis function networks and significant amino acid pairs for predicting GTP binding sites in transport proteins, *BMC Bioinformatics*, 17 (2016) 501.

Figure Legends

Fig. 1. The flowchart for identifying Clathrin proteins using 2D convolutional neural networks

Fig. 2. Amino acid composition of Clathrin and non-Clathrin sequences

Fig. 3. The training and validation accuracy of different optimizers in this study (the epoch range from 0 to 150).

Fig. 4. Confusion matrices of: (a) cross-validation test, (b) independent test

Fig. 5. List of the important features generated by F-score analysis. x-axis: 20 amino acids, y-axis: F-scores for each amino acid position

Tables

Table 1. Statistics of all retrieved Clathrin and non-Clathrin sequences

	Original	Non-redundancy	Cross-validation	Independent
Clathrin	1584	1546	1288	258
Non-Clathrin	1464	1360	1133	227

Table 2. All layers and trainable parameters in 2D CNN architecture

Layer (type)	Remarks	Output shape	Parameters #
zeropadding2d_1	padding=(2,2)	(None, 22, 22, 1)	0
convolution2d_1	filters=32, kernels=(3,3)	(None, 20, 20, 32)	320
max_pooling2d_1	pool size=(2,2)	(None, 20, 10, 16)	0
zeropadding2d_2	padding=(2,2)	(None, 22, 12, 16)	0
convolution2d_2	filters=64, kernels=(3,3)	(None, 20, 10, 64)	9280
max_pooling2d_2	pool size=(2,2)	(None, 20, 5, 32)	0
zeropadding2d_3	padding=(2,2)	(None, 22, 7, 32)	0
convolution2d_3	filters=128, kernels=(3,3)	(None, 20, 5, 128)	36992
max_pooling2d_3	pool size=(2,2)	(None, 20, 2, 64)	0
flatten_1	flatten	(None, 2560)	0
dropout_1	p=0.2	(None, 2560)	0
dense_1	units=128	(None, 128)	327808
dense_2	units=2	(None, 2)	258
activation_1	softmax	(None, 2)	0

Table 3. Performance results of identifying Clathrins with different filter numbers

Filter numbers	Cross-validation				Independent			
	Sens	Spec	Acc	MCC	Sens	Spec	Acc	MCC
32	91.1	77.5	84.8	0.70	81	85.9	83.3	0.67
32-64	92.1	79.6	86.2	0.73	94.2	68.7	82.3	0.66
32-64-128	90	86.6	88.4	0.77	89.5	93.4	91.3	0.83
32-64-128-256	91	84.9	88.1	0.76	92.6	79.3	86.4	0.73

Table 4. All of the optimal hyperparameters using in this study

Hyperparameter	Value
Number of epochs	80
Learning rate	0.001
Batch size	10
Kernel size	3
Dropout rate	0.2
Optimizer	Adam

Table 5. Comparative performance between 2D CNN and other shallow neural networks

Filters	Cross-validation				Independent			
	Sens	Spec	Acc	MCC	Sens	Spec	Acc	MCC
kNN	89.6	75.7	83.1	0.66	88.8	73.6	81.6	0.63
RandomForest	90.9	91.5	91.2	0.82	88.8	91.6	90.1	0.80
SVM	95.8	84.2	90.4	0.81	96.1	82.8	89.9	0.80
MLP	88.0	87.0	87.6	0.75	91.9	87.2	89.7	0.79
1D CNN	84.6	86.2	85.4	0.71	84.5	82.8	83.7	0.67
2D CNN	91.7	91.6	91.7	0.83	92.2	91.2	91.8	0.83

Figures

Fig. 1. The flowchart for identifying Clathrin proteins using 2D convolutional neural networks

Fig. 2. Amino acid composition of Clathrin and non-Clathrin sequences

Fig. 3. The training and validation accuracy of different optimizers in this study (the epoch range from 0 to 150).

Fig. 4. Confusion matrices of: (a) cross-validation test, (b) independent test

Fig. 5. List of the important features generated by F-score analysis. x-axis: 20 amino acids, y-axis: F-scores for each amino acid position.