DeepETC: A deep convolutional neural network architecture for investigating and classifying electron transport chain's complexes

Nguyen Quoc Khanh Le\textsuperscript{a,b,*}, Quang-Thai Ho\textsuperscript{c}, Edward Kien Yee Yapp\textsuperscript{d}, Yu-Yen Ou\textsuperscript{c,*}, Hui-Yuan Yeh\textsuperscript{a,*}

\textsuperscript{a} Medical Humanities Research Cluster, School of Humanities, Nanyang Technological University, 48 Nanyang Ave, 639798, Singapore
\textsuperscript{b} Professional Master Program in Artificial Intelligence in Medicine, Taipei Medical University, Taipei 106, Taiwan
\textsuperscript{c} Department of Computer Science and Engineering, Yuan Ze University, Chung-Li 32003, Taiwan
\textsuperscript{d} Singapore Institute of Manufacturing Technology, 2 Fusionopolis Way, #08-04, Innovis 138634, Singapore

**A R T I C L E I N F O**

Article history:
Received 30 September 2018
Revised 23 September 2019
Accepted 24 September 2019
Available online xxx
Communicated by FangXiang Wu

Keywords:
Deep learning
Cellular respiration
Position specific scoring matrix
Molecular function
Computational biology
Protein complex classification

**A B S T R A C T**

An electron transport chain is a series of protein complexes embedded in the transport protein, which is an important process to transfer electrons and other macromolecules throughout the cell. It is the primary process to extract energy via redox reactions in the case of oxidation of sugars in cellular respiration. According to the molecular functions, the components of the electron transport chain could be formed with five complexes and with several different electron carriers. The functional loss of a specific molecular function in electron transport chain has been implicated in a variety of human diseases such as diabetes, neurodegenerative disorders, Parkinson, and Alzheimer’s disease. Therefore, creating a precise model to identify its functions is pertinent to the understanding of human diseases and designing of drug targets. Previous bioinformatics studies have almost exclusively focused on the electron transport proteins without information on the five complexes. Here we present DeepETC, a deep learning model that uses a two-dimensional convolutional neural network and position-specific scoring matrices profiles to classify electron transport proteins into the five complexes. DeepETC can classify the electron transporters with the independent test accuracy of 99.6%, 99.7%, 99.7%, 99.1% and 99.8% for complex I, II, III, IV, and V, respectively. Our performance results are significantly more accurate than the state-of-the-art traditional neural networks in all typical measurement metrics. Throughout the proposed study, we provide an effective tool for investigating electron transport proteins and our achievement could promote the use of deep learning in bioinformatics and computational biology. DeepETC can be freely accessible via http://www.biologydeep.com/deepetc/.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Cellular respiration is a mechanism that creates adenosine triphosphate (ATP) and aids cells in obtaining energy from food molecules (i.e., sugar). To achieve this goal, cellular respiration uses a complex of proteins to accumulate electrons, which are called electron transport chains [1]. Fig. 1 (adapted from [2]) indicates the process of the electron transport chain; a pathway to store and transfer electrons in cellular respiration. The electron transport chain can be categorized into five complexes: complex I, II, III, IV, and V (ATP Synthase). Each complex consists of different electron carriers and it executes various molecular functions [3]. An electron would be donated to complex I from NADH and sequentially passed to complex II, III, IV, and V. During this movement, the hydrogen ions, or protons, pump across the membrane and release the water molecules (H2O). Complex V uses the energy created by the pumping process to phosphorylate adenosine diphosphate (ADP) to ATP.

Numerous types of electron transport proteins have been identified in humans. A series of studies conducted indicated that in many diseases, there was a functional loss of specific complexes in the electron transport protein. For instance, in [4], all the Parkinson's disease patients had a significant reduction of complex I (NADH dehydrogenase activity). From this result, it can be hypothesized that the complex I abnormality may have an etiological role in the pathogenesis of Parkinson's disease. Parker et al. [5] suggest that in Alzheimer's disease patients, there is a cytochrome c oxidase deficiency (complex IV) in the terminal portion of the

---

\* Corresponding authors.
\** Corresponding author at: Professional Master Program in Artificial Intelligence in Medicine, Taipei Medical University, Taipei 106, Taiwan

E-mail addresses: khanhle@tmu.edu.tw (N.Q.K. Le), yien@saturn.yzu.edu.tw (Y.-Y. Ou), hyeh@ntu.edu.sg (H.-Y. Yeh).

https://doi.org/10.1016/j.neucom.2019.09.070
0925-2312/© 2019 Elsevier B.V. All rights reserved.

Please cite this article as: N.Q.K. Le, Q.-T. Ho and E.K.Y. Yapp et al., DeepETC: A deep convolutional neural network architecture for investigating and classifying electron transport chain's complexes, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.09.070
electron transport chain and in the platelet mitochondria. Mutations in BCS1, which is an assembly factor for complex III, are associated with a syndrome called GRACILE (growth retardation, aminoaciduria, cholestasis, iron overload, lactic acidosis, and early death) [6]. The ubiquinone of complex III is regarded as a major site of reactive oxygen species generation, which plays a crucial role in the aging process and the pathogenesis of neurodegenerative diseases [7]. The complex IV of the electron transport chain has been linked to the pathogenesis of diabetes mellitus [8]. Thus, the classification of the electron transport protein complexes would help biologists better understand that the molecular functions in human diseases is an essential problem. This would then spur them on to develop bioinformatics techniques to resolve it.

Recently, there have been many published scholarly papers on electron transport proteins using computational techniques. Its popularity can be attributed to the fact that electron transport proteins play an essential role in cellular respiration, energy production, and human diseases. For instance, one of the most prominent studies done is on TCDB [9], which is a web-accessible, curated, relational database containing the sequence, classification, structural, functional and evolutionary information on transport systems, including electron transport proteins from a variety of living organisms. Research done by Gromiha discriminates the function of electron transport proteins from membrane proteins using machine learning techniques [10]. According to Chen [11], in the experiment, the transport targets were divided into four types, including the electron transporters, for prediction and analysis. The analysis was done using the amino acid composition (AAC), amino acid pair composition (DPC) and position specific scoring matrix (PSSM). The property of these three attributes continued to individually perform cross-merger forecast and group data using ten-fold cross-validation method to do the performance evaluation. Furthermore, Le et al. [12] implemented deep learning framework and PSSM in their study for accurate identification of electron transport proteins.

Research based on previous studies can only be considered as the first step toward a more profound understanding of electron transport proteins. A new approach is therefore needed to investigate the details of the electron transport proteins’ complexes. Even the previous work from Le et al. [13] has investigated the molecular functions of electron transport chain, however, they used a small set of data with a shallow neural network. Here we present DeepETC, a web server for classifying electron transport protein’s complexes using deep learning on a bigger dataset. The idea of constructing a 2D convolutional neural network (CNN) from PSSM profiles has been presented in earlier works [12,14,15], and here we extend this approach with a different dataset and a more in-depth analysis. We document several vital contributions of our study to the field of biology: (1) a database for collecting all the complexes of electron transport proteins, (2) a first computational model to genuinely classify electron transport protein into their complexes using deep learning, which has been successfully applied in some biological applications yielding outperformed results [16–19], (3) a benchmark dataset and newly discovered data for further study on electron transport chain (4) a study that would provide much information to biologists and researchers, allowing them to better understand the electron transport protein structures and to conduct the future research.

2. Materials and methods

Most experiments have been carried out with a 2D CNN and PSSM profiles. There are four steps in our methodology: data collection, feature extraction, CNN implementation, and model evaluation. Our flowchart is illustrated in Fig. 2 and described in detail in the following paragraphs.

2.1. Data collection

This study used two extensive resources for gene and protein sequence, which are Uniprot (release-2017.12) [20] and the latest version (2017) of Gene Ontology [21], to collect the data. In Gene Ontology, three pieces of genomic information which can be collected are its cellular component, its molecular function, and its
Fig. 2. The flowchart for classifying electron transport proteins using two-dimensional convolutional neural networks. It includes four subprocesses: data collection, feature set generation, CNN generation and model evaluation.

### Table 1

The molecular functions of electron transport proteins complexes according to Gene Ontology annotation.

<table>
<thead>
<tr>
<th>Molecular function</th>
</tr>
</thead>
<tbody>
<tr>
<td>I  oxidoreductase activity, acting on NADH or NADPH</td>
</tr>
<tr>
<td>NADH dehydrogenase activity</td>
</tr>
<tr>
<td>oxidoreductase activity, acting on NADH or NADPH, quinone or similar compound as acceptor</td>
</tr>
<tr>
<td>NADH dehydrogenase (quinone) activity</td>
</tr>
<tr>
<td>NADH dehydrogenase (ubiquinone) activity</td>
</tr>
<tr>
<td>II succinate dehydrogenase activity</td>
</tr>
<tr>
<td>succinate dehydrogenase (ubiquinone) activity</td>
</tr>
<tr>
<td>ubiquinol-cytochrome-c reductase activity</td>
</tr>
<tr>
<td>electron transporter, transferring electrons within CoQH2-cytochrome c reductase complex activity</td>
</tr>
<tr>
<td>IV cytochrome-c oxidase activity</td>
</tr>
<tr>
<td>V proton-transporting ATPase activity, rotational mechanism</td>
</tr>
</tbody>
</table>

biological process. In this study, the molecular function annotation information was used to obtain the benchmark dataset. The Gene Ontology molecular functions of electron transport proteins were listed in Table 1 as well as described in [13]. Note that in this section, we chose all of the reviewed proteins with information extracted from literature and curator-evaluated computational analysis. In most of the bioinformatics problems, the important step is to eliminate redundant sequences with a similarity of more than 30 or 40%. This ensures that our dataset does not contain any similar sequences between the training and the testing dataset. However, after using BLAST [22] with a similarity of more than 40%, the rest of the proteins reached 427 proteins in complex I, 43 proteins in complex II, 29 proteins in complex III, 166 proteins in complex IV, and 95 proteins in complex V. Considering the typical size of datasets (>100 K) used in the learning of CNNs in the field of computer vision, we decided to use an identity level of 100% (only removed sequences with 100% similarity) in our dataset to allow a bigger dataset and to fully utilize deep learning.

To address the multi-classification issue, we applied the approach from the previous works [15,23]. We separated it into five binary classification problems, in which we used one complex as the positive data and the rest as the negative data. The experiment was then repeated four more times with each of the other four complexes as the positive data and the rest as the negative data. To conduct the experiments, we divided the collected sequences into two datasets: the training dataset and the independent dataset. We used the training dataset with cross-validation to create our model and used the independent dataset to evaluate the performance of the model. The details of all the dataset used in this study are listed in Table 2.

2.2. Feature extraction for identifying five complexes of electron transport proteins

In this study, the feature extraction method that was applied is the PSSM profile, which is a matrix represented by all motifs in biological sequences as well as protein sequences. It is created by rendering two sequences having similar structures but with different amino acid compositions. Numerous biological researchers have successfully used PSSM profiles to address a lot of bioinformatics problems, i.e., prediction of DNA binding protein [24], protein–protein interaction [25], and binding sites [26] with impressive improvements.

The first step that needs to be done is to convert the sequences as FASTA format into PSSM profiles. To carry out this undertaking,
we employed PSI-BLAST [22] to search our electron transport protein sequences in the non-redundant (NR) database. To recognize the complexes of electron transport chain, we determined the most dependable protein sequence for each amino acid. We put 20 types of amino acids in the correct sequences, leading to the occurrence of a matrix. We firstly summed up all of the identical amino acid values and each value of the 400D input vector was divided by the sequence length. (In the feature extraction part of Fig. 2.)

### 2.3. Input layers of 2D CNN

DeepETC model was implemented by using 2D CNN architecture, which is a common class of deep neural networks. 2D CNN had been shown to perform significantly in various fields, i.e., image classification [27], emotion recognition [28], handwriting recognition [29] and more. The bottom part of Fig. 2 fundamentally depicted the layer construction of a simplified CNN. Keras library with Tensorflow backend [30] was used as a deep learning framework to build the CNN architecture. We also accelerated the performance via graphic processing unit (GPU) computing and CUDA kernel. In general, CNN contains multiple layers with each layer performing a specific function of translating its input into an appropriate representation. All layers were integrated using a specific order to form the architecture of our CNN model. In this study, an input of the CNN is a PSSM representing for the sequence of electron transport proteins. Traditionally, the 2D CNNs take images as an input. Thus, we treat the PSSM profile with a 20 × 20 matrix as an image with 20 × 20 pixels. The input PSSM profile was then connected to our hidden layers, in order to enhance the performance of the model with different hyperparameters. By using the 2D CNN, we aim to capture the hidden spatial features in PSSM matrices rather than other shallow neural networks. In a deep neural network, well-designed hidden layers can generate sufficient discriminative features to classify electron transport chain's complexes easily. A number of filter layers were used in this work and each filter consisted of three kernel sizes.

### 2.4. Multiple hidden layers of 2D CNN

Hidden layers are the most important layers in 2D CNN. They consisted of different sub-layers with specific weights and parameters which aim to learn the input features. Those sub-layers are 2D zero padding layers, 2D convolutional layers, 2D max pooling layers, and fully connected layers. We put all the layers with different parameters in a correct ordering and they formed our CNN architecture. The number of parameters and weights in the hidden layers will decide the quality of the model. The first layer of our DeepETC structure is a 2D zero padding layer, which adds zeros to the beginning and the end of the 20 × 20 matrices. Whenever zero padding is added to this matrix, the shape of the matrix is changed to 22 × 22. When applying the zero padding layer, we guarantee a controlled size of the output through the formula:

\[
zp = \frac{k - 1}{2}
\]  

where \( k \) is the filter size. Next, a 2D convolution layer with a kernel size of 3 × 3 is inserted, meaning that the features will be learned with the 3 × 3 matrices and shifted one unit at a time. As the input shape is 20 × 20 dimension, we cannot choose a kernel size with a different width and length. Thereafter, the next layer will pick up the weights and biases from its previous layer and train again. The 2D convolutional was followed by a 2D max-pooling layer which aims to decrease the processing time of the next layers. In this layer, there are a few parameters that could be set, i.e., stride and loop size. In this study, we would like to select the maximum value of the matrices, thus we performed max pooling with a stride of 2. The output size of a convolutional layer was calculated by using formula (2) as follows:

\[
aw = \frac{w - k + 2p}{s} + 1
\]  

where \( w \) is the input size, \( k \) is the filter size, \( p \) is the padding size, and \( s \) is the stride size. After every convolutional layer, we inserted a Furthermore, an additional non-linear operation called Rectified Linear Unit (ReLU) which is an extra non-linear operation defining by the formula:

\[
f(x) = \max(0, x)
\]  

where \( x \) is the input size.

### 2.5. Output layers of 2D CNN

The output layers were constructed to export the classification results after convolutional steps. Firstly, a flattened layer is regularly added before fully-connected layers to transform the data from convolution layers into vectors. In the fully connected layers, each node is fully connected with all the nodes of the preceding layers. Fully connected layers are commonly inserted in the final stages of the CNN. In the present model, two fully connected layers were included. The first layer connected all the nodes into the flatten layer to enable our model to acquire more knowledge and achieve better performance. The second fully connected layer connected the first one to the output layers. Furthermore, a dropout layer was added to enhance the overall performance of the present model and avoid overfitting [31]. In the dropout layer, the model will randomly deactivate a number of neurons with a certain probability \( p \). By ranging this \( p \) value from 0 to 1, the neural network will learn different, redundant representations, and reducing time-consuming. Finally, our model contained a softmax function by which the probability for each possible output was established. It is a logistic function defined by the formula:

\[
\sigma(z) = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}}
\]  

where \( z \) is the input vector with \( K \)-dimensional vector, \( K \)-dimensional vector \( \sigma(z) \) is real values in the range \((0, 1)\) and \( i \)th class is the predicted probability from sample vector \( x \). \( K \) is 2 in this study which represents the 2D vector. In Table 3, we show all the layers as well as a total of 158,981 trainable parameters in

### Table 2

Statistics of all retrieved electron transport proteins with molecular function annotation in Gene Ontology.

<table>
<thead>
<tr>
<th>Complex</th>
<th>Original Sequence Identity &lt; 40%</th>
<th>Sequence Identity &lt; 100%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cross-validation</td>
<td>Independent</td>
</tr>
<tr>
<td>Complex I</td>
<td>5148</td>
<td>356</td>
</tr>
<tr>
<td>Complex II</td>
<td>154</td>
<td>36</td>
</tr>
<tr>
<td>Complex III</td>
<td>1781</td>
<td>24</td>
</tr>
<tr>
<td>Complex IV</td>
<td>1072</td>
<td>138</td>
</tr>
<tr>
<td>Complex V</td>
<td>444</td>
<td>79</td>
</tr>
</tbody>
</table>

Please cite this article as: N.Q.K. Le, Q.-T. Ho and E.K.Y. Yapp et al., DeepETC: A deep convolutional neural network architecture for investigating and classifying electron transport chain’s complexes, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.09.070
Table 3
All layers and trainable parameters in DeepETC.

<table>
<thead>
<tr>
<th>Layer (type)</th>
<th>Output shape</th>
<th>Parameters #</th>
<th>Connected to</th>
</tr>
</thead>
<tbody>
<tr>
<td>zeropadding2d_1</td>
<td>(None, 1, 22, 22)</td>
<td>0</td>
<td>zeropadding2d_input_1</td>
</tr>
<tr>
<td>convolution2d_1</td>
<td>(None, 32, 20, 20)</td>
<td>320</td>
<td>convolution2d_1[0][0]</td>
</tr>
<tr>
<td>maxpooling2d_1</td>
<td>(None, 32, 10, 10)</td>
<td>0</td>
<td>convolution2d_1[0][0]</td>
</tr>
<tr>
<td>zeropadding2d_2</td>
<td>(None, 32, 12, 12)</td>
<td>0</td>
<td>maxpooling2d_2[0][0]</td>
</tr>
<tr>
<td>convolution2d_2</td>
<td>(None, 64, 10, 10)</td>
<td>18,496</td>
<td>convolution2d_2[0][0]</td>
</tr>
<tr>
<td>maxpooling2d_2</td>
<td>(None, 64, 5, 5)</td>
<td>0</td>
<td>convolution2d_2[0][0]</td>
</tr>
<tr>
<td>zeropadding2d_3</td>
<td>(None, 64, 7, 7)</td>
<td>0</td>
<td>maxpooling2d_3[0][0]</td>
</tr>
<tr>
<td>convolution2d_3</td>
<td>(None, 128, 5, 5)</td>
<td>73,856</td>
<td>convolution2d_3[0][0]</td>
</tr>
<tr>
<td>maxpooling2d_3</td>
<td>(None, 128, 2, 2)</td>
<td>0</td>
<td>convolution2d_3[0][0]</td>
</tr>
<tr>
<td>flatten_1</td>
<td>(None, 512)</td>
<td>0</td>
<td>flatten_1[0][0]</td>
</tr>
<tr>
<td>dropout_1</td>
<td>(None, 512)</td>
<td>0</td>
<td>dropout_1[0][0]</td>
</tr>
<tr>
<td>dense_1</td>
<td>(None, 128)</td>
<td>65,664</td>
<td>dense_1[0][0]</td>
</tr>
<tr>
<td>dense_2</td>
<td>(None, 2)</td>
<td>645</td>
<td>dense_2[0][0]</td>
</tr>
<tr>
<td>activation_1</td>
<td>(None, 5)</td>
<td>0</td>
<td>dense_2[0][0]</td>
</tr>
</tbody>
</table>

DeepETC. Across all five classification problems, we used the same number of layers as well as most of the parameters inside each layer. An only parameter was not shared across all five classes is dropout value. Since we ranged dropout value from 0 to 1 to find the optimal values for each binary classification, thus each of them has different optimal dropout value. As a result, the dropout values for five complexes classification are 0.2, 0.1, 0.1, 0.2, and 0.1, respectively. Based on this point, we can say that our model setup is consistent with all complexes of the electron transport chain.

2.6. Performance evaluation

We used 5-fold cross-validation technique on the training set to train the model and independent test to evaluate the performance. To find the optimal CNN model, we performed a hyper-parameter optimization process in the 5-fold cross-validation testing. Furthermore, the independent dataset was re-used to evaluate the performance accuracy to avoid any systematic bias in the cross-validation set [15].

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, FN, TN, FN as true positive, false positive, true negative, and false negative, respectively. Thus, the evaluation metrics are defined as follows:

\[
\text{Sensitivity} = \frac{TP}{TP + FN} \tag{5}
\]

\[
\text{Specificity} = \frac{TN}{TN + FN} \tag{6}
\]

\[
\text{Accuracy} = \frac{TP + TN}{TP + FN + FN + TN} \tag{7}
\]

\[
\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FN)(TN + FN)}} \tag{8}
\]

3. Results and discussions

3.1. Amino acid composition of five complexes in electron transport chain

We investigated the distribution of the amino acid composition and the variance among five complexes of the electron transport chain. Fig. 3 shows the amino acids which have had the substantially highest frequency in five distinct datasets. It is no doubt to say that there were a number of differences among the five complexes of electron transport chain in their amino acid compositions. For instance, the amino acid E could be adopted for classifying complexes III and V; the amino acid C could be adopted for classifying the Complex II. Combined into one multi-classification problem, the variance of composition amino acid pairs illustrates that amino acids E, L, and F contain different information and features within five complexes. Thus, these amino acid distributions indeed possess an important role in classifying electron transport proteins into five complexes.

3.2. Comparative performance of classifying electron transport chain’s complexes with shallow neural networks

We have classified the complexes of electron transport proteins using different classifiers, i.e., nearest neighbor (kNN) [32], RandomForest [33], LibSVM [34], QuickRBF [35], and LibD3C [36]. We also determined the optimal hyper-parameters for each of these classifiers. We ranged the number of nearest neighbors in kNN from one to ten, performed a grid search to estimate the accuracy of each parameter combination to find the optimal cost and gamma in LibSVM (log2c was ranged from −5 to 5, log2g was ranged from −4 to 0); number of trees were ranged from 100 to 500 in RandomForest; the bandwidth set to five in QuickRBF; and cross-validation for parameter tuning in libD3C [15]. We described the experimental results of PSSM profiles features with five-fold cross-validation and independent dataset, which show in Table 4. We observed that there were not many differences in performance results among these traditional classifiers. In particular, kNN classifier achieved the highest result in complex V (sensitivity of 93.7% and MCC of 0.95). Random Forest classifier achieved the highest result in complex IV while LibSVM performed well in complex I, II, and III. Moreover, LibD3C is a classifier which helps us solve the imbalance dataset, especially in complex II and IV (with the highest sensitivity compared with other classifiers).

3.3. Performance of classifying electron transport proteins with 2D CNN

In site of the high-performance results from traditional classifiers in the previous section, we would like to enhance them using 2D CNN architecture. First, we tried to find the optimal setup for the hidden layers using four different filters ranging from 32 to 256. It should be noted that we only ran the hyper-parameters tuning in the cross-validation process. This aids in reducing the biases and overfitting problem when we feed blind data into our model. The experimental results then showed that the model with 128 filters performed better compared to the other filter levels on the cross-validation. Therefore, we applied 128 filters in our hidden layer to develop this study’s model.

We also tried different optimizers in our experiments and compared the performance. The results determined that the ADADELTA optimizer produced the best performance in term of sensitivity, specificity, accuracy, and MCC. Moreover, many experimental studies mentioned the importance of epochs in deep learning and increasing the number of epochs may affect performance results. Therefore, we ran our experiments by ranging the epoch value from the first epoch to the 500th epoch to find the optimal epoch. The model checkpoint was used in this process to capture the epoch point that had the highest performance. Fig. 4 showed the accuracy and loss when we ranged the epochs from one to 500. From the data gathered we can say that the accuracy and loss are stable when the epoch value ran to around 50. The next parameter that was applied in this section is the dropout, which was intended to enhance the performance of the neural network and to avoid overfitting inside the dataset. After several experimental testing, we set the optimal dropout values for each binary problem. Therefore, our CNN model applied 128 filter layers, 500 epochs and flexible dropout values to classify the electron transport chain’s complexes. Table 5 (one-vs-rest part) shows the predictive performance...
Table 4
Comparative performance of classifying electron transport proteins using various traditional machine learning techniques.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sen</td>
<td>MCC</td>
<td>Sen</td>
<td>MCC</td>
<td>Sen</td>
</tr>
<tr>
<td>Cross-validation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kNN</td>
<td>97.6</td>
<td>0.94</td>
<td>65.8</td>
<td>0.77</td>
<td>96.2</td>
</tr>
<tr>
<td>RF</td>
<td>98.9</td>
<td>0.96</td>
<td>60.8</td>
<td>0.77</td>
<td>95.5</td>
</tr>
<tr>
<td>LibSVM</td>
<td>99.9</td>
<td>0.96</td>
<td>68.3</td>
<td>0.82</td>
<td>97.8</td>
</tr>
<tr>
<td>QuickRBF</td>
<td>99.9</td>
<td>0.96</td>
<td>65</td>
<td>0.8</td>
<td>97.6</td>
</tr>
<tr>
<td>libD3C</td>
<td>98.9</td>
<td>0.95</td>
<td>85.8</td>
<td>0.51</td>
<td>95</td>
</tr>
<tr>
<td>Independent</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kNN</td>
<td>99.9</td>
<td>0.98</td>
<td>75</td>
<td>0.86</td>
<td>97.3</td>
</tr>
<tr>
<td>RF</td>
<td>99.9</td>
<td>0.98</td>
<td>58.3</td>
<td>0.76</td>
<td>95.2</td>
</tr>
<tr>
<td>LibSVM</td>
<td>100</td>
<td>0.97</td>
<td>70.8</td>
<td>0.84</td>
<td>97.3</td>
</tr>
<tr>
<td>QuickRBF</td>
<td>100</td>
<td>0.97</td>
<td>70.6</td>
<td>0.84</td>
<td>96.9</td>
</tr>
<tr>
<td>libD3C</td>
<td>99.9</td>
<td>0.97</td>
<td>83.3</td>
<td>0.73</td>
<td>95.2</td>
</tr>
</tbody>
</table>

(Optimal parameters: kNN: k = 100, RF: num_trees=500, num_features=20, LibSVM: c = 8.0, g = 0.03125, QuickRBF: bandwidth=5).

Fig. 3. Amino acid composition and variance of amino acid composition in five complexes of electron transport proteins. Amino acid E could be adopted for classifying complexes III and V; the amino acid C could be adopted for classifying the Complex II. Combined into one multi-classification problem, the variance of composition amino acid pairs illustrates that the amino acid E, L, and F contains many different information and features within five complexes.

results of the deep learning model. It can be seen that our 2D CNN revealed higher performance than those of the other shallow neural networks at the same level comparison.

Here, of importance is the way we treat the multi-class classification problem as five separate binary classification problems (one-vs-rest). This diverges from the deep learning literature where multi-class classification is typically handled by having the output layer to have the same number of nodes as the number of classes and applying a softmax function on the output layer to get a class prediction. The disadvantage of the one-vs-rest approach is that it needs to learn separate parameters for each binary classification problem which would increase the risk of overfitting.
However, as shown in Table 3, we set the same configurations and parameters for all five binary problems so that this disadvantage can be addressed. Moreover, with different dropout values (which are not shared across five classes), we could generate particular strong models for five different complexes via one-vs-rest solution. We also show the comparative performance between one-vs-rest and multi-classification method on our dataset. As shown in Table 5, the performance from one-vs-rest method outperforms multi-classification in most of metrics. Therefore, we applied the one-vs-rest method in this problem to reach the best model.

3.4. Comparative performance between PSSM profiles and the other sequence features

In this section, we focus on investigating the importance of using PSSM profiles in our deep neural network. We tried to compare the performance between PSSM profiles and some latest sequence features in the bioinformatics field. Two advanced features we chose were pseudo-amino acid composition (PseAAC) and pseudo-K-tuple reduced amino acids composition (PseKRAAC), which are state-of-the-art features and reached the significant results in this field [37]. To generate these two features, we used iFeature, which is a python toolkit for calculating a wide range of structural and physicochemical feature descriptors from protein and peptide sequences [38]. Table 6 shows the comparative performance of the three features. It was observed that the results of the PSSM feature outperformed the other two features in most of the measurement metrics. Therefore, it provides compelling evidence for the ability to use PSSM profiles in CNN to increase the performance results of this problem.

3.5. Comparison between two cut-off levels from blast

As mentioned in the previous section, we aim to take advantages of deep neural networks in this study, and hence used the identity level of 100% in BLAST. Now we attempt to show a comparison between the two cases: sequence identity of 40% and 100%. The amount of data used in this part is also shown in Table 1 where we can easily see the differences between the two levels of sequence identity. We then used the same CNN architecture to perform experiments on the dataset with a similarity of 40% and compare them to the dataset with a similarity of 100% (via multi-classification method). As shown in Table 7, the performance results with a similarity of 100% completely outperformed the 40% one. Moreover, there was no overfitting in our model. Consequently, this means that we can use this cut-off level in bioinformatics to create a more precise model than other traditional methods. This is because we used deep learning, and thereby require a large amount of data in order to access the hidden information and produce the best results.

3.6. Implementing DeepETC server for classifying electron transport proteins

We built DeepETC containing two primary functions, a database resource for retrieving data as well as a deep learning model for classifying the molecular function of electron transport chains. First, the database is essential for biologists who would like to retrieve the data and conduct the research on electron transport proteins. Our sequences have been collected from UniProt [20] and GeneOntology [21], which are the extensive resources for the gene products. Our latest database contains 8556 specific electron transport proteins which include all the information, such as electron transport proteins’ complexes, sequences, 3D structures, and binding sites information. The database was built based on PHP and MySQL techniques; it was also optimized by advanced techniques to have an efficient framework. For its second function, we created a submission page which would aid the users in the
identification of an unknown protein to be belonging to the electron transport protein. Additionally, the submission page would help in the classification of an electron transport protein into the five complexes. Users are only required to input the sequences, and our models will process them and return the results efficiently. The web interface is built with user-friendliness in mind, to allow for comfortable and easy access to its functions. The users can access our web server freely via http://biologydeep.com/deepectc/.

4. Conclusion

Deep learning has been increasingly used in different fields, especially in bioinformatics and computational biology, with significant results. According to its development, this study presents DeepETC, a web server for storing and identifying the electron transport chain’s complexes through the use of deep learning. Our DeepETC contains two primary functions e.g., a database contains all electron transport chain’s information and a submission page contains deep learning models to classify them. The idea of the model is the transformation of PSSM profiles into matrices and then a 2D CNN was constructed to capture the features. Our DeepETC is able to classify the electron transport chain into five complexes with the independent test accuracy of 99.6%, 99.7%, 99.7%, 99.8%, and 99.8% respectively. It produced a relatively superior performance in comparison to other shallow neural networks as well as other related published works. Using our model, new electron transport proteins with correct complex annotation can be discovered and used for drug development. Furthermore, the contribution of this study could help further researchers to collect all electron protein sequences with sufficient information.

Declaration of Competing Interest

The authors have no conflicts of interest to disclose.

Acknowledgments

This research is partially supported by the Nanyang Technological University Start-Up Grant and the Ministry of Science and Technology, Taiwan, R.O.C. under Grant no. MOST 106-2221-E-155-068.

References


Please cite this article as: N.Q.K. Le, Q.-T. Ho and E.K.Y. Yapp et al., DeepETC: A deep convolutional neural network architecture for investigating and classifying electron transport chain’s complexes, Neurocomputing. https://doi.org/10.1016/j.neucom.2019.09.070
Nguyen Quoc Khanh Le is an Assistant Professor with the Department of Computer Science and Engineering, Taipei Medical University, Taiwan. He received his BS and MS degree in Computer Science from the National Taiwan University, Taiwan. Currently, he is working as a Research Assistant in the same department. His research interests are Artificial Intelligence, Machine Learning, Data Science, and Healthcare Informatics.


Yu-Yen Ou is an Associate Professor in the Department of Computer Science and Engineering, National Taiwan University, Taiwan. He received his BS in Computer Science and Engineering from the National Taiwan University, Taiwan. Currently, he is working as an Assistant Professor in the Department of Computer Science and Information Engineering at National Taiwan University, Taiwan. His research fields of interest are Machine Learning, Data Mining, and Web Mining.

Please cite this article as: N.Q.K. Le, Q.-T. Ho and E.K.Y. Yapp et al., DeepETC: A deep convolutional neural network architecture for investigating and classifying electron transport chain’s complexes, Neurocomputing. https://doi.org/10.1016/j.neucom.2019.09.070