Malphite: A Convolutional Neural Network and Ensemble Learning Based Protein Secondary Structure Predictor

Yang Li
Department of Computer Science, Graduate School of Information Science and Technology, University of Tokyo,
Tokyo, Japan
&
CREST, JST, Japan
lconvers@hgc.jp

Tetsuo Shibuya
Human Genome Center, Institute of Medical Science,
University of Tokyo,
Tokyo, Japan
&
CREST, JST, Japan
tshibuya@hgc.jp

***Abstract*—We developed a convolution neural networks (CNN) and ensemble learning based method, called Malphite, to predict protein secondary structures. Maphite has three sub-models: the 1st CNN, PSI-PRED and the 2nd CNN. The 1st CNN and PSI-PRED are used to predict the initial secondary structure based on the position specific scoring matrix generated from PSI-BLAST. The 2nd CNN performs ensemble learning by combining the prediction result of the 1st CNN and PSI-PRED and generate the final predictions. Malphite achieved a Q3 score of 82.3% and 82.6% for independently built dataset of 400 and 538 proteins respectively, and 82.6% ten-fold-cross validated accuracy for a dataset of 3000 proteins. In addition, Malphite accomplished a remarkable Q3 score of 83.6% for 122 targets from CASP10 (Critical Assessment of protein Structure Prediction), surpassing any secondary structure prediction technique to date. For all four datasets, Malphite consistently makes 2% more accurate prediction than PSI-PRED, which is a significantly step towards the estimated upper limit of protein secondary structure prediction accuracy of 90%.**

Keywords—protein secondary structure, convolutional neural network, ensemble learning;

I. INTRODUCTION

Protein secondary structure (SS) is the regular local structure segment of protein formed by hydrogen bonds. There are three different types of SS: α-helix (H), β-strand (E), and coil (C). The prediction of secondary structure from amino acid sequence plays an important role in protein structure modeling and function analysis. For example, many *ab initio* folding based protein 3D structure prediction methods, like VECFOLD [1] and RosettaDom [2], use predicted SS elements as the basic subunits to fold into protein tertiary structure. The accuracy of secondary structure prediction is critical for the challenging task of protein tertiary structure prediction.

The most often used criteria to measure the performance of a protein secondary structure predictor is the Q3 score, which is the total number of correctly predicted residues divided by the total number of residues. The accuracies for helices (QH), sheets (QE) and coils (QC) are also assessed in term of the fraction of correctly predicted residues out of the total number of residues in a given state. The Q3 accuracy, however, is believed to have an estimated upper bond of about 90% due to the 5~15% discrepancy between X-ray determined protein 3D structures and NMR based ones, and the ignorance of long-range amino acid interactions by current protein secondary structure assignment tools [3]. CASP (Critical Assessment of protein Structure Prediction) [4] is a biennially held worldwide experiment since 1994, aiming at establishing the current state of the art in protein structure prediction. The protein targets from CASP experiments are frequently used as the benchmarks for evaluating the accuracies of many protein secondary structure prediction methods.

Protein secondary structure prediction has been intensively studied during the past decades, but the Q3 accuracy remains at a bottleneck of around 81% and further improvement is very hard. Machine learning approach is widely used for protein secondary structure prediction. Many methods apply Support Vector Machines [5-9], Hidden Markov Model [10-12] or Artificial Neural Networks [13-21]. In the currently most effective secondary structure prediction methods, a two stage neural network based method PSI-PRED [20], achieves a Q3 accuracy of 81.7%; SPINE X [21], which makes use of 5 iterative neural networks, yields a Q3 score of 81.8%.

Though great efforts were paid by previous researchers to find better protein secondary structure predictors, no serious attempt was made to try to improve prediction accuracy by combining existing methods. TABLE I. shows the secondary structure prediction accuracy of PSI-PRED and SPINE X for CASP9 targets. We found that there is a significant diversity in the prediction distributions of the two methods: though the overall Q3 accuracy of the two methods almost tie, PSI-PRED makes more accurate prediction in coil residues (5%) while SPINE X does better in helical residues (5.8%). Such kind of diversity can also be observed in many other secondary structure prediction methods. Therefore, the concept of ensemble learning [22], which achieves a strong classifier by combining several base classifiers, might be able to improve protein secondary structure prediction by taking the advantage of the diversity in existing SS prediction methods.

This work is partially supported by JSPS KANENHI grant numbers 252-8002 and 24106007.

978-1-4673-6799-8/15/$31.00 ©2015 IEEE

TABLE I. SECONDARY STRUCTURE PREDICTION ACCURACY FOR THE CASP9 SET

Method	Score			
	Q3	*QH*	*QE*	*QC*
PSI-PRED	81.7	82.2	75.9	84.5
SPINE X	81.8	88.0	76.2	79.5

The data in this table come from [21]

In this paper, we proposed a convolutional neural network (CNN) and ensemble learning based protein secondary structure prediction method named Malphite. The design of CNN was inspired by visual mechanisms in living organisms and later improved by Yan LeCun [23]. CNN has been applied to many pattern recognition problems like image classification [24], voice analysis and natural language processing [25], and achieves high performance. We used a modified version of CNN so that it is prepared for the task of protein structure prediction. Two CNNs with different configurations are used in Malphite, where a first CNN is trained to predict initial secondary structure from amino acid sequence and a second CNN is trained to perform ensemble learning by combining the prediction of the first CNN and PSI-PRED. A set of 122 prediction targets from CASP10 was used as main benchmarks. The first CNN and PSI-PRED achieve Q3 accuracy of 82.4% and 81.4% respectively, showing that CNN is a more capable method than PSI-PRED. The great potential of ensemble learning for protein secondary prediction was also revealed by the second CNN, which further improved the Q3 accuracy to 83.6% against the same 122 CASP10 targets. The 83.6% Q3 accuracy is by far the highest performance for any published method. Given the success of Malphite for CASP10, it is reasonable to be confident that the evaluation presented here gives a fair indication of the performance of the method in general.

II. METHODS

A. Convolutional Neural Network

A Convolutional Neural Network (CNN) is comprised of one or more convolutional layers and then followed by one or more fully connected layers as in a standard multilayer neural network. The input and output of each convolutional layer are sets of arrays called feature maps. Generally, the feature maps can vary from 1D to 3D arrays. For instance, if the input is a RGB mode image, 3 feature maps will be used and each of them is a 2D array containing the red, green or blue color channel. In this work, the feature maps are 1D arrays because the input data are related to 1D amino acid sequence. For more details on the CNN, readers are referred to [23, 24, 25, 26, 27, 28]. In addition, an easy CNN tutorial and implementation for MINIST handwriting digit recognition problem can be found here:(http://deeplearning.net/tutorial/lenet.html).

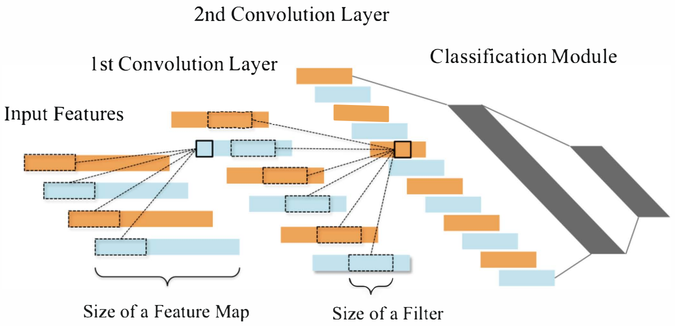

Fig. 1. Fully connected convolutional neural network. Each horizatal color strip represents a 1D featur map. Classification module is a standard fully connected neural network with one hiden layer.

The overall architecture of the CNN for protein secondary structure prediction is depicted in Fig. 1. It has 2 convolutional layers, and an universal classifier with one hidden layer. All feature maps are 1D arrays that are related to protein sequence (Two CNNs are used in this work, each of them uses different kind of data as input feature maps. The input feature maps for the 1st CNN will be explained in II-B and II-C-1), and the ones for the 2nd CNN can be seen in II-C-3)). Filters are smaller 1D arrays. In a convolutional layer, the convolution products between a single filter and all input feature maps are calculated and then added together, and the result will be used to generate a single output feature map. Due to the smooth effect of convolution operator, the newly generated feature map tends to contain less noise data and becomes more representative than the input feature maps. By using different filters of the same size, a convolutional layer can generate many different new feature maps, the number of which equals to the number of filters have been used. The purpose of stacking two convolutional layers is to extract a deep hierarchical representation of features, which is able to improve the performance of the classification module. Finally, the topmost feature maps will be classified by classification module. The detail of the CNN is now described.

1) Convolutional layer: As depicted in Fig. 2, the input is a 2D array with n_1 1D feature maps of length n_2. Each feature map is denoted x_i and i is in range [1, n_1]. The output of a convolution layer is also a 2D array with m_1 1D feature maps of length m_2. Each feature map is denoted y_j and j is in range [1, m_1]. A trainable 1D filter k_j in the filter bank has length l and connects input feature maps $x_1, x_2, \ldots, x_{n1}$ to an output feature map y_j. For j in range [1, m_1], the convolution layer computes $y_j = tanh\ (b_j + \sum_{i=1}^{n1} k_j * x_i)$, where $*$ is the 1D discrete convolution operator, b_j is a trainable bias parameter

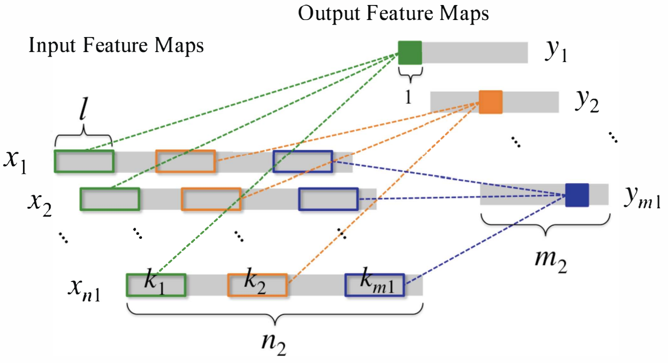

Fig. 2. A convolutional layer. x_i and y_j represents a input and an output feature map respectively. k_j is the filter that connects all input features with an output feature y_j

and *tanh()* is a pointwise function that transfers each data point into range [-1, 1].

2) Classification module: A standard feed forward back propagation neural network architecture [29] with a single hidden layer was used for classification module. The number of input units equals the number of feature maps times the length of each feature map in the second convolution layer.

Supervised learning [30] is done using gradient descent to minimize the difference between the expected output and the real output of the network. All the parameters, including the values of all the filters, weights and bias in all the layers, are updated simultaneously by the learning procedure. The gradients are computed with the back-propagation method.

B. *Feature Maps from Position Specific Scoring Matrix*

It has been widely considered that position-specific scoring matrix (PSSM) is the most informative resource for protein secondary structure prediction [20]. PSSM is the intermediate sequence profile generated from the searching process of PSI-BLAST. This matrix has M×b elements, where M is the length of target protein sequence and b, equals 20, is the number of amino acid types (The latest version of PSI-BLAST generate PSSM with 40 columns, which have been shown to reveal more valuable information about the protein). Each element in PSSM represents the log-likelihood of that particular residue substitution at that position in the template (based on a weighted average of BLOSUM62 matrix scores for the given alignment position).

Then, here comes the key question for applying convolutional neural network to protein secondary structure prediction: how to get input feature maps from PSSM?

We trained a small protein secondary structure classifier to answer this question: The classifier has 3 neurons, and no hidden layer was used. We used $P(y=i\,|\vec{X})=softmax(\vec{X}\cdot\vec{W}_i+b)$, as the activation function where $i\in\{$'H', 'E', 'C'$\}$, representing helix, sheet and coil respectively, $\vec{X}$ is the input vector of size 300 (15-residue window of 20 column version PSSM), $\vec{W}_i$ is the weight vector for i, and b is the bias value. We trained this module with 1044 proteins and it achieved a Q3 accuracy of about 68%. The three neurons' weight vectors were plotted after the module was well trained. The plot is shown in Fig. 3. Note that the plot of a neurons' weight vector roughly implies the pattern of the input vector that can activate this neuron because the inner product $\vec{X}\cdot\vec{W}_i$ become huge when $\vec{X}$ matches $\vec{W}_i$.

Despite that the classifier only got a score of 68%, the weight vectors' plot still expose two significant points about the relationships between PSSM elements' values and secondary structure. First of all, weight's absolute values grow bigger towards the center of the window. This is because the model was trained to predict the secondary structure of the center residue in the 15-residue window, and protein secondary structure is primarily determined by local interactions between residues closely spaced [31]. Secondly, elements that belong to the same columns of PSSM are very regularly weighted by all

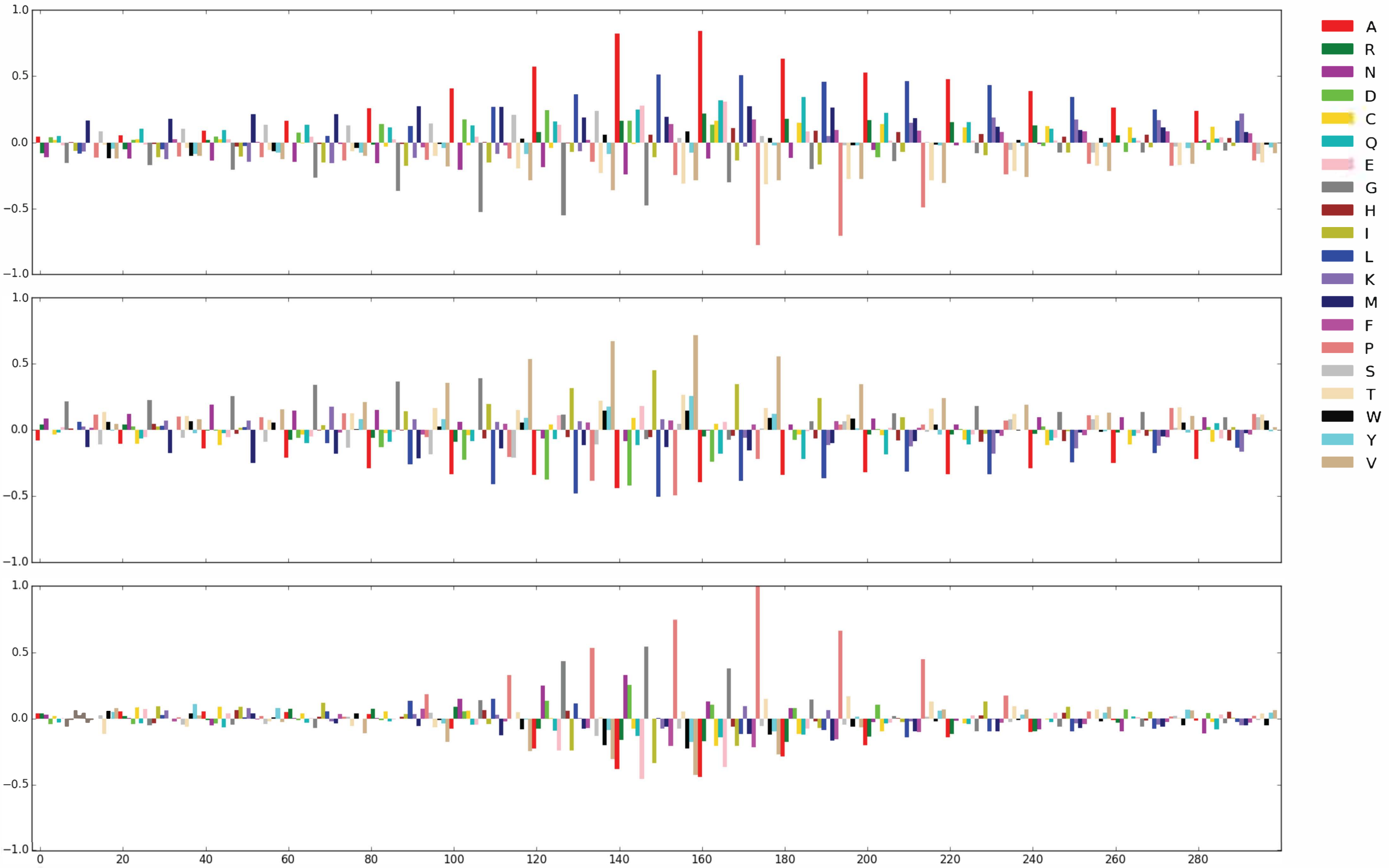

Fig. 3. Weight vectors of 3 neurons in the order of H, E and C. The weigth vectors have a size of 300 connecting to 15 rows of PSSM values. 20 different color are used to mark the corresponding input elements' column positions in PSSM. It is clear that elements belong too the same PSSM columns are simultaneously positively or negtively weight by all three nerons, which indicates that the column arrays can be regarded as independent feature of the protein sequence.

the three neurons. That is, they are simultaneously positively weighted or negatively weighted by the same neurons, which indicates some high correlation. While no restrict relationship can be observed between elements in different PSSM columns. This suggests that the 20 column arrays of PSSM should be regarded as 20 different independent features of the protein sequence.

C. Architecture of Maphite

Fig. 4 demonstrates the overall architecture of Malphite. Maphite consists of three sub-models: the 1st CNN, PSI-PRED and the 2nd CNN. The reason why we introduce PSI-PRED to Malphite is that there exists a significant diversity in the prediction distribution of the 1st CNN and PSI-PRED, which will be shown in results section. And ensemble learning tends to yield better results when there is a significant diversity among the models. The 1st CNN and PSI-PRED are used to predict the initial secondary structure from amino acids based on the sequence profile generate from PSI-BLAST. The goal of the 2nd CNN is to perform ensemble learning by combing the prediction result from the 1st CNN and PSI-PRED. Both CNNs uses the structure shown in Fig. 1. And the latest version of PSI-PRED (Version 3.5) with default parameters is used.

Both the 1st CNN and PSI-PRED (Version 3.5) use 40-column version PSSM. It can be generated by running the latest version PSI-BLAST against NCBI non-redundant database (Uniref90filt database is used for PSI-PRED) for 3 iterations. The profile matrix elements are then scaled to the range [0,1] by a standard sigmoid function: $1/(1+e^x)$, where x is the value of PSSM elements.

1) First CN: CNN's topology is determined by many hyper-parameters: the size and number of feature maps in each convolutional layers, the length of filters and the number of hidden units in classification module. As we have discussed before, we use column arrays of PSSM as input feature maps for the first CNN. We applied 40-column version of PSSM and used 15-residue sliding window. Therefor the 1st CNN has 40 input feature maps with length 15. There are not too much options for the size of filters: 6 would be too large not to lose information, while 2 is too small to smooth noise effectively. We tried filter size of 3, 4 and 5, and 4 was found to be optimal for both the first and second convolutional layer. Upon knowing the sizes of filters and input feature map, the feature maps in the first and second convolutional layer become a fixed size of 12 and 9 (only valid values are picked by each convolution operator). Since the figure map size decrease with depth, we put 35 and 60 feature maps on the first and second convolutional layer to roughly equalize the computation at each layer. Those numbers are not critical as long as there are enough features to preserve the information about the input [32]. The classification module has 540 input units, 100 hidden units and 3 output units representing the three states of secondary structure.

2) PSI-PRED: Since 1999, when David Jones [20] first time discovered the effectiveness of using PSSM as neural network input, PSI-PRED has kept being the top ranked accurate secondary structure predictor. PSI-PRED is a two-stage standard feed-forward back propagation neural network, with the first network to predict structure from sequence and the second one to predict structure from structure. A window size of 15 residues is used for both neural networks. To use it in Malphite, we downloaded the standalone version of PSI-PRED from (http://bioinf.cs.ucl.ac.uk/psipred/), which can run on a PC.

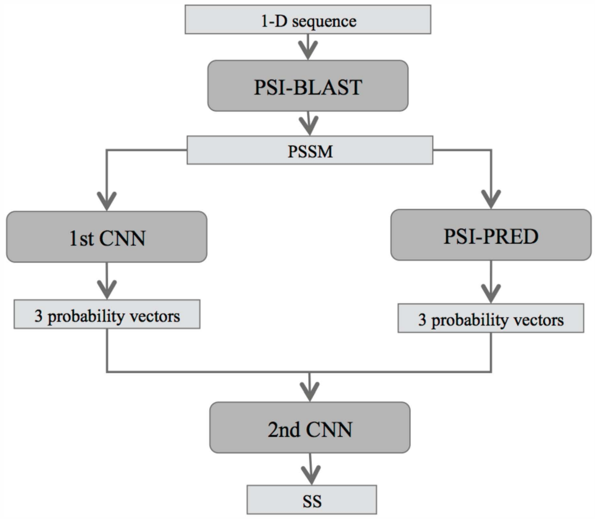

Fig. 4. The workflow of Malphite.

3) Second CNN: As we have discussed, the job of the second CNNs is to perform ensemble learning on two base models: the 1st CNNs and PSI-PRED. Both the 1st CNNs and PSI-PRED output 1 predicted SS vector and 3 probability vectors of length n where n is the length of protein amino acid sequence and 3 stands for 3 secondary structure classes. The elements in the probability vectors are always in range $[0,1]$, representing the probability of the residue being helix, strand or coil. Compared with the discrete SS vector, the probability vectors are more informative features because they contain real value of prediction results. Thus, we use the six probability vectors with 15-residue window from the output of PSI-PRED and the first CNNs as the input feature maps for the second CNN (Since all the elements in the 6 probability vectors are in the range $[0,1]$, pre-normalization is not needed). Therefore the second CNN has 6 feature maps of length 15. Like the 1st CNN, we use 4 for filters' size and 12 and 9 for features' size in the 2nd CNN. Since the 2nd has less input features, we use a reduced 15 and 20 feature maps on the first and second convolutional layers. The classification module contains 180 input units, 40 hidden units and 3 output units representing the three states of secondary structure.

From the definition of the 2nd CNN, it is easy to see that Malphite is open to ensembles of more protein secondary structure prediction models. As long as the models generate 3 probability vectors during the prediction processes, Malphite can absorb them in by plugging the probability vectors into the 2nd CNN.

D. Datasets for Training and Testing

We obtained a large non-homologous sequence and structure dataset of 7538 proteins (after removing proteins with less than 40 or more than 500 amino acids and discontinuous chains) from the protein sequence culling server PISCES [33] with sequence identity less than 25% and resolution better than 3Å. We divided the 7538 proteins randomly into datasets of 3600, 3000, 538 and 400 proteins.

The dataset of 3600 proteins was used only to train the 1st CNN (The dataset becomes too biased to train or evaluate the 2nd CNN after being used to train the 1st CNN). The dataset of 3000 proteins is used to evaluate the performance of the 1st CNN and PSI-PRED (Note that the 3000 proteins are not used for the training of either the 1st CNN or PSI-PRED, they can be used as fair benchmarks for them), and perform ten-fold-cross validation on the 2nd CNN. During the ten-fold-cross validate period. The dataset is randomly separated into ten equal parts, nine of which are used for training and the left part is used for testing. The process is repeated ten times for each of the ten parts, and each trial uses the identically configured CNN. To combat over-fitting, 10 percent of the training data were set aside to evaluate the performance of the model during the training process. When the performance of the model on the subset starts to drop or fails to improve for many epochs the training would be halted. After all 10 trainings, the final cross validation score can be obtained by averaging the validation score of all ten times.

Now, since the dataset of 3600 proteins becomes biased after training the 1st CNN, and the dataset of 3000 proteins produce different test score on three models (Normal scores for PSI-PRED and the 1st CNN, while cross-validation score for the 2nd CNN), extra datasets are needed to compare the performance of the three models directly. Here we get the datasets of 400 and 538 proteins to do this job. In addition, we downloaded 122 targets from CASP10, which allow us to compare the accuracy of secondary structures predicted by Malphite with those from other structure prediction techniques.

The three secondary structure label for each residue in the above datasets were assigned by DSSP software [34]. The eight states (H, I, G, E, B, S, T, −) DSSP assignments are grouped as follows: G and I into H, representing helix; B and E into E, representing strand; T, S and (−) into C representing coil.

E. Implementation

The CNNs are trained in a supervised form by Stochastic Gradient Decent (SGD), which estimates the error gradient from a batch from the training examples. We used a batch size of 1000 labeled examples. The weight of the CNNs are randomly initialized between the two values: $-6/\sqrt{fan_{in}+fan_{out}}$ and $6/\sqrt{fan_{in}+fan_{out}}$, where fan_{in} is the number of inputs and fan_{out} is the number of outputs of the layer. Learning rate of 0.008 was found to be effective for both the first and second CNNs. *Tanh()* activation function is used for all cases. All the models are implemented using Python's deep learning library: Theano [35, 36]. The trainings are parallelized on the Shirokane3 Super Computer of Human Genome Center (https://supcom.hgc.jp/english/), which is composed of about ten thousands Xeon E5-2670 CPUs.

III. Results

A. The Datasets of 400, 538 and 3000 proteins

TABLE II summarizes the performance given by the three components of Malphite against the datasets of 400, 538 and 3000 proteins.

The 1st CNN's Q3 scores consistently exceed PSI-PRED in all 3 dataset. Despite that the improvements are small (0.4%, 0.3% and 0.4%), CNN can outperform PSI-PRED with no doubt. Because PSI-PRED use two consecutive neural networks while the 1st CNN only uses one. We did another experiment which shows that (0.8%~1.2%) improvement over PSI-PRED can be achieved by stacking two CNNs straightly. On the other hand, as we have expected, there is a significant diversity in the prediction distributions of PSI-PRED and the 1st CNN. That is, PSIPRED makes the most accurate prediction for coil residues while the most accurate prediction in the 1st CNN is for helical residues. The accuracy of helical residues predicted by the 1st CNN is about 6% higher than the prediction by PSIPRED for all three datasets while the accuracies of strand residues slightly prefer the 1st CNN and prediction of coil residues is 5.8%~6.2% more accurate for PSI-PRED.

The combined prediction Q3 accuracies achieved by the 2nd CNN are consistently 2% higher than PSI-PRED's and 1.6% higher than the 1st CNN's for all three datasets. The 2nd CNN surpass both the 1st CNN and PSI-PRED for the prediction accuracies of helical and strand residues. QH is improved by about 1% and QE by 2.7% over the 1st CNN for all three datasets. However, in coil residues, the 2nd CNN can make 1.1~1.8% more accurate predictions than the 1st CNN but still lose to PSI-PRED's 86%. It seems that the 2nd CNN can't "inherit" the high QC performance from PSI-PRED. The reason for this will be shown later. Anyway, a significantly improved Q3 accuracy of about 2% over PSI-PRED is achieved by Malphite for the datasets of 400, 538 and 3000 proteins.

TABLE II. Performance of the Three Components of Malphite for 400, 538, and 3000 Sets

Models	**Score**	**Dataset**		
		400	***538***	***3000***
PSI-PRED (V 3.5)	***Q3***	80.5	80.5	80.6
	QH	80.4	79.2	79.7
	QE	71.3	73.0	72.6
	QC	86.3	86.7	86.5
1st CNN	***Q3***	80.9	80.8	80.9
	QH	86.2	85.5	85.5
	QE	72.6	74.0	74.0
	QC	80.5	80.6	80.3
2nd CNN	***Q3***	82.3	82.6	82.6[a]
	QH	87.2	86.4	86.6[a]
	QE	75.3	76.9	76.7[a]
	QC	81.6	82.4	82.1[a]

[a] Ten-fold-cross validation accuracy

TABLE III shows the compositions of secondary structure types predicted three sub-models of Malphite. All three models under predict helical and strand residues, and over predict coil residues. The amount of helical, strand and coil residues provided by two CNNs are very close to the amount of native ones. For the two datasets, both CNNs only under predict helical and strand residues by around 1% and 1.5% respectively, and over predict coil residues by 2.5~2.9%. The overall secondary structure composition predicted by the 2nd CNNs is 97.2~97.4% close to the native composition. By comparison, PSIPRED under predicts helical residues by 5%, strand residues by 3% and over predicts coil residues by about 8%, getting a total of 92% similarity to the native composition.

The over prediction of coil residues by both the 1st CNN and PSI-PRED can explain why the 2nd CNN can not "inherit" PSI-PRED's high coil prediction performance: PSI-PRED's high prediction accuracies in coil residues are not result from a good coil pattern recognizing ability but from high coil over prediction (8%), hence the coil probability vector generated by PSI-PRED can not provide equivalently beneficial coil information for the 2nd CNN, as a result, the 2nd CNN lose to PSI-PRED on QC score. For an extreme case, a model can get 100% QC score by over predicting all amino acids to coil residues, but only become meaningless.

TABLE III. COMPOSITION OF ACTUAL AND PREDICTED STRUCTURE TYPE FOR 538 AND 3000 SETS

Models	538			3000		
	%H	*%E*	*%C*	*%H*	*%E*	*%C*
Native	37.7	24.3	38.0	38.6	23.6	37.9
PSI-PRED	32.5	21.5	46.0	33.6	20.8	45.6
1st CNN	36.8	22.3	40.9	37.7	21.8	40.5
2nd CNN	36.5	22.7	40.9	37.5	22.0	40.4

The data for 400 proteins is removed for cleanness.

B. CASP 10 targets

We have also investigated the accuracy of secondary structure prediction for target proteins in CASP 10 competition. A total of 122 proteins are included in this set.

TABLE IV. shows the results given by the three sub-models of Malphite. PSI-PRED is widely known as the CASP-winning protein secondary structure prediction method, and it yields a decent 81.6% Q3 accuracy on CASP10. However, our 1st CNN, which comprises of a single stage neural network, nailed it with 82.5%. The 1st CNN makes more accurate prediction in helical residues while PSI-PRED is more accurate in coil residues consistent with the result from datasets of 400, 538 and 3000 proteins. On the other hand, the 2nd CNN' Q3 accuracy keeps surpassing PSI-PRED by 2%, resulting in a remarkable score of 83.6%. The effect of ensemble learning is still obvious in helical (+1.4% based on 1st CNN) and strand (+2.2% based on 1st CNN) residues, while remains insufficient in coil residues (0.5% higher than 1st CNN but 3.1% worse than PSI-PRED).

TABLE IV. PROTEIN SECONDARY STRUCTURE PREDICTION ACCURACY FOR CASP10 SET

Models	Score			
	Q3	*QH*	*QE*	*QC*
PSI-PRED	81.6	79.2	76.0	87.0
1st CNN	82.5	85.1	77.4	83.4
2nd CNN	83.6	86.5	79.6	83.9

IV. DISSCUSSION

We have developed a new secondary structure predictor called Malphite, which achieves 82.6% ten-fold-cross validated accuracy for a dataset of 3000 proteins. And this method makes accurate predictions of 82.3% and 82.6% on independently created testing sets of 400 and 538 proteins respectively. The outstanding performance of Malphite is further confirmed by testing on 122 targets from CASP10, for which a remarkable accuracy of 83.6% is obtained. Malphite consistently makes 2% more accurate prediction than PSI-PRED for the datasets of 400, 538 and 3000 proteins and 122 targets from CASP10, which is a significant step toward the theoretical limit for the prediction accuracy of secondary structure of 90%.

There are two main keys to the success of Malphite.

1) Application of convolution neural network to PSSM processing. Each column array of PSSM, which has been observed to be weighted as a unit by all three neurons of a small secondary structure classifier, can be regarded as an independent feature of the protein amino acid sequence. CNN can extract deep hierarchical features from the input feature maps. The nature of convolution operator, in addition, can smooth noise data and highlight useful features in each feature map from PSSM. As a result, the 1st CNN, even with only one neural network, outperforms the two-stage neural network method, PSI-PRED.

2) Ensemble learning. We have observed that there is a great diversity in the prediction distribution from many existing protein secondary structure predictors. For example, SPINEX is good at predicting helical residues, while PSI-PRED is the expert for predicting coil residues. The result shows that our 1st CNN also has distinct prediction distribution with PSI-PRED. Instead of using traditional ensemble learning algorithms like boosting or bagging, we proposed a new ensemble learning method, which makes use of convolutional neural network by taking predicted probability vectors from base classifiers as input feature maps. Though no serious comparison between the effectiveness of CNN ensembles and traditional ones was made, our CNN ensemble seems to be the most convenient one. Because, by CNN, ensemble learning is done by simply merging the input probability vectors, and the automatic supervised learning will teach CNN how to select the best prediction results from input probability vectors.

Malphite is expandable. Though the current Malphite only has 2 base classifiers, new classifiers can be added by plugging their prediction probability vectors into the 2nd CNN. In future, the performance of Malphite might be further improved by introducing other protein secondary structure predictors that can bring more diverse prediction. Beside prediction diversity, the overall accuracy of new base models is also important. Models that can not bring more diversity end up with adding redundant input features for the 2nd CNN, likewise, models with very poor accuracy only create noise data.

Acknowledgment

The super-computing resource was provided by Human Genome Center (the University of Tokyo).

References

[1] Wu Y, Dousis AD, Chen M, Li J, Ma J, OPUS-Dom: Applying the Folding-Based Method VECFOLD to Determine Protein Domain Boundaries. J Mol Boil 2009, 385:1314-1329.

[2] Kim,D.E. et al. Automated prediction of domain boundaries in CASP6 targets using Ginzu and RosettaDOM. Proteins, 2005, 61 (Suppl. 7), 193–200.

[3] Kihara D. The effect of long-range interactions on the secondary structure formation of proteins. Protein Sci 2005;14:1955–1963.

[4] Moult, J., Hubbard, T., Bryant, S. H., Fidelis, K. & Pedersen, J. T. Critical assessment of methods of protein structure prediction (CASP): round II. Proteins: Struct. Funct. Genet. 1997, S1, 2-6.

[5] Hua, S., Sun, Z., A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. *J. Mol. Biol.* 2001, *308*, 397–407.

[6] Kim, H., Park, H., Protein secondary structure prediction based on an improved support vector machines approach. *Protein Eng.* 2003, *16*, 553–560.

[7] Ward, J. J., McGuffin, L. J., Buxton, B. F., Jones, D. T., Secondary structure prediction with support vector machines. *Bioinformatics* 2003, *19*, 1650–1655.

[8] Guo, J., Chen, H., Sun, Z., Lin, Y., A novel method for protein secondary structure prediction using dual-layer SVM and profiles. *Proteins* 2004, *54*, 738–743.

[9] Duan, M., Huang, M., Ma, C., Li, L., Zhou, Y., Position- specific residue preference features around the ends of helices and strands and a novel strategy for the prediction of secondary structures. *Protein Sci.* 2008, *17*, 1505–1512.

[10] Rabiner, L. R., A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 1989, 275–286.

[11] Asai, K., Hayamizu, S., Handa, K. I., Prediction of protein secondary structure by the hidden Markov model. *Bioin- formatics* 1993, *9*, 141–146.

[12] Aydin, Z., Altunbasak, Y., Borodovsky, M., Protein secondary structure prediction for a single-sequence using hidden semi-Markov models. *BMC Bioinformatics* 2006, *7*, 178.

[13] Qian, N., Sejnowski, T. J., Predicting the secondary structure of globular proteins using neural network models. *J. Mol. Biol.* 1988, *202*, 865–884.

[14] Holley, L. H., Karplus, M., Protein secondary structure prediction with a neural network. *Proc. Natl. Acad. Sci. USA* 1989, *86*, 152–156.

[15] Kneller, D. G., Cohen, F. E., Langridge, R., Improvements in protein secondary structure prediction by an enhanced neural network. *J. Mol. Biol.* 1990, *214*, 171–182.

[16] Rost, B., Sander, C., Prediction of protein secondary struc- ture at better than 70% accuracy. J. Mol. Biol. 1993, 232, 584–599.

[17] Rost, B., Sander, C., Combining evolutionary information and neural networks to predict protein secondary structure. Proteins 1994, 19, 55–72.

[18] Rost, B., PHD: predicting one-dimensional protein structure by profile-based neural networks. Methods Enzymol. 1996, 266, 525–539.

[19] Cuff, J. A., Barton, G. J., Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. Proteins 1999, 34, 508–519.

[20] Jones, D. T., Protein secondary structure prediction based on position-specific scoring matrices. J. Mol. Biol. 1999, 292, 195–202.

[21] E. Faraggi, T. Zhang, Y. Yang, L. Kurgan, and Y. Zhou, "SPINE X: Improving protein secondary structure prediction by multistep learning coupled with prediction of solvent accessible surface area and backbone torsion angles," *Journal of Computational Chemistry,* vol. 33, pp. 259-267, 2012.

[22] Polikar, R. "Ensemble based systems in decision making". IEEE Circuits and Systems Magazine 2006, 6 (3): 21–45.

[23] LeCun, Yann; Léon Bottou; Yoshua Bengio; Patrick Haffner. "Gradient-based learning applied to document recognition". *Proceedings of the IEEE*, 1998, 86 (11): 2278–2324.

[24] Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012

[25] Collobert, Ronan, and Jason Weston. "A unified architecture for natural language processing: Deep neural networks with multitask learning."Proceedings of the 25th international conference on Machine learning. ACM, 2008.

[26] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In International Conference on Computer Vision, pages 2146–2153. IEEE, 2009.

[27] H. Lee, R. Grosse, R. Ranganath, and A.Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In Proceedings of the 26th Annual International Conferenceon Machine Learning, pages 609–616. ACM, 2009.

[28] Y. LeCun, K. Kavukcuoglu, and C. Farabet. Convolutional networks and applications in vision. In Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on, pages 253–256. IEEE, 2010.

[29] Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. Nature, 1986, 323, 533-536.

[30] M. Mohri, A. Rostamizadeh, and A. Talwalkar. Foundations of Machine Learning. The MIT Press, 2012.

[31] Crooks, G.E. and Brenner, S.E. 2004. Protein secondary structure: Entropy, correlations and prediction. *Bioinformatics* 20: 1603–1611.

[32] Simard, P. Y., Steinkraus, D., & Platt, J. 2003. Best practice for convolutional neural networks applied to visual document analysis. In *International Conference on Doc- ument Analysis and Recogntion (ICDAR)* (pp. 958–962). Los Alamitos, CA: IEEE Computer Society

[33] Wang, G. and Dunbrack, R. L., Jr. Pisces: a protein se- quence culling server. *Bioinformatics*, 19(12):1589–91, Aug 12 2003. ISSN 1367-4803 (Print) 1367-4803 (Linking).

[34] Kabsch, Wolfgang and Sander, Christian. Dictionary of protein secondary structure: pattern recognition of hy- drogenbonded and geometrical features. *Biopolymers*, 22(12):2577–2637, 1983. ISSN 1097-0282.

[35] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley and Y. Bengio. “Theano: new features and speed improvements”. NIPS 2012 deep learning workshop.

[36] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley and Y. Bengio. “Theano: A CPU and GPU Math Expression Compiler”. Proceedings of the Python for Scientific Computing Conference (SciPy) 2010. June 30 - July 3, Austin, TX.