Structural Bioinformatics

SAINT: self-attention augmented inception-inside- inception network improves protein secondary structure prediction

Mostofa Rafid Uddin[1,2,†], **Sazan Mahbub**[1,†], **M. Saifur Rahman**[1], and **MD Shamsuzzoha Bayzid**[1,*]

[1]Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka 1205, Bangladesh and [2]Department of Computer Science and Engineering, East West University, Dhaka 1212, Bangladesh
[*]To whom correspondence should be addressed.
Associate Editor: Arne Elofsson
[†] The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.
Associate Editor: Arne Elofsson

Received on December 7, 2019; revised on May 10, 2020; editorial decision on May 12, 2020; accepted on May 16, 2020

Abstract

Motivation: Protein structures provide basic insight into how they can interactwith other proteins, their functions and biological roles in an organism. Experimentalmethods (e.g. X-ray crystallography and nuclearmagnetic resonance spectroscopy) for predicting the secondary structure (SS) of proteins are very expensive and time consuming. Therefore, developing efficient computational approaches for predicting the SS of protein is of utmost importance. Advances in developing highly accurate SS prediction methods have mostly been focused on 3-class (Q3) structure prediction. However, 8-class (Q8) resolution of SS contains more useful information and ismuchmore challenging than the Q3 prediction.
Results: We present SAINT, a highly accurate method for Q8 structure prediction, which incorporates self-attention mechanism (a concept from natural language processing) with the Deep Inception-Inside-Inception network in order to effectively capture both the short- and long-range interactions among the amino acid residues. SAINT offers a more interpretable framework than the typical black-box deep neural network methods. Through an extensive evaluation study, we report the performance of SAINT in comparison with the existing best methods on a collection of benchmark datasets, namely, TEST2016, TEST2018, CASP12 and CASP13. Our results suggest that self-attention mechanism improves the prediction accuracy and outperforms the existing best alternate methods. SAINT is the first of its kind and offers the best known Q8 accuracy. Thus, we believe SAINT represents a major step toward the accurate and reliable prediction of SSs of proteins.
Availability and implementation: SAINT is freely available as an open-source project at https://github.com/SAINTProtein/SAINT.
Contact: shams.bayzid@gmail.com

1 Introduction

Proteins are bio-molecules made of long chains of amino acid residues connected by peptide bonds. The functions of proteins are usually determined by their tertiary structure and for determining the tertiary structure and related properties, the secondary structure (SS) information is crucial. Protein structure can be experimentally determined by X-ray crystallography and multi-dimensional magnetic resonance in laboratory, but these methods are very costly and time consuming and are yet to be consistent with the proliferation of protein sequence data (Jiang et al., 2017). Thus, the proteins with known primary sequence continue to outnumber the proteins with experimentally determined SSs. The structural properties of a protein depend on its primary sequence (Anfinsen, 1973; Baker and Sali, 2001; Bradley et al., 2005; Dill et al., 2008), yet it remains as a difficult task to accurately determine the secondary and tertiary structures of proteins. Hence, the problem of predicting the structures of a protein — given its primary sequence — is crucially important and remains as one of the greatest challenges in computational biology. SS — a conformation of the local structure of the polypeptide backbone — prediction dates back to the work of Pauling and Corey in 1951 (Pauling, 1951).

2 Approach

2.1 Feature representation

SAINT takes a protein sequence feature vector $X = (x_1, x_2, x_3, ..., x_N)$ as input, where x_i is the vector corresponding to the ith residue, and it returns the protein structure label sequence vector $Y = (y_1, y_2, y_3, ..., y_N)$ as output, where y_i is the structure label (one of the eight possible states) of the ith residue. Similar to SPOT-1D-base and MUFOLD-SS, our base model contains 57 features from PSSM profiles, HHM profiles and physicochemical properties. To generate PSSM, PSI-BLAST (Altschul et al., 1997) was run against Uniref90 database (Consortium, 2007) with inclusion threshold 0.001 and three iterations. The HHM profiles were generated using HHblits (Remmert et al., 2012) using default parameters against uniprot20_2013_03 sequence database. HHblits also generates seven transition probabilities and three local alignment diversity values, which we used as features as well. Seven physicochemical properties of each amino acid [e.g. steric parameters (graph-shape index), polarizability, normalized van der Waals volume, hydrophobicity, isoelectric point, helix probability and sheet probability were obtained from (Meiler et al., 2001). So, in our base model, the dimension of x_i is 57 as this is the concatenation of $x_h hm_i \in \mathbb{R}^{d_{hhm}} (d_{hhm} = 30), x_{pss} m_i \in \mathbb{R}^{d_{pssm}} (d_{pssm} = 20)$ and $x_{physical_i} \in \mathbb{R}^{d_{physical}} (d_{physical} = 7)$. Additional features were generated by windowing the predicted contact information as was done in SPOT-1D. The contact maps were generated using SPOT-Contact (Hanson et al., 2018) and were used as our features by varying window lengths (the number of preceding or succeeding residues whose pairwise contact information were extracted for a target residue). Our ensemble model constitutes of four different models, that we trained with varying input features: one without the contact maps (base model) and three with different window lengths (10, 20 and 50) of the contact-map-based features. The features were normalized to ensure 0 mean and SD of 1 in the training data, similar to SPOT-1D.

2.2 Architecture of SAINT

The architecture of SAINT can be split into three separate discussions: (i) the architecture of our proposed self-attention module, (ii) the architecture of the existing inception module and the proposed attention-augmented inception module and finally (iii) the overall pipeline of SAINT.

2.2.1 Self-attention module

Attention mechanism implies paying attention to specific parts of input data or features while generating output sequence (Bahdanau et al., 2014; Vaswani et al., 2017). It calculates a probability distribution over the elements in the input sequence and then takes the weighted sum of those elements based on this probability distribution while generating outputs.In self-attention mechanism (Cheng et al., 2016; Parikh et al., 2016; Vaswani et al., 2017), each vector in the input sequence is transformed into three vectors- query, key and value, by three different functions. Each of the output vectors is a weighted sum of the value vectors, where the weights are calculated based on the compatibility of the query vectors with the key vectors by a special function, called compatibility function (discussed later in this section). The self-attention module, we designed and augmented with the Deep3I network (Fang et al., 2018) is inspired from the selfattention module proposed by (Vaswani et al., 2017) and is depicted in Figure 1a. Our self-attention module takes two inputs: (i) the features from the previous inception module or layer, $x \in \mathbb{R}^{d_{protein} \times d_{feature}}$ and (ii) position identifiers, $pos_i d \in \mathbb{R}^{d_{protein}}$, where $d_{protein}$ is the length of the protein sequence and $d_{feature}$ is the length of the feature vector.

2.2.1.1 Positional Encoding Sub-module. The objective of positional encodings is to inject some information about the relative or absolute positions of the residues in a protein sequence. The Positional Encoding PosEncp for a position p can be defined as follows (Vaswani et al., 2017).

$$PosEnc_{(p,2i)} = \sin(p/10000^{2i/d_{feature}}) \quad (1)$$

$$PosEnc_{(p,2i+1)} = \cos(p/10000^{2i/d_{feature}}), \quad (2)$$

where i is the dimension. We used such function as it may allow the model to easily learn to attend by relative positions since for any fixed offset $k, PosEnc_{(p,k)}$ can be represented as a linear function of $k, PosEnc_p$ (Vaswani et al., 2017). $p, PosEnc_p$ has the dimension $d_{protein} \times d_{feature}$. The output of positional encoding is added with the inputs x, esulting in new representations [see Equation (3)]

$$h_{pos} = x_{pos} + PosEnc_{pos}. \quad (3)$$

2.2.1.2 Scaled dot-product attention sub-module. The input features in this sub-module, $h \in \mathbb{R}^{d_{protein} \times d_{feature}}$ are first transformed into three feature spaces Q, K and V, representing query, key and value, respectively, in order to compute the scaled dot-product attention, where $Q(h) = W_Q h, k(h) = W_K h, V(h) = W_V h$. Here, W_Q, W_K, W_V are parameter matrices to be learned. Figure 1b shows a schematic diagram of this module.

Among various compatibility functions [e.g. scaled dot-product attention (Vaswani et al., 2017), additive attention (Bahdanau et al., 2014), similarity-attention (Graves et al., 2014), multiplicative-attention (Luong et al., 2015), biased general attention (Sordoni et al., 2016), etc., we have chosen the scaled dot-product attention as it showed much promise in case of sequential data. (Vaswani et al., 2017) showed that in practice,

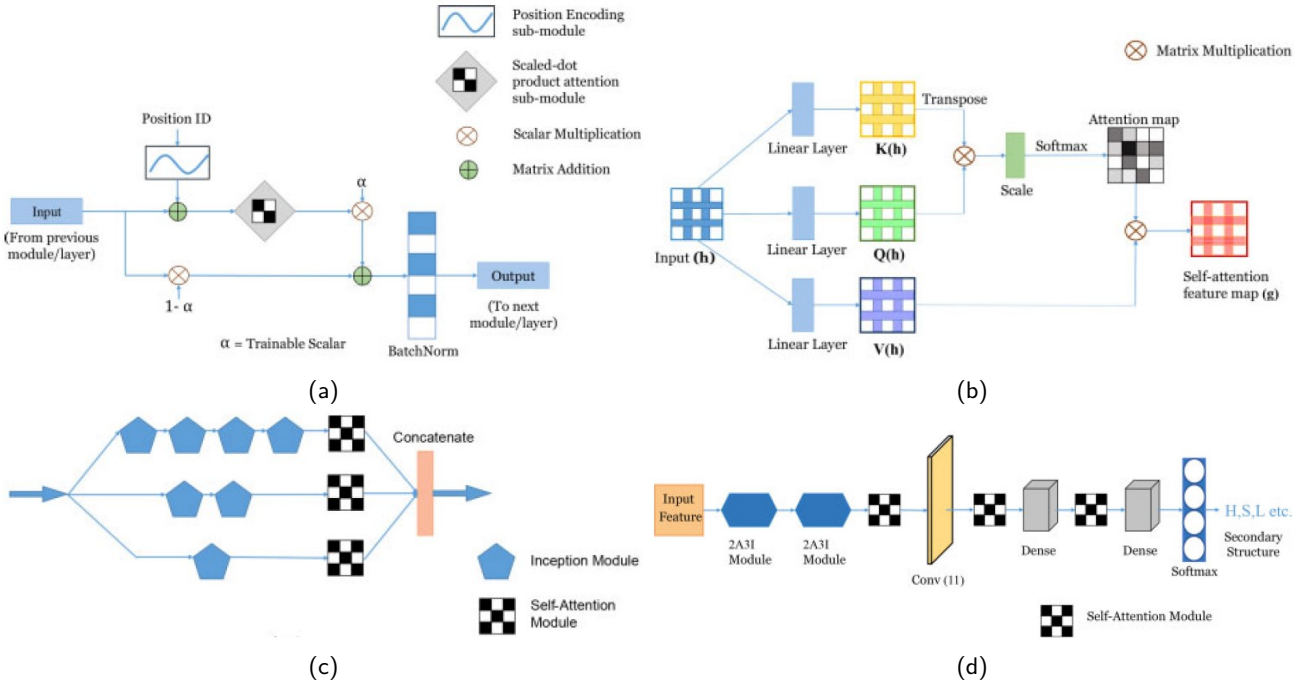

Fig. 1: Schematic diagrams of SAINT and its various components. (a) Architecture of the self-attention module used in SAINT. (b) Architecture of the scaled dot-product attention sub-module. (c) Architecture of our proposed 2A3I module by augmenting self-attention within the 3I network. (d) A schematic diagram of the overall architecture of SAINT, which comprises two 2A3I modules, three self-attention modules, convolutional layers with window size 11 and 2 dense layers

the dot-product attention is much faster and space-efficient as it can be implemented using highly optimized matrix multiplication code, though theoretically both dot-product and additive attention have similar complexity. Scaled dot-product s i;j of two vectors h i and h j is calculated as shown in Equation 4.

$$s_{i,j} = \frac{Q(h_i)K(h_i)^T}{\sqrt{d_K}} \tag{4}$$

where d_K is the dimension of the feature space K. The numerator of the equation, $Q(h_i)K(h_i)^T$ is the dot product between these two vectors, resulting in the similarity between them in a specific vector space. Here, $\sqrt{d_K}$ is the scaling factor, which ensures that the result of the dot product does not get prohibitively large for very long sequences.

The attention weights $e \in \mathbb{R}^{d_{protein} \times d_{feature}}$ are calculated as shown in Equation 5, where $e_{i,j}$ represents how much attention have been given to the vector at position i while synthesizing the vector at position j

$$e_{i,j} = \frac{exps_{i,j}}{\sum_{n=1}^{d_{protein}} exp(s_{i,j})} \tag{5}$$

The attention distribution e is multiplied with the feature vectors $V(h)$ and then in order to reduce the internal covariate shift, this multiplicand is normalized using *batch normalization* (Ioffe and Szegedy, 2015), producing g, the output of the scaled dot-product attention sub-module, following the Equation 6.

$$g_j = BatchNorm(\sum_{n=1}^{d_{protein}} e_{i,j}V(h_i)). \tag{6}$$

Here, $BatchNorm(.)$ is the batch-normalization function and g_j is the j-th vector in the output sequence of this sub-module. Finally, according to the Equation 7, g is multiplied by a scalar parameter α, the original input feature map x is multiplied by $(1-\alpha)$ and these two multiplicands are summed to synthesize the final output y.

$$y_i = (\alpha)g_i + (1-\alpha)x_i, \tag{7}$$

where y_i is the ith output and α is a learnable scalar. By introducing weighed sum of g_i and x_i, we give our model the freedom to choose how much weight should be given to each of the features maps, g_i and x_i while generating the output y_i. The optimal value of the parameter α is learnt through back propagation along with the rest of the model.

3 Results and discussion

3.1 Results on benchmark dataset

Table 1: Statistical significance of the Q8 accuracy between SAINT and other state-of-the-art methods

Method	TEST2016 (1213)	TEST2018 (250)	CASP13 (31)	CASP12 (49)	CASP-FM (56)
SPOT-1D	$8.168e^{-27}$	$3.893e^{-5}$	0.101	0.0345	0.0791
NetSurfP-2.0	$2.607e^{-57}$	$3.258e^{-18}$	0.179	$1.55e^{-6}$	0.0001
MUFOLD-SS	$1.531e^{-88}$	$3.145e^{-21}$	0.179	$6.51e^{-5}$	0.005

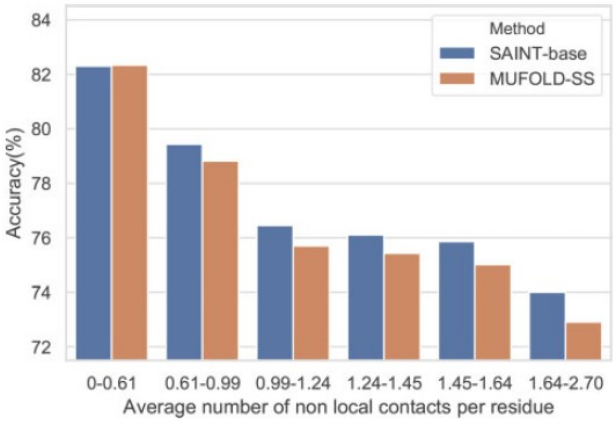

Fig. 2: Accuracy of SAINT-base and MUFOLD-SS under various levels of non-local interactions. We show the results on the TEST2016 test set using six bins of proteins as shown in Table 1

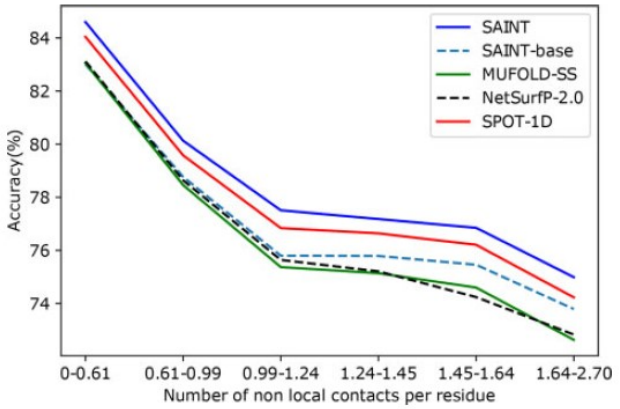

Fig. 3: Accuracy of SAINT, SPOT-1D, NetSurfP-2.0 and MUFOLD-SS as a function of the average number of non-local interactions per residue. We show the results on the six bins as shown in Table 1

3.2 Running time

SAINT is much faster than the best alternate method SPOT-1D. For generating the structures of 1213 protein chains in TEST2016, given the necessary input files, SAINT took $\sim 360 \pm 5$ s whereas SPOT-1D took $\sim 2485 \pm 5$ s on our local machine [Intel core i7-7700 CPU (4 cores), 16 GB RAM, NVIDIA GeForce GTX 1070 GPU]. Under the same settings, SAINT took $\sim 197 \pm 5$ s to generate SSs for the 250 proteins in TEST2018, whereas SPOT-1D took $\sim 668 \pm 5$ s. Since both these methods use the same input files for feature generation, this substantial difference in running time can be attributed to the efficiency of our attention based method over the LSTM networkbased model used in SPOT-1D.

Acknowledgement

The authors thank the anonymous reviewers for their insightful comments and suggestions, and the authors of SPOT-1D for providing the PSSMs, HMMs and contact maps of the proteins in the training dataset.

References

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–3402.

Anfinsen, C. B. (1973). Principles that govern the folding of protein chains. *Science*, 181(4096):223–230.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Baker, D. and Sali, A. (2001). Protein structure prediction and structural genomics. *Science*, 294(5540):93–96.

Bradley, P., Misura, K. M., and Baker, D. (2005). Toward high-resolution de novo structure prediction for small proteins. *Science*, 309(5742):1868–1871.

Cheng, J., Dong, L., and Lapata, M. (2016). Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*.

Consortium, U. (2007). The universal protein resource (uniprot). *Nucleic acids research*, 36(suppl_1):D190–D195.

Dill, K. A., Ozkan, S. B., Shell, M. S., and Weikl, T. R. (2008). The protein folding problem. *Annu. Rev. Biophys.*, 37:289–316.

Fang, C., Shang, Y., and Xu, D. (2018). Mufold-ss: New deep inception-inside-inception networks for protein secondary structure prediction. *Proteins: Structure, Function, and Bioinformatics*, 86(5):592–598.

Graves, A., Wayne, G., and Danihelka, I. (2014). Neural turing machines. *arXiv preprint arXiv:1410.5401*.

Hanson, J., Paliwal, K., Litfin, T., Yang, Y., and Zhou, Y. (2018). Accurate prediction of protein contact maps by coupling residual two-dimensional bidirectional long short-term memory with convolutional neural networks. *Bioinformatics*, 34(23):4039–4045.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr.

Jiang, Q., Jin, X., Lee, S.-J., and Yao, S. (2017). Protein secondary structure prediction: A survey of the state of the art. *Journal of Molecular Graphics and Modelling*, 76:379–402.

Luong, M.-T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Meiler, J., Müller, M., Zeidler, A., and Schmäschke, F. (2001). Generation and evaluation of dimension-reduced amino acid parameter representations by artificial neural networks. *Molecular modeling annual*, 7(9):360–369.

Parikh, A. P., Täckström, O., Das, D., and Uszkoreit, J. (2016). A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*.

Pauling (1951). The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. 37:205–211.

Remmert, M., Biegert, A., Hauser, A., and Söding, J. (2012). Hhblits: lightning-fast iterative protein sequence searching by hmm-hmm alignment. *Nature methods*, 9(2):173–175.

Sordoni, A., Bachman, P., Trischler, A., and Bengio, Y. (2016). Iterative alternating neural attention for machine reading. *arXiv preprint arXiv:1606.02245*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.