Empirical Study of Protein Feature Representation on Deep Belief Networks Trained With Small Data for Secondary Structure Prediction

Shamima Rashid, Suresh Sundaram, and Chee Keong Kwoh

Abstract—Protein secondary structure (SS) prediction is a classic problem of computational biology and is widely used in structural characterization and to infer homology. While most SS predictors have been trained on thousands of sequences, a previous approach had developed a compact model of training proteins that used a **C**-**A**lpha, C-**B**eta **S**ide Chain (**CABS**)-algorithm derived energy based feature representation. Here, the previous approach is extended to Deep Belief Networks (DBN). Deep learning methods are notorious for requiring large datasets and there is a wide consensus that training deep models from scratch on small datasets, works poorly. By contrast, we demonstrate a simple DBN architecture containing a single hidden layer, trained only on the CB513 dataset. Testing on an independent set of G Switch proteins improved the Q_3 score of the previous compact model by almost 3%. The findings are further confirmed by comparison to several deep learning models which are trained on thousands of proteins. Finally, the DBN performance is also compared with **P**osition **S**pecific **S**coring **M**atrix (*PSSM*)-profile based feature representation. The importance of (i) structural information in protein feature representation and (ii) complementary small dataset learning approaches for detection of structural fold switching are demonstrated.

Index Terms—Deep belief networks, protein secondary structure, small dataset, G switch proteins

1 Introduction

Protein secondary structure (SS) is used to infer evolutionary and functional relationships for newly determined sequences. Secondary structures can guide phylogentic tree reconstruction [1], [2], assess binding site fit to ligands [3], [4] and guide atomic placements for contact prediction maps [5]. As whole genome sequencing drives exome discovery, rapid and effective methods for SS prediction that can accurately model structural changes or fold switches at point mutated sites, are a powerful complement to experimental approaches.

Pioneering early SS prediction works were the GOR group of methods that used information entropy functions defined by residue frequencies [6], [7], [8] and the development of a sliding window scheme to account for residue correlations [9]. Next, profiles by PHD with multiple sequence alignments [10] and PSI-BLAST [11] derived position specific scoring matrices (PSSM) used in PSIPRED [12] improved prediction accuracy. PHD was also the first to introduce a second level structure to structure neural network to refine SS prediction. This two-level network architecture was effective and adopted in later works like Spine [13] and Porter [14] that both utilized two level bidirectional recurrent NN (BRNN). However, none of these methods obtained a three-state SS accuracy (Q_3) greater than 80%. Besides the use of sequence based protein profiles and various network architectures, the incorporation of structural alignments of related templates was another attempted direction of improvement. Several notable works that included structural alignments from homologous templates were Proteus [15], Porter with homology [16], and SSpro [17]. But again, if only the template-free predictions were to be considered, the Q_3 did not exceed 80%.

The upper limit for Q_3 is theoretically estimated to be between 88-90% [18], [19]. Recently, large protein datasets and data driven deep learning methods have obtained Q_3 as high as 82%-84% [20]. SPIDER 2 developed iterative deep learning networks and included predicted solvent accessibility and torsion angles to refine SS prediction [21]. In the next development, SPIDER 3 employed 4 levels of iterative BRNN, comprising 2 layers of long short-term memory (LSTM) network cells each. SPIDER 3 obtained a Q_3 accuracy of 84.16% and was trained on more than 4000 proteins [22]. Porter 5.0 implemented cascaded BRNN and convolutional layers to predict SS and reported Q_3 close to 82% [23]. Spencer *et. al.*, used the summed predictions of two first level deep belief networks (DBN) fed into a second level network [24]. In the last few years, deep convolutional neural networks (CNN) have

- *Shamima Rashid and Chee Keong Kwoh are with Nanyang Technological University, Singapore 639798. E-mail: sham0012@e.ntu.edu.sg, asckkwoh@ntu.edu.sg.*
- *Suresh Sundaram is with Nanyang Technological University, Singapore 639798, and also with the Indian Institute of Science, Bangalore, Karnataka 560012, India. E-mail: sureshsundaram@iisc.ac.in.*

Manuscript received 29 Jan. 2021; revised 14 Nov. 2021; accepted 8 Apr. 2022. Date of publication 19 Apr. 2022; date of current version 3 Apr. 2023.
This work was supported by the Ministry of Education, Singapore, under Grants Tier 1 2020-T1-001-130(RG15/20) and Tier 2 MOE2019-T2-2-175. (Corresponding author: Shamima Rashid.)
Digital Object Identifier no. 10.1109/TCBB.2022.3168676

been used to capture the spatial relationship in protein sequences. RaptorX modeled the input and output layers as conditional random fields with convolutional hidden layers in a deep neural fields model [25]. eCRRNN used an ensemble BRNN containing gated recurrent units with residual connections, cascaded with convolutional blocks [26], with a reportedly high Q_3 of 87.3%. SecNet presented a standard CNN 4 convolutional layers deep, obtaining a Q_3 score of 84.3% [27]. Recently, DNSS2 integrated layers of diverse network types including convolutional, residual, and recurrent layers among others and reported a Q_3 of 84.6% [28]. Other deep learning models include SAINT that employed an attention mechanism with inception networks for 8-state SS prediction [29]. All of these approaches used the largest datasets available for training, often consisting of thousands of proteins. The SS prediction problem attracts such a volume of research, that it is beyond the scope of this paper to treat them in the thorough manner they deserve. Excellent reviews of SS prediction methods are given in [20], [30].

The above discussed methods, have used PSSM-profile based features often in conjunction with various protein property descriptors. These are Meiler's [31] parametric representation of amino acid characteristics (as used in [26]), Atchley factors [32] (as used in [24]) and predicted torsion angles to iteratively refine predictions (such as SPIDER2). SecNet and Porter 5.0 used hidden Markov Model (HMM) derived sequence profiles using HHBlits [33] in addition to PSSM-profiles. All methods and the deep learning network models in particular, have been trained on the largest datasets available, often containing thousands of proteins [34].

It seems indisputable in deep learning discourse that training deep models on small datasets from scratch is futile. To address the lack of data issue, one common approach is to add a pre-trained model's weights from related scenarios for which large datasets are available, known as *transfer learning*. Hence, a majority of research involving prediction on small datasets has focused on transfer learning, where the pre-trained network weights of related source scenarios are used as initial layer weights. Other synthetic data approaches such as minority oversampling are also used for a small dataset scenario.

However, these approaches have their own drawbacks. For instance, training a model on thousands of proteins, or strategies such as transfer learning that assume relations between distinct datasets can be problematic in at least two aspects. First, the failure to generalize to new proteins that have distinct structural segments [34] and second, an assumption that the source domain is related to the target domain that results in a high accuracy but loses biologically meaningful explanations. Good generalization from limited data is the hallmark of true intelligence [35]. Given huge amounts of data, even a weak model can memorize the relations [36].

Although transfer learning may be effective in fields such as image recognition which require that shape or pixel intensity relations are upheld between source and target, similar assumptions of relations between distinct protein folds may be problematic. For instance, convergent sequences may not share the same fold. In some fields such as precision medicine, big data would be meaningless. Given these considerations, works aimed at learning from small datasets without external data remain surprisingly scarce.

A precise definition of "small dataset" remains elusive because effective dataset size can differ greatly depending on the chosen problem. In a previous definition, a small dataset was taken as the highest number of training proteins, beyond which the Q_3 score improved no further and denoted as the compact model of 55 proteins (SSP_{55}) [34]. Using a Fully Complex-valued Relaxation Network (FCRN) trained on SSP_{55}, better performance compared to predictors trained on thousands of SS proteins, had been demonstrated [34]. The rationale behind the compact model method, was to adopt a heuristically approximate solution to find the minimal set of training proteins needed to achieve the maximum performance on a dataset. As 55 training proteins would be insufficient to train a deep belief network (and as the results here would prove) here, the DBN is constrained to learn from 385 homology reduced proteins in the benchmark CB513 dataset [37].

These 385 CB513 proteins are henceforth defined as a "small dataset" in the rest of this paper, which is used as a training dataset. An extremely simple but surprisingly effective deep belief network (DBN) model trained on previously generated **C**-**A**lpha, C-**B**eta **S**ide Chain (*CABS*)-algorithm derived energy profiles is proposed. The DBN is separately trained on PSSM-profile based feature representation for comparison. In [24], the DBN architecture consisted of the summed predictions of two first tier networks being sent to a second tier for refined predictions. Here, a single DBN predicts the SS given CABS-algorithm based energy profile feature representation. However for the PSSM-profile based feature representation, a second tier DBN is added to improve the predicted accuracy. The architecture is kept simple to reduce the computational complexity of the model, since the requirement is to learn from a small training dataset. Although recently CNN and LSTM based architectures have emerged as two popular choices for exploiting the sequential and spatial relationships inherent in protein sequences, here DBN were selected for their natural relationship to physical energy systems.

DBN are modeled with the Boltzmann energy function and hence have a natural fit to the CABS-algorithm derived energy potentials of protein structures. Minimizing the joint energy of the Boltzmann equation (Eq. (2)) is analogous to finding the optimal hidden parameters that correspond to the direction of the lowest energy protein structure in the energy landscape. This gives DBN indirect physical support for the prediction of protein secondary structures.

The results of DBN trained with CABS-algorithm derived features show substantial improvement over other SS predictors on a blind test dataset of G Switch proteins. An improvement of Q_3 by 2.86% over the previously developed compact training model containing 55 training proteins (abbreviated as SSP_{55} -FCRN) [34] is also shown. In the rest of the paper, the term pre-training refers to the unsupervised learning of weights in the DBN and does not refer to transfer learning.

The paper is organized as follows. Section 2 presents the datasets and describes the PSSM-profile based and CABS-algorithm derived feature representations used. The theory and architecture of deep belief networks and the joint energy

conformation with the Boltzmann equation used to model the SS prediction problem is presented. Section 3 reports the experimental results of 5-fold cross-validation on CB513 and compares the network performance on the independent blind test dataset (GSW25) against those of several recent deep learning methods. The implications of the findings are discussed in Section 4 and finally, Section 5 concludes the work.

2 Methods

Deep belief networks were trained with two types of protein feature representations to predict secondary structures. The two feature representations compared are (i) PSSM-profile based features commonly used by SS predictors and (ii) the C-Alpha, C-Beta Side Chain (CABS)-algorithm derived energy potentials based features that had previously been generated [38].

2.1 Datasets

CB513

The CB513 dataset developed by Cuff and Barton in 1999 [37] is used for training. Of these, 128 chains found homologous to CATH structural templates used in the generation of energy potentials, had been removed in an earlier work [38]. Section 2.2.2 describes the procedure. After homology removal, the final dataset contained 385 proteins comprising 63,079 residues. For five-fold cross-validation, this homology reduced set was randomly divided into train and test sets, containing 80% and 20% proteins respectively.

G Switch Proteins (GSW25)

This dataset was developed by [38] and contains 25 protein chains, derived from the G_A and G_B domains of the bacterial *Streptococcus* G protein [39], [40]. The G_A and G_B domains adopt a 3α and $4\beta + \alpha$ fold respectively. It is considered a challenging dataset because a single site mutation of Lysine to Tyrosine (K45Y) plays a role in the switching of folds. A series of experiments have indicated that the K45Y triggers a switch between the G_A and G_B folds [40]. While GSW25 contains similar sequences, it was strictly used in blind test experiments. The list of sequences is given in the Appendix.

Homology Removal and Independence of Train and Test Datasets

Careful steps were taken to ensure that the CB513 and GSW25 datasets are independent sequentially and structurally. Any related folds to GSW25 were excluded from training and DBN model development of the blind test results presented in Table 5. First, homologous templates to CB513 proteins had already been removed in the generation of CABS-algorithm derived energy profiles, resulting in 385 proteins. This process is described in Section 2.2.2 and in detail in [38].

Second, the sequence identity between GSW25 and the CB513 dataset proteins was checked to be below 25% using the PISCES sequence culling server [41]. Third, to detect structural homologues to GSW25, the CB513 proteins were annotated with their respective SCOP folds [42] which identified two CB513 proteins with similar structural folds to GSW25. This was the β-Grasp ubiquitin-like fold to which both G_A and G_B domains belong, according to the SCOPe database version 2.07 [43]. Hence, in reporting the DBN performance in Table 5, a separate five-fold cross-validation model was developed, in which the two related CB513 proteins were not used in training and were restricted to the test partitions.

TABLE 1
Secondary Structure Composition of Datasets

Dataset	Proteins	Residues	Classes (%)
CB513	385	63,079	H (35), E (23), C (42)
GSW25	25	1400	H (52), E (39), C (9)

Table 1 shows the breakdown by secondary structure composition of both datasets. The SS had been assigned in [38] using the Dictionary of Protein Secondary Structure program (DSSP) [44] following the common 8 to 3 state reduction rule used in other works [13], [27]. States H, G and I corresponding to α, 3_{10} and π helices were reduced to Helix (H). States E (extended strand) and B (isolated β-bridge) were reduced to Sheet (E). Lastly, states T and S (containing β-turns, loops, irregular structures and inclusive of blanks) were reduced to Coil (C).

2.2 Protein Feature Representation

To study the effect of protein feature representation on deep learning, position specific scoring matrices (PSSM) and CABS-algorithm derived energy potentials were compared.

2.2.1 PSSM-Profile Based Features

Position **S**pecific **I**terative-**B**asic **L**ocal **A**lignment **S**earch **T**ool or PSI-BLAST [11] is a commonly used method to generate protein sequence profiles from an alignment of related proteins. Distantly related protein sequences are first detected by querying a database and a local alignment is dynamically constructed. The E-value or expectation score serves as a threshold for the sequences that are included in the alignment. The probability of an amino acid occurring at a given position in the query sequence for each column in the alignment is then calculated and converted to log-likelihood scores. The resultant matrix profile generated is termed **P**osition **S**pecific **S**coring **M**atrix (or *PSSM*, an example of which is shown in Fig. 1.

PSSM-profiles were constructed for the CB513 and GSW25 datasets by querying target proteins against the *RefSeq* database [45]. To avoid hits to low complexity regions, transmembrane and coiled coil segments, the pfilt program [1] from the PSIPRED server [46] was first used to filter the database. The PSI-BLAST program from the BLAST+ toolbox [47] was run for 3 iterations with an e-value cut-off of 10^{-5} and SEG filtering set to yes. All other values were left as default. For a rationale of the choice of settings and further experiments on an additional set of PSSM-profiles generated with legacy blast (not reported), please refer to Section 3.1.

2.2.2 CABS-Algorithm Based Features

The CABS-algorithm is a lattice model to calculate reference energies of protein alignments using knowledge-based statistical potentials in its force-field computation [48]. Potentials

1. http://bioinfadmin.cs.ucl.ac.uk/downloads/pfilt/

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-3	-2	0
T	0	-1	0	-1	-1	-1	-1	-2	-1	-2	-1	-1	-1	-2	-1	2	4	-3	-2	0
Y	-2	-2	-2	-3	-3	-2	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1
K	1	2	0	-1	-2	1	0	-1	-1	-3	-2	4	-1	-3	-1	0	-1	-3	-2	-2
L	-2	-2	-4	-4	-1	-2	-3	-4	-3	1	4	-3	2	0	-3	-3	-1	-2	-1	1
I	-1	-3	-3	-3	-1	-2	-3	-3	-3	3	1	-2	1	-1	-3	-2	0	-3	-1	4
L	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-2	-1	2
N	-1	1	3	0	-3	1	0	-1	0	-3	-3	4	-2	-3	-1	0	0	-3	-2	-3
L	0	-2	-1	-2	-2	-2	-2	4	-2	-1	2	-2	0	-1	-3	-1	-2	-2	-2	-1
K	2	1	-1	-1	-2	2	1	-1	-1	-2	-2	4	-1	-3	-1	0	-1	-3	-2	-2
Q	-1	-1	3	0	-2	2	0	-1	-1	-2	-2	0	-1	-3	-1	1	4	-3	-2	-1
A	2	-2	-3	-3	-1	-2	-2	-2	-2	0	2	-2	1	3	-2	-1	-1	-1	0	0
K	0	1	0	-1	-2	1	0	-1	-1	-2	-2	4	-1	-3	-1	2	2	-3	-2	-2
E	0	-1	0	0	-3	0	3	5	-1	-4	-3	-1	-3	-3	-2	0	-1	-3	-3	-3
E	-1	-1	0	3	-3	1	4	-2	-1	-3	-3	0	-2	-3	-1	0	1	-3	-2	-2
A	2	-1	-1	-1	-1	-1	-1	-1	-2	-1	-1	-1	-1	-2	-1	1	4	-3	-2	0
I	1	-2	-1	-2	-1	-1	-2	-2	-2	2	0	-1	0	-1	-2	1	3	-3	-2	1
K	0	1	0	-1	-2	0	0	-2	-1	-1	-2	3	-1	-3	-1	0	3	-3	-2	0
E	-1	1	0	1	-3	2	4	-2	0	-3	-3	3	-2	-3	-1	0	-1	-3	-2	-3
L	3	-2	-2	-2	-1	-1	-2	-1	-2	0	2	-1	0	-1	-2	0	-1	-2	-2	0

Fig. 1. Diagram of a Position Specific Scoring Matrix (PSSM). The first column contains the residues of the query sequence. The first row indicates the 20 possible amino acid types. A given matrix element (m_{ij}) represents the log-likelihood score $\log(p_{ij}/b_{ij})$, where p_{ij} is the probability of amino acid at row i being of the type indicated at column j. b_{ij} denotes the background score of amino acid at position i being substituted for an amino acid at position j in a substitution matrix (such as BLOSUM62).

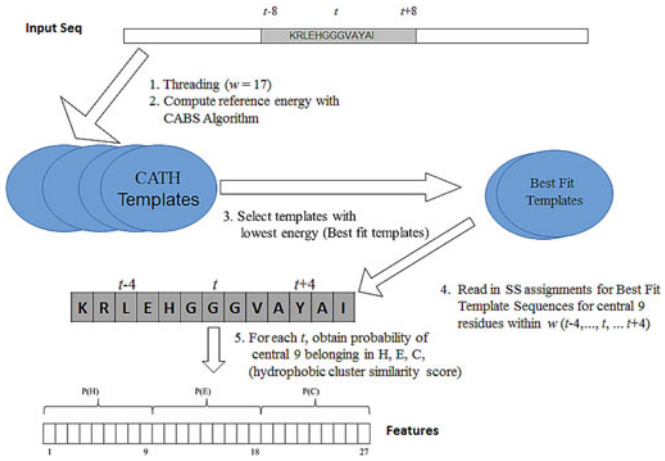


Fig. 2. Representation of features. A target residue, t in the input sequence is represented as a 27-dimensional feature vector. Reproduced from [34], under Open Access permission of BMC Bioinformatics.

data for CB513 and GSW25 had been generated in a previous work [38] and the method is briefly described here. Threading was used to align a target protein to a library of CATH templates [49] and the reference energy computed with the CABS-algorithm. The probabilities of a residue adopting each of the three SS states (H, E or C) was calculated with a scoring function [50]. A target protein residue was represented as a 27-dimensional vector, with the first 9 containing probabilities to adopt Helix (denoted P(H)), the next 9, the probabilities of adopting Sheet (denoted P(E)) and the final 9, those of adopting Coil (denoted P(C)). Fig. 2 shows the process. More details are available in [34], [38].

Removal of Highly Similar Templates

Around 1000 CATH [49] templates had been downloaded and aligned against CB513 targets with the Needleman-Wunsch global pairwise alignment [51]. 97% of alignments had similarity scores lower than 20%. Structural similarities between CB513 and CATH templates had been removed by checking target names against Homology-derived Secondary Structure of Proteins (HSSP) [52]. Following the removal of sequence and structural similarities, 422 CATH templates and 385 CB513 proteins had been obtained for threading and reference energy computation [38]. DSSP secondary structures were assigned to templates and heavy atom contact maps were computed before threading.

Threading and Computation of Reference Energy

Target to template alignment and threading had been performed using a window of size of 17 residues and the CABS-algorithm used to compute the reference energies [38]. The reference energy function takes short and long range contacts into account, based on attributes such as geometric and chemical complementarity and adds a contact energy term for the long-range interactions [48]. SS assignments from best fitting (lowest energy) templates were read in for the central 9 residues within the window of 17. Probabilities of adopting H, E or C were then calculated by a hydrophobic cluster similarity method from the templates [50]. This resulted in a 27-dimensional feature vector for each protein residue, as illustrated in Fig. 2. For full details on the generation of energy potentials, refer to [38]. An alternative summary is available in [34].

2.3 Deep Belief Networks

The set of observations to predict protein secondary structures is defined as $\{(\mathbf{x^1}, \mathbf{y^1}), \ldots, (\mathbf{x^t}, \mathbf{y^t}), \ldots, (\mathbf{x^N}, \mathbf{y^N})\}$, where $\mathbf{x^t} \in \Re^{\mathbf{m}}$ are the m-dimensional input features describing the structure of the t^{th} residue. Also, $\mathbf{y^t} \in \Re^{\mathbf{n}}$ are the n-dimensional coded class labels. N denotes the total number of protein residues. The coded class labels $\mathbf{y^t}$ are obtained by

$$y_j^t = \begin{cases} 1 & \text{if } c^t = j \\ -1 & \text{otherwise} \end{cases} j = \{1, 2, 3\} \tag{1}$$

where c^t is the numeric class {1, 2, 3} and corresponds to H, E and C respectively. Formally, the prediction of secondary structures is defined as estimating the functional relationship $F : \mathbf{x^t} \in \Re^{\mathbf{m}} \rightarrow \mathbf{y^t} \in \Re^{\mathbf{n}}$.

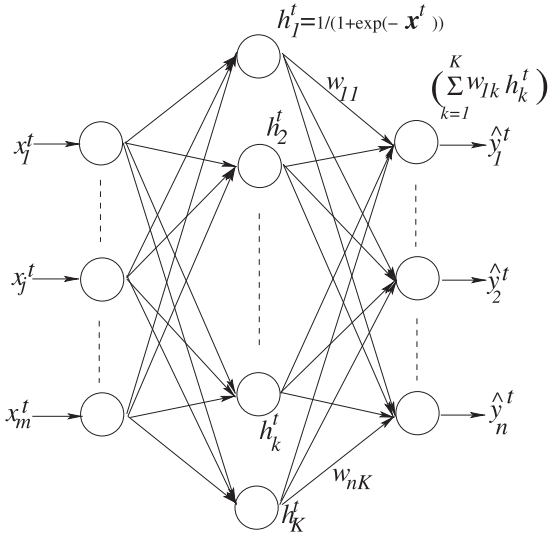
Theory and Architecture. A DBN consists of one or more Restricted Boltzmann Machines (RBM) stacked in a layer-wise manner [53]. In an RBM, the input energy potentials and hidden neurons are modelled with the energy configuration E($\mathbf{x}$, $\mathbf{h}$), shown in Eq. (2). Here, $\mathbf{x}$ is the visible input layer consisting of protein features and $\mathbf{h}$ is the hidden layer that learns the inherent feature patterns. In Eq. (2), x_j denotes the j^{th} visible unit and h_k represents the k^{th} hidden unit . The weight connecting them is represented by W_{kj}. The hidden and visible biases are denoted b_k and c_j, respectively. Eq. (3) gives equivalent matrix form of Eq. (2)

$$\mathrm{E}(\mathbf{x}, \mathbf{h}) = -\sum_k \sum_j h_k x_j W_{kj} - \sum_j c_j x_j - \sum_k b_k h_k \tag{2}$$

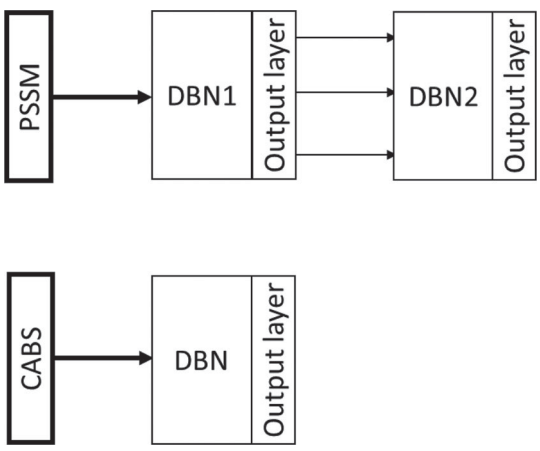
$$\mathrm{E}(\mathbf{x}, \mathbf{h}) = -\mathbf{h^T W x} - \mathbf{c^T x} - \mathbf{b^T h} \tag{3}$$

$$\mathrm{P}(\mathbf{x^t}, \mathbf{h}) = \frac{\exp(-E(\mathbf{x^t}, \mathbf{h}))}{Z} \tag{4}$$

Then, the joint probability for an input protein feature $\mathbf{x^t}$ and the network's hidden states is represented with the Boltzmann equation $\mathrm{P}(\mathbf{X} = \mathbf{x^t},\ \mathbf{h})$ as shown in Eq. (4), where Z denotes the partition function. Observe that, for the CABS-algorithm derived energy potentials, the minimization of the negative log-likelihood of Eq. (4) neatly captures the secondary structure prediction as an energy minimization problem,



(a) Individual DBN



(b) DBN Overview

Fig. 3. Architecture of Deep Belief Network (DBN). (a) The three-layer architecture of an individual DBN for prediction of secondary structures. It consists of an input layer with m neurons, a single RBM in the hidden layer and an output layer with n neurons. For CABS-algorithm derived features, $m = 27$ and for PSSM-profile based features, $m = 420$ while $n = 3$ for the three-state SS in the output layer. The hidden layer contains Q neurons, where w_n^Q represents the weight connecting the Q^{th} hidden neuron to the n^{th} output neuron. (b) Overview of DBN architectures using two protein feature representations. For the PSSM-profile based features a second level structure-to-structure DBN, with input features $m = 27$, is stacked onto the first level DBN.

that indirectly models the sequence to structure dependency. Fig. 3 shows the DBN architecture.

For CABS-algorithm derived features, a DBN consisting of a single RBM layer was trained to predict secondary structures. The input $\mathbf{x^t} \in \Re^{\mathbf{m}}$ was scaled into the range [0,1] using $\frac{\mathbf{x^t} - \mathbf{min(x^t)}}{max(\mathbf{x^t}) - \mathbf{min(x^t)}}$. The second layer contains K hidden neurons with a sigmoidal activation ($h^t = 1(1 + \exp(-\mathbf{x^t}))$ and the third and final layer consists of n output neurons, that use a linear activation ($\hat{y}_l^t = \sum_{k=1}^{K} w_{lk} h_k^t$), to obtain the predicted output. For the PSSM-profile based features, a second level DBN whose input is the final layer activation of the first level DBN, is used to predict secondary structures. The second level input was read in a window of 9 residues to give $m = 3 \times 9 = 27$ features.

For N protein residues, the negative log-likelihood error function to be minimized is $\frac{1}{N}\sum_{t=1}^{N} -\log \mathrm{P}(\mathbf{x^{(t)}})$, which is computationally intractable due to the partition function term. Hence, a stochastic gradient descent approach was used to train the DBN. The weight update rule is derived from taking the partial derivative of $\frac{1}{N}\sum_{t=1}^{N} -\log \mathrm{P}(\mathbf{x^{(t)}})$ with respect to the hidden weight, w_{kj}. The weight update rules and Gibbs sampling approach in the reconstruction step are given in Section 2.4.

2.4 Contrastive Divergence & Weight Update Rules

$$\frac{\partial}{\partial W_{kj}}(-\log p(\mathbf{x^{(t)}})) = \left\langle \frac{\partial \mathbf{E}(\mathbf{x^{(t)}}, \mathbf{h})}{\partial \mathbf{W_{kj}}} | \mathbf{x^{(t)}} \right\rangle_{\mathbf{h}} - \left\langle \frac{\partial \mathbf{E}(\mathbf{x}, \mathbf{h})}{\partial \mathbf{W_{kj}}} \right\rangle_{\mathbf{x,h}} \tag{5}$$

Eq. (5) shows the minimization of the partial derivative of the negative log likelihood with respect to the weights. Since the expectation term $-\langle \frac{\partial E(\mathbf{x},\mathbf{h})}{\partial W_{kj}} \rangle_{x,h}$ is intractable, it is replaced with the Gibbs sampled point estimate $\frac{\partial E(\tilde{\mathbf{x}},\tilde{\mathbf{h}})}{\partial W_{kj}}$ in the contrastive divergence (CD) algorithm. From the reconstructed input ($\tilde{\mathbf{x}}$) and Eq. (5), the weight update rules as shown in Eq. (6) are obtained. Here, α denotes the learning rate

$$\mathbf{W} = \mathbf{W} + \alpha(\mathbf{h}(\mathbf{x^{(t)}})\mathbf{x^{(t)^T}} - \mathbf{h}(\tilde{\mathbf{x}})\tilde{\mathbf{x}}^{\mathbf{T}}) \tag{6}$$

$$\mathbf{b} = \mathbf{b} + \alpha(\mathbf{h}(\mathbf{x^{(t)}}) - \mathbf{h}(\tilde{\mathbf{x}})) \tag{7}$$

$$\mathbf{c} = \mathbf{c} + \alpha(\mathbf{x^{(t)}} - \tilde{\mathbf{x}}) \tag{8}$$

Similarly, the update rules for the hidden ($\mathbf{b}$) and input ($\mathbf{c}$) biases are given in Eq. (7) and Eq. (8), respectively.

The Gibbs sampling steps to obtain $\tilde{\mathbf{x}}$ are as follows. The contrastive divergence algorithm is applied during the unsupervised pre-training to reconstruct the input protein features by minimizing the divergence between the observed data $\mathbf{x^{(t)}}$ and (Gibbs) sampled data ($\tilde{\mathbf{x}}$).

1) For each training sample $\mathbf{x^{(t)}}$
   a) Perform k steps of Gibbs sampling to obtain $\tilde{\mathbf{x}}$, with $\mathbf{x^{(t)}}$ as initial point
   b) Update parameters using Eq. (6),(7), (8)
2) Repeat for i epochs

The algorithm description and equations were based on Geoffrey Hinton's paper [54] and on Dr. Hugo Larochelle's video lectures [55], [56], [57]. The DBN code was written by Rasmus B. Palm [58] and adapted for SS prediction here.

In the pre-training step, the DBN learns in an unsupervised manner to reconstruct the given input protein features. During the reconstruction, it extracts the underlying higher order representation inherent in the input features. In case of the CABS-algorithm derived energy profiles, correlated energy scores between neighbouring residues and their respective secondary structures can be captured. In the case of the PSSM-profiles, correlated profile scores between residues and the adopted SS may be captured.

After the reconstruction error converges, the pre-trained weights are initialized onto a feed-forward network. In the supervised learning stage, the network learns from the known secondary structures to iteratively refine the weights for a preset number of epochs. To enable a faster convergence

TABLE 2
Comparison of CABS-Algorithm Derived and PSSM-Profile Based Protein Feature Represenations

Method	K	m	Train		Test	
			Q_3 (%)	Q_a (%)	Q_3 (%)	Q_a (%)
DBN-CABS	100	27	**82.70** (0.16)	**82.02** (0.33)	**82.60** (0.63)	**81.89** (0.40)
DBN1-PSSM	3000	420	79.49 (0.76)	78.09 (1.52)	72.87 (0.87)	71.20 (1.96)
DBN2-PSSM	1000	27	80.45 (0.40)	79.31(0.59)	73.97 (1.18)	72.58 (1.27)

DBN$i-$ indicates the network level and features, while m and K denote the number of neurons in the input and hidden layers, respectively. The average scores of 5-fold cross-validation are given, with standard deviation in brackets.

of the network, the network error ($y^t - \widehat{y}^t$) was calculated with the hinge-loss error [59].

2.4.1 Hinge-Loss Error

The hinge-loss error is defined as

$$E = \begin{cases} r * (y - \hat{y}) & y\hat{y} < 0 \\ y - \hat{y} & 0 \leq y\hat{y} < 1 \\ 0 & y\hat{y} \geq 1 \end{cases} \quad (9)$$

where r is the risk factor, set to 1.01.

2.5 Performance Measures

The measures to assess the DBN performance are the (i) single residue accuracy, $Q_3 = \frac{\sum_{\forall j} t_{jj}}{N} \times 100$ and (ii) class-wise accuracy $Q_j = \frac{t_{jj}}{N_j} \times 100$, where $j \in \{H, E, C\}$. The number of correctly predicted residues in class j is denoted t_{jj} and N_j represents the total number of protein residues in class j. The average class-wise accuracy for the three state SS is given by (iii) $Q_a = \frac{\sum_{\forall j} Q_j}{3}$.

3 Experiments

Experiments were conducted on (i) a Windows 7 PC with 3.6GHz clock speed and 8.0G RAM, running MATLAB 2012b and (ii) a Windows 10 laptop with 16.0G RAM, 1.8 GHz clock speed and MATLAB version 2019b. Some experiments were repeated and the results were similar across both machines, except for minor differences in accuracy ($<1\%$) caused by the random weight initialization during pre-training and the stochastic gradient descent function.

In the case of CABS-algorithm derived energy profiles, 5-fold cross-validation was performed twice on CB513; once without removing the two proteins that had fold similarity to GSW25 (results in Tables 2, 3 and 4) and once after removal (results in Table 5). In either case, the average Q_3 of five-fold cross-validation on CB513, before and after fold similarity removal to GSW25 was almost identical with no significant difference. The learning rate was set to 0.1 for both types of feature representations. More details on hyper-parameters and choice of model settings are available in Section 3.1.

Table 2 shows the results of 5-fold cross-validation experiments using DBN trained separately with PSSM-profile based and CABS-algorithm derived energy profile based representations. For the PSSM-profile based features, the first level and second level structure to structure network are shown as DBN1 and DBN2 respectively. Averages and standard deviations for train and test partitions are given in brackets. The results show that the CABS-algorithm derived protein feature representation (DBN-CABS) had a substantial improvement of 9.73% over the level 1 network using PSSM (DBN1-PSSM) and 8.63% over the second level network (DBN2-PSSM). Similarly, the average class wise accuracy Q_a is much higher for the DBN-CABS model compared to both levels of DBN-PSSM.

TABLE 3
Best Performing Test Partition on CB513, Total of 12,327 Residues

Method	Observed j	Predicted j			Q_j (%)	Q_3 (%)	Q_a (%)
		H	E	C			
DBN-CABS	H	4050	3	460	89.75	**83.2**	**81.59**
	E	16	1854	709	71.89		
	C	431	453	4351	83.12		
DBN1-PSSM	H	3684	227	602	81.64	73.69	73.13
	E	250	1758	571	68.17		
	C	878	716	3641	69.56		
DBN2-PSSM	H	3617	149	747	80.15	75.48	73.9
	E	159	1672	748	64.84		
	C	657	563	4015	76.7		

For PSSM-profile based protein feature representation, the second level network (DBN2-PSSM) was able to improve the performance by a small amount of 1.79%. However, despite trying an extensive number of hyper parameter settings and architecture variants, there was very little improvement in Q_3. The momentum, α and learning rate parameters were each searched coarsely in $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ but the network performance did not improve further. At best, these attempts either gave a slightly lower test Q_3 (by $<1\%$) compared to Table 2, or the network failed to converge. While only the best performance is discussed here, further descriptions of other experiments that have been performed can be found in Section 3.1.

Table 3 shows the confusion matrices and class-wise accuracies for the best performing test partitions corresponding to the DBN models given in Table 2. Consistent with other works [13], [30], Helix proved the easiest structure to predict, followed by Coil and Sheet.

In table 4, the reported $k-$fold cross-validated Q_3 scores for several works are presented. All method names in tables have been capitalized for consistency. The highest Q_3 score (84.3%) was obtained by SecNet, while DBN-CABS obtained

TABLE 4
Comparison of Reported Cross-Validation Scores on CB513

Method	Q_3 (%)	No. of Training Proteins	Source
SECNET	84.3	8563	[27]
DBN-CABS	82.6	385	This Work
ECRRNN	81.2[1]	11948	[27]
RAPTORX	82.3	5600	[25]
FCRN	82.14	385	[60]
FLOPRED	81.3	387	[38]
PSIPRED	79.2	Not available[2]	[25]
JNET	76.4	480	[61]

[1] *Initially reported as 87.3% [26], owing to a lenient SS assignment scheme that reduces the isolated β-bridge residues (B) to Coil instead of Sheet. Following the same stringent 8 to 3 state SS reduction rules used here results in a lower score [27].*

[2] *Not stated in publications or on the web-server.*

TABLE 5
Methods Comparison on G Switch Proteins (GSW25)

Method	Observed j	Predicted j			Q_j (%)	Q_3 (%)	Q_a (%)
		H	E	C			
	H	683	0	39	94.6		
DBN-CABS	E	23	376	147	68.87	**83.22**	**81.26**
	C	24	2	106	80.31		
	H	680	0	42	94.19		
SSP_{55}-FCRN*	E	25	384	137	70.33	80.36	70.25
	C	51	20	61	46.22		
	H	665	19	38	92.11		
FLOPRED*	E	41	380	125	69.6	78.72	68.3
	C	49	26	57	43.19		
	H	688	0	34	95.3		
SSP_{55}-DBN-CABS	E	63	241	242	44.14	69.79	58.61
	C	72	12	48	36.37		
	H	556	50	116	77.01		
PROTEUS2*	E	17	302	227	55.32	61.72	45.63
	C	2	124	6	4.55		
	H	609	6	107	84.35		
PORTER5.0	E	100	153	293	28.03	59.72	56.15
	C	17	41	74	56.07		
	H	479	119	124	66.35		
SECNET	E	53	283	210	51.84	58.79	54.81
	C	0	71	61	46.22		
	H	519	99	104	71.89		
PSIPRED*	E	167	243	136	44.51	57.36	49.16
	C	5	86	41	31.07		
	H	490	67	165	67.87		
DNSS2	E	106	165	275	30.22	51.43	49.12
	C	9	58	65	49.25		
	H	368	162	192	50.97		
SSPRO*	E	13	312	221	57.15	50.43	42.61
	C	1	105	26	19.7		
	H	616	0	106	84.35		
RAPTORX	E	266	22	258	28.03	49.08	56.15
	C	83	0	49	56.07		
	H	573	16	133	79.37		
SPIDER3	E	172	45	329	8.25	47.22	40.07
	C	87	2	43	32.58		
	H	289	59	374	40.03		
DBN1-PSSM	E	12	128	406	23.45	34.93	39.35
	C	0	60	72	54.55		

All method names are capitalized for consistency.
**indicates that scores were reported in [34].*

a slightly lower performance of about about 1.7%. Besides DBN-CABS, FCRN and FLOPRED, all other methods in Table 4 had been developed on sequence based feature representation such as PSSM or HMM based profiles. A Q_3 score of 87.3% was initially reported by eCRRNN [26]. However, a direct comparison of this high score may not be fair due to the use of a more lenient 8 to 3 state SS reduction rule [27]. In the lenient scheme, isolated β-bridge (B) residues are assigned to Coil instead of Sheet. Upon following the same SS assignment rule as adopted in this paper, the Q_3 score of eCRRNN was calculated to 81.2% [27].

Table 5 compares the DBN performance with several deep learning methods on the independent GSW25 dataset. The highest performance was obtained by the proposed model (DBN-CABS) at 83.22%, which is higher by 2.86% over the previously developed compact model (known as SSP_{55}-FCRN). Most of the other prediction models' Q_3 were below 70% which indicates the challenge of the GSW25 test set. Despite all of them being trained on much larger datasets, the difficulty remained. In particular as shown in the confusion matrices of Table 5, a substantial number of Sheet residues were wrongly predicted as Coils, resulting in Q_E as low 8.25% (SPIDER3). The worst Coil accuracy, was Q_C of 4.55% (PROTEUS2). The best Helix and Coil class-wise accuracies were shown by DBN-CABS ($Q_H = 94.6\%$, $Q_C = 80.31\%$) and the best Sheet accuracy, $Q_E = 70.33\%$ was given by SSP_{55}-FCRN. A comparison of the average class-wise accuracy (Q_a) indicates that DBN-CABS shows a substantial improvement

of 11.01% over the second-best compact model (SSP_{55}-FCRN) and a vast improvement over the other methods. Finally the DBN1-PSSM model failed spectacularly at $Q_3 = 39.35\%$, demonstrating the inability to learn sufficiently from the small training dataset using PSSM-profiles.

3.1 Hyperparameter Settings and other Experiments

In the unsupervised pre-training stage, the learning rate and momentum parameters were set to 0.1 and 0.05 respectively. In the supervised training stage, the learning rate was 5×10^{-5} while momentum was 10^{-6}. The number of epochs was 50 during the pre-training stage and 5000 for the supervised learning stage. The batchsize for both stages was kept the same at 2113.

To obtain a higher DBN performance especially for the PSSM-profile based features, up to 5 hidden layers were added and the batch sizes were varied from 1 to 5000. The Rectified Linear Unit (ReLU) and Tangent activation functions were also trialed in the hidden layers, followed by the Softmax function in the output layer. The Meiler properties [31] (steric graph shape index, polarizability, volume, hydrophobicity, iso-electric point), and one-hot encoding of residues were incorporated with PSSM-profile based feature representation. To increase the number of samples, Synthetic Minority Oversampling (SMOTE) was also attempted for the PSSM-profile based features. None of these attempts resulted in Q_3 higher than 75% when training the DBN with PSSM-profile based features from the CB513 dataset.

3.1.1 Sliding Window Size

There has been no consensus on the optimal sliding window size (w) to model the long range interactions between residues. Various window sizes have been used in the following works: in PSIPRED [12], $w = 15$, in SPIDER2 [21], $w = 17$, in DNSS [24], $w = 19$ and in SECNET [27], $w = 29$. Nevertheless, it has generally been agreed that for odd sizes of w in $13 < w < 27$, the effect on Q_3 is not too severe ($\leq 1.1\%$) [13], [21], [24]. Here, the sliding window was set to 21 like in Spine [13] and RaptorX [25]. The resultant PSSM contained $F = 20 * 21$ feature columns. A protein chain N residues long was encoded as an NxF matrix for DBN training.

3.1.2 Blast Settings

Due to the constrained training dataset size, the choice of query database and PSI-BLAST settings were made with stringent criteria. The *RefSeq* database was selected to only include cleaned, well-curated and representative proteins [62] as well as a low e-value threshold of 10^{-5} (compared to 0.01 or 0.001 chosen in most works). The *RefSeq* database was carefully filtered to avoid low complexity and transmembrane regions which could produce stray hits to CB513, which consists of mostly globular proteins. The strict criteria with a smaller database and high cut-off values used, had a resultant trade-off that 15 sequences failed to produce hits. For these, the e-value threshold was relaxed to 100 and the PAM70 matrix was used for proteins shorter than 100 residues to generate the PSSM-profiles.

The PSSMs generated by the psiblast program from the BLAST+ software suite contain internally scaled (transformed) scores, which are typically integers in a narrow range such as $[-10, 10]$. Due to the loss of precision from the transformation, another set of PSSM was generated by the blastpgp program from blast-2.2.19 (legacy blast). The makemat program by IMPALA [63] was then used to recover the unscaled PSSM. The resultant PSSM profile scores fell in a wider range of $[-1000, 1000]$, but they still failed to improve Q_3 scores beyond 73%.

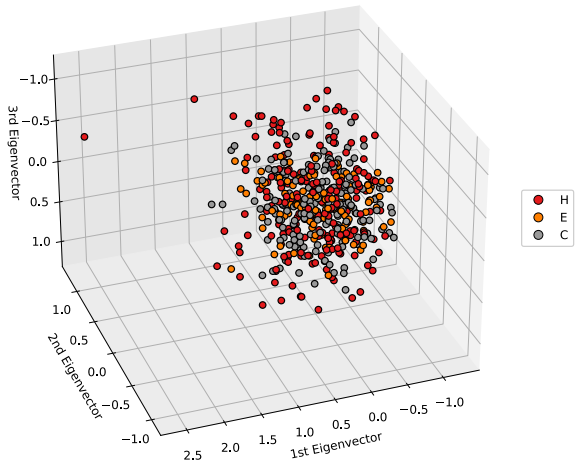
4 Discussion

Role of Protein Feature Representation in Q_3 Accuracy

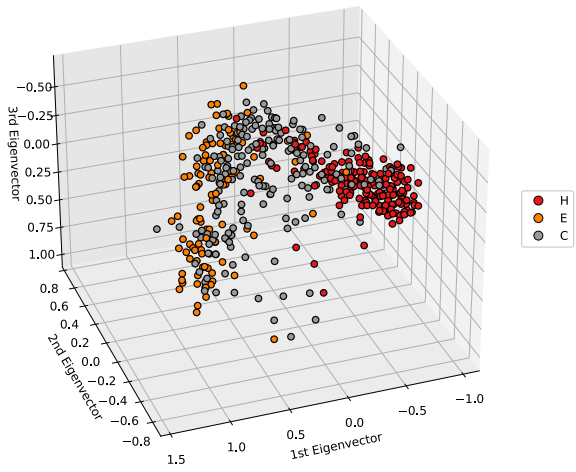
All results indicate that the choice of protein feature representation is extremely important when training a deep learning model from scratch on a small dataset. For models using sequence based input such as PSSM or HMM profiles, large numbers of training proteins are needed to ensure $Q_3 > 80\%$. For example in Table 4, the top performing program SECNET had been trained on more than 8000 proteins and had 1.7% higher accuracy than DBN-CABS. However, for a training set that was more than 22 times the size of the small dataset, the gain in accuracy is less than 2%. As the results of Table 5 indicate, the use of large training datasets results in a trade-off in the sensitivity to new datasets. Most SS prediction programs were developed on thousands of training proteins, which may have affected the generalization ability to new sequences with distinct site specific structural shifts, as is the case for GSW25.

The results of the DBN-CABS model are likely due to the joint energy configuration of the DBN successfully capturing the CABS-algorithm derived energy profiles. The joint energy function in Eq. (3) relates the model parameters (hidden nodes ($\mathbf{h}$) and weights ($\mathbf{W}$)) with the CABS-algorithm derived features represented by $\mathbf{x}$. The CABS-algorithm derived features employ knowledge-based statistical potentials in the force-field calculation, which consist of unique context-dependent potentials. The potentials depend on the local geometry of the protein main chain, which determines the secondary structure. The CABS algorithm derived features also account for the conformational preferences of protein sequence fragments since they contain proper averages of structural regularities of thousands of known proteins. During training, the optimum model weights and hidden state for a given input residue's features that will minimize the joint energy configuration given by E($\mathbf{x}$, $\mathbf{h}$) are learnt. Then, minimizing the gradient of the negative log-likelihood of Eq. (4) to achieve network convergence is likened to finding the direction of the energy landscape that contains the lowest-energy secondary structures. This allows DBN to extract the energy based structural relationships present in the features to effectively predict secondary structures, as shown for the G Switch Proteins dataset.

One major reason for the performance of DBN-CABS compared to sequence profile based features, could be the loss of precision by PSSM-profile based representation. Due to the use of repeated integer values being mapped to the same structure space in the PSSM-profiles, the context-dependent structure signals could be overlapping across the three SS classes. This is demonstrated with the principal components analysis shown in Fig. 4. Both the PSSM-profiles based and CABS-algorithm energy profiles based features were scaled to lie in [0,1]. Next, 500 residues were



(a) First Three PCA Directions, PSSM



(b) First Three PCA Directions, CABS

Fig. 4. Principal Components Analysis of two types of feature representation on the CB513 dataset. The PSSM-profile based representation is shown in (a) and the CABS-algorithm derived energy profile based representation in (b). Randomly chosen 500 residues were selected for visualization. The same 500 residues are depicted in both (a) and (b). One circle denotes a single protein residue. Helix, Sheet and Coil structures are shown as red (H), orange (E) and grey (C) circles, respectively. Clearly, compared to one dense cluster in (a), three visible clusters coloured according to their respective SS structures are seen in (b), indicating the ease of class-wise seperation for the CABS-algorithm derived energy profile based features.

selected at random for the visualization. The first three components (i.e., top three axes of highest variance) were visualized with Python's matplotlib library (version 3.3.3).

Fig. 4b indicates that using CABS-algorithm based features results in three visibly distinct clusters, compared to the PSSM profile based features which form a single cluster Fig. 4a, making the former more effective for the DBN model. Consistent with the findings in Tables 2 and 3, learning discriminable representation from PSSM-profile based features is difficult and many thousands of training proteins may be needed for a good Q_3 score. In Fig. 4b, the location of the three clusters with respect to each other, reflects the difficulty of prediction of the classes observed here and in other works; namely Helix (H) and Sheet (E) classes are well separated in the feature space, with Coils (grey circles) being the overlap class. Wrongly classified Helix or Sheet residues are overwhelmingly likely to be predicted as Coil rather than Sheet or Helix. As shown in Table 3 for the DBN2-PSSM model, more than 83% of wrongly classified Helices and Sheets belong to the Coil class. It is unclear why the PSSM-profile based features were rather unsuccessful in replicating the success of the small training dataset with DBN models.

As other works employing large training datasets (ranging in the thousands) have demonstrated high accuracies ($Q_3 \geq 80\%$) in using PSSM-profile based features with deep learning networks [22], [24], [27], the main reason could be that much more data samples are needed to learn an adequate feature representation for the reconstruction step. In the case of CB513, all PSSM values were discrete numbers in the range $[-17, 14]$ and could not be discriminated easily by the DBN with the limits of the small dataset.

In contrast, the findings of Table 2 show that using the CABS-algorithm derived feature representation results in an excellent model using a single layer RBM. It is capable of detecting large structural changes for small changes in sequence. Hence, the residues' structural correlations are captured effectively by the joint energy configuration (Eq. (3)) in the DBN-CABS model. DBN-CABS illustrates the potential use in protein design or engineering applications where there are few pre-existing homologous sequences or large changes in folds for small site specific substitutions. Other work such as done by Tubiana et.al., has also presented evidence of the effectiveness of RBM in capturing features related to protein secondary structure motifs and in the generation of new lattice protein sequences [64].

Independence of Train and Test Datasets

The performance of DBN-CABS in Table 5 cannot be attributed to homology or direct assignment of SS between train and test datasets, for two reasons. First as described in Section 2, strict protocol had been implemented to establish train and test independence. Second, other SS prediction programs had been trained with highly similar proteins to GSW25. For instance, Proteus2 [65] contains PDB ID 2IGD in its train dataset, which is about 60% similar to the G_A and G_B domains. The Porter 4.0 train dataset [66] contain PDB IDs 2J5Y and 3FIL, which again are at least 58% similar to the G_A and G_B domains. Yet, the results show that the presence of similar training proteins need not indicate that the model can be accurate for unseen datasets and that overtraining can occur on large datasets.

It is emphasized that the DBN model demonstrated here has been very carefully developed after sequence similarity and fold similarity have been removed between the CB513 and GSW25 datasets, and that GSW25 was strictly used in blind tests. Hence, it represents an ab-initio form of secondary structure prediction without direct assignment of structures from templates.

For PSSM-profile based features, the stacking of more RBM layers did not improve performance. In many instances, the Q_3 deteriorated upon adding more layers, most likely due to being constrained on a small training set ($<$ 400 proteins). Hence, the architecture was kept simple, which has the benefit of reduced computational complexity and rapid prediction of structures.

One drawback of the proposed model is that it is computationally intensive to generate CABS-algorithm derived

energy profiles as compared to standard PSSM-profile based features. For a protein with 100 residues, it took approximately 26 hours to generate the potentials on a Linux machine with 8G RAM and 2.3GHz of processor speed. For a protein with 100 residues, the time to generate PSSM-profiles with BLAST+ was approximately 2 hours (speed improves with availability of identical hits) as tested on a Linux machine with 8G RAM and 3.6GHz of processor speed. Another drawback of the proposed approach is that the CABS-algorithm is not publicly available for energy profile generation on larger datasets.

Here the experiments were reported based on pre-generated potentials for CB513 and GSW25. Therefore, it is difficult to compare with more benchmark datasets. However, for reproducibility and comparison of the results reported, the CB513 and GSW25 energy potentials are available upon request. A next step would be the use of predicted structural fragments from public ab-initio models (such as QUARK [67]) to build a set of quasi-energy based profiles.

5 Conclusion

The DBN model, trained with CABS-algorithm derived protein feature representation (DBN-CABS) demonstrated here, showed improvement in secondary structure prediction and is capable of detecting large fold changes for small position changes in the sequence. DBN-CABS obtained higher Q_3 (of more than 8%) and also higher class-wise accuracies compared with PSSM-profile based features (DBN2-PSSM) on the CB513 dataset. In a blind test dataset consisting of G Switch Proteins, the DBN-CABS model showed improvement in accuracy of almost 3%, over a previously developed compact model. Despite being trained on a small dataset containing less than 400 proteins, the findings indicate that Q_3 accuracy was substantially improved compared to previously developed SS predictors that were trained on thousands of proteins.

This work investigates the importance of protein feature representation and clearly indicates that for deep learning to be successful on a small dataset, informative protein features that incorporate energetics are vital to detect sensitive fold changes. Potential applications include cases where the requisite designed proteins may not have enough homologues (e.g., neutralizing antibodies for a novel virus or specific drug targets). It presents a methodololgy for scientists to build deep predictive networks by training on compact datasets based on energy profiles. Finally, it signifies the need for continued research into small data approaches in future work.

References

[1] E. D. Scheeff and P. E. Bourne, "Structural evolution of the protein kinase–like superfamily," *PLoS Comput. Biol.*, vol. 1, no. 5, Oct. 2005. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1261164/

[2] K. L. Tkaczuk, S. Dunin-Horkawicz, E. Purta, and J. M. Bujnicki, "Structural and evolutionary bioinformatics of the SPOUT superfamily of methyltransferases," *BMC Bioinf.*, vol. 8, no. 1, Mar. 2007, Art. no. 73.

[3] K. Chen, M. J. Mizianty, and L. Kurgan, "Prediction and analysis of nucleotide-binding residues using sequence and sequence-derived structural descriptors," *Bioinformatics*, vol. 28, no. 3, pp. 331–341, Feb. 2012. [Online]. Available: https://academic.oup.com/bioinformatics/article/28/3/331/188242

[4] Y. Cui, Q. Dong, D. Hong, and X. Wang, "Predicting protein-ligand binding residues with deep convolutional neural networks," *BMC Bioinf.*, vol. 20, no. 1, Feb. 2019, Art. no. 93.

[5] B. Adhikari and J. Cheng, "Protein residue contacts and prediction methods," *Methods Mol. Biol.*, vol. 1415, pp. 463–476, 2016.

[6] J. Garnier, D. J. Osguthorpe, and B. Robson, "Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins," *J. Mol. Biol.*, vol. 120, no. 1, pp. 97–120, Mar. 1978.

[7] A. Kloczkowski, K.-L. Ting, R. L. Jernigan, and J. Garnier, "Combining the GOR v algorithm with evolutionary information for protein secondary structure prediction from amino acid sequence," *Proteins*, vol. 49, no. 2, pp. 154–166, Nov. 2002.

[8] T. Z. Sen, R. L. Jernigan, J. Garnier, and A. Kloczkowski, "GOR V server for protein secondary structure prediction," *Bioinformatics*, vol. 21, no. 11, pp. 2787–2788, Jun. 2005.

[9] J. Garnier, J. F. Gibrat, and B. Robson, "GOR method for predicting protein secondary structure from amino acid sequence," *Methods Enzymol.*, vol. 266, pp. 540–553, 1996.

[10] B. Rost, "PHD: Predicting one-dimensional protein structure by profile-based neural networks," *Methods Enzymol.*, vol. 266, pp. 525–539, 1996.

[11] S. F. Altschul *et al.*, "Gapped BLAST and PSI-BLAST: A new generation of protein database search programs," *Nucleic Acids Res.*, vol. 25, no. 17, pp. 3389–3402, Sep. 1997.

[12] D. T. Jones, "Protein secondary structure prediction based on position-specific scoring matrices," *J. Mol. Biol.*, vol. 292, no. 2, pp. 195–202, Sep. 1999.

[13] O. Dor and Y. Zhou, "Achieving 80% ten-fold cross-validated accuracy for secondary structure prediction by large-scale training," *Proteins*, vol. 66, no. 4, pp. 838–845, Mar. 2007.

[14] G. Pollastri and A. McLysaght, "Porter: A new, accurate server for protein secondary structure prediction," *Bioinformatics*, vol. 21, no. 8, pp. 1719–1720, Apr. 2005.

[15] S. Montgomerie, S. Sundararaj, W. J. Gallin, and D. S. Wishart, "Improving the accuracy of protein secondary structure prediction using structural alignment," *BMC Bioinf.*, vol. 7, 2006, Art. no. 301.

[16] G. Pollastri, A. J. Martin, C. Mooney, and A. Vullo, "Accurate prediction of protein secondary structure and solvent accessibility by consensus combiners of sequence and structure information," *BMC Bioinf.*, vol. 8, no. 1, Jun. 2007, Art. no. 201.

[17] J. Cheng, A. Z. Randall, M. J. Sweredoski, and P. Baldi, "SCRATCH: a protein structure and structural feature prediction server," *Nucleic Acids Res.*, vol. 33, no. suppl 2, pp. W72–W76, Jul. 2005.

[18] B. Rost, C. Sander, and R. Schneider, "Redefining the goals of protein secondary structure prediction," *J. Mol. Biol.*, vol. 235, no. 1, pp. 13–26, Jan. 1994.

[19] D. Kihara, "The effect of long-range interactions on the secondary structure formation of proteins," *Protein Sci.*, vol. 14, no. 8, pp. 1955–1963, Aug. 2005.

[20] Y. Yang *et al.*, "Sixty-five years of the long march in protein secondary structure prediction: The final stretch?," *Brief. Bioinf.*, vol. 19, no. 3, pp. 482–494, 2018.

[21] R. Heffernan *et al.*, "Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning," *Sci. Rep.*, vol. 5, 2015, Art. no. 11476.

[22] R. Heffernan, Y. Yang, K. Paliwal, and Y. Zhou, "Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility," *Bioinformatics*, vol. 33, no. 18, pp. 2842–2849, Sep. 2017.

[23] M. Torrisi, M. Kaleel, and G. Pollastri, "Deeper Profiles and Cascaded Recurrent and Convolutional Neural Networks for state-of-the-art Protein Secondary Structure Prediction," *Sci. Rep.*, vol. 9, no. 1, pp. 1–12, Aug. 2019. [Online]. Available: https://www.nature.com/articles/s41598–019-48786-x

[24] M. Spencer, J. Eickholt, and J. Cheng, "A deep learning network approach to ab initio protein secondary structure prediction," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 12, no. 1, pp. 103–112, Jan./Feb. 2015.

[25] S. Wang, J. Peng, J. Ma, and J. Xu, "Protein secondary structure prediction using deep convolutional neural fields," *Sci. Rep.*, vol. 6, no. 1, Jan. 2016, Art. no. 18962. [Online]. Available: https://www.nature.com/articles/srep18962

[26] B. Zhang, J. Li, and Q. Lü, "Prediction of 8-state protein secondary structures by a novel deep learning architecture," *BMC Bioinf.*, vol. 19, no. 1, Aug. 2018, Art. no. 293.

[27] M. Shapovalov, R. L. D. Jr, and S. Vucetic, "Multifaceted analysis of training and testing convolutional neural networks for protein secondary structure prediction," *PLoS One*, vol. 15, no. 5, May 2020, Art. no. e0232528.
[28] Z. Guo, J. Hou, and J. Cheng, "DNSS2: Improved ab initio protein secondary structure prediction using advanced deep learning architectures," *Proteins*, vol. 89, no. 2, pp. 207–217, Feb. 2021.
[29] M. R. Uddin, S. Mahbub, M. S. Rahman, and M. S. Bayzid, "SAINT: Self-attention augmented inception-inside-inception network improves protein secondary structure prediction," *Bioinformatics*, vol. 36, pp. 4599–4608, Nov. 2020.
[30] H. Zhang *et al.*, "Critical assessment of high-throughput standalone methods for secondary structure prediction," *Brief. Bioinf.*, vol. 12, no. 6, pp. 672–688, Nov. 2011.
[31] J. Meiler, M. Müller, A. Zeidler, and F. Schmäschke, "Generation and evaluation of dimension-reduced amino acid parameter representations by artificial neural networks," *Mol. Model. Annu.*, vol. 7, no. 9, pp. 360–369, Sep. 2001.
[32] W. R. Atchley, J. Zhao, A. D. Fernandes, and T. Drüke, "Solving the protein sequence metric problem," *Proc. Nat. Acad. Sci. USA*, vol. 102, no. 18, pp. 6395–6400, May 2005. Available: https://www.pnas.org/content/102/18/6395
[33] M. Remmert, A. Biegert, A. Hauser, and J. Söding, "HHblits: Lightning-fast iterative protein sequence searching by HMM-HMM alignment," *Nature Methods*, vol. 9, no. 2, pp. 173–175, Feb. 2012. [Online]. Available: https://www.nature.com/articles/nmeth.1818
[34] S. Rashid, S. Saraswathi, A. Kloczkowski, S. Sundaram, and A. Kolinski, "Protein secondary structure prediction using a small training set (compact model) combined with a complex-valued neural network approach," *BMC Bioinf.*, vol. 17, no. 1, 2016, Art. no. 362.
[35] B. Barz and J. Denzler, "Deep learning on small datasets without pre-training using cosine loss," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2020, pp. 1360–1369.
[36] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *Proc. Int. Conf. Learning Representations*, Apr. 2017.
[37] J. A. Cuff and G. J. Barton, "Evaluation and improvement of multiple sequence methods for protein secondary structure prediction," *Proteins*, vol. 34, no. 4, pp. 508–519, Mar. 1999.
[38] S. Saraswathi, J. L. Fernández-Martínez, A. Kolinski, R. L. Jernigan, and A. Kloczkowski, "Fast learning optimized prediction methodology (FLOPRED) for protein secondary structure prediction," *J. Mol. Model.*, vol. 18, no. 9, pp. 4275–4289, 2012.
[39] P. A. Alexander, Y. He, Y. Chen, J. Orban, and P. N. Bryan, "A minimal sequence code for switching protein structure and function," *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 50, pp. 21149–21154, Dec. 2009.
[40] P. N. Bryan and J. Orban, "Proteins that switch folds," *Curr. Opin. Struct. Biol.*, vol. 20, no. 4, pp. 482–488, Aug. 2010.
[41] G. Wang and R. L. Dunbrack, "PISCES: A protein sequence culling server," *Bioinformatics*, vol. 19, no. 12, pp. 1589–1591, Aug. 2003.
[42] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, "SCOP: A structural classification of proteins database for the investigation of sequences and structures," *J. Mol. Biol.*, vol. 247, no. 4, pp. 536–540, Apr. 1995.
[43] N. K. Fox, S. E. Brenner, and J.-M. Chandonia, "SCOPe: Structural classification of proteins–extended, integrating SCOP and ASTRAL data and classification of new structures," *Nucleic Acids Res.*, vol. 42, no. Database issue, pp. D304–D309, Jan. 2014.
[44] W. Kabsch and C. Sander, "Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features," *Biopolymers*, vol. 22, no. 12, pp. 2577–2637, 1983.
[45] K. D. Pruitt, T. Tatusova, G. R. Brown, and D. R. Maglott, "NCBI reference sequences (RefSeq): Current status, new features and genome annotation policy," *Nucleic Acids Res.*, vol. 40, no. D1, pp. D130–D135, Nov. 2011.
[46] K. Bryson, L. J. McGuffin, R. L. Marsden, J. J. Ward, J. S. Sodhi, and D. T. Jones, "Protein structure prediction servers at university college london," *Nucleic Acids Res.*, vol. 33, no. Web Server issue, pp. W36–W38, Jul. 2005.
[47] C. Camacho *et al.*, "BLAST+: Architecture and applications," *BMC Bioinf.*, vol. 10, 2009, Art. no. 421.
[48] A. Kolinski, "Protein modeling and structure prediction with a reduced representation," *Acta Biochimica Polonica*, vol. 51, no. 2, pp. 349–371, 2004.
[49] C. A. Orengo, A. D. Michie, S. Jones, D. T. Jones, M. B. Swindells, and J. M. Thornton, "CATH–A hierarchic classification of protein domain structures," *Structure*, vol. 5, no. 8, pp. 1093–1108, Aug. 1997.
[50] P. J. Silva, "Assessing the reliability of sequence similarities detected through hydrophobic cluster analysis," *Proteins*, vol. 70, no. 4, pp. 1588–1594, Mar. 2008.
[51] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *J. Mol. Biol.*, vol. 48, no. 3, pp. 443–453, Mar. 1970.
[52] C. Sander and R. Schneider, "Database of homology-derived protein structures and the structural meaning of sequence alignment," *Proteins*, vol. 9, no. 1, pp. 56–68, 1991.
[53] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
[54] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, Aug. 2002.
[55] H. Larochelle, Neural networks [5.4]: Restricted boltzmann machine - Contrastive divergence, video lectures. Accessed: Apr 25, 2017, Nov. 2013. [Online]. Available: https://www.youtube.com/watch?v=MD8qXWucJBY
[56] H. Larochelle, "Neural networks [5.5]: Restricted boltzmann machine - contrastive divergence (parameter update)," video lectures. Accessed: Apr 25, 2017, Nov. 2013. [Online]. Available: https://www.youtube.com/watch?v=wMb7cads0go
[57] H. Larochelle, "Neural networks [5.6]: Restricted boltzmann machine - Persistent contrastive divergence," video lectures. Accessed: Apr 25, 2017, Nov. 2013. [Online]. Available: https://www.youtube.com/watch?v=S0kFFiHzR8M
[58] R. B. Palm, "Prediction as a candidate for learning deep hierarchical models of data," Master's thesis, DTU Informatics, Tech. Univ. Denmark, Kgs. Lyngby, Denmark, 2012.
[59] S. Suresh, N. Sundararajan, and P. Saratchandran, "Risk-sensitive loss functions for sparse multi-category classification problems," *Informat. Sci.*, vol. 178, no. 12, pp. 2621–2638, Jun. 2008.
[60] B. Shamima, R. Savitha, S. Suresh, and S. Saraswathi, "Protein secondary structure prediction using a fully complex-valued relaxation network," in *Proc. Int. Joint Conf. Netw.*, 2013, pp. 1–8.
[61] J. A. Cuff and G. J. Barton, "Application of multiple sequence alignment profiles to improve protein secondary structure prediction," *Proteins*, vol. 40, no. 3, pp. 502–511, Aug. 2000.
[62] K. D. Pruitt, T. Tatusova, and D. R. Maglott, "NCBI reference sequences (RefSeq): A curated non-redundant sequence database of genomes, transcripts and proteins," *Nucleic Acids Res.*, vol. 35, no. Database issue, pp. D61–D65, Jan. 2007. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1716718/
[63] A. A. Schäffer, Y. I. Wolf, C. P. Ponting, E. V. Koonin, L. Aravind, and S. F. Altschul, "IMPALA: Matching a protein sequence against a collection of PSI-BLAST-constructed position-specific score matrices," *Bioinf.*, vol. 15, no. 12, pp. 1000–1011, Dec. 1999.
[64] J. Tubiana, S. Cocco, and R. Monasson, "Learning protein constitutive motifs from sequence data," *eLife*, vol. 8, Mar. 2019, Art. no. e39397.
[65] S. Montgomerie, J. A. Cruz, S. Shrivastava, D. Arndt, M. Berjanskii, and D. S. Wishart, "PROTEUS2: A web server for comprehensive protein structure prediction and structure-based annotation," *Nucleic Acids Res.*, vol. 36, no. Web Server issue, pp. W202–209, Jul. 2008.
[66] C. Mirabello and G. Pollastri, "Porter, PaleAle 4.0: High-accuracy prediction of protein secondary structure and relative solvent accessibility," *Bioinformatics*, vol. 29, no. 16, pp. 2056–2058, Aug. 2013.
[67] D. Xu and Y. Zhang, "Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field," *Proteins*, vol. 80, no. 7, pp. 1715–1735, Jul. 2012. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3370074/

Shamima Rashid received the BSc degree in computational science (Life science specialization) from the National University of Singapore, in 2007 and the PhD degree from Nanyang Technological University (NTU), in Singapore, 2018. She is currently working with the School of Computer Science and Engineering in NTU as a research fellow. Her research interests include protein structural bioinformatics, the relationship of sequence to structure and the application of neural networks to the above areas.

Suresh Sundaram received the BE degree in electrical and electronics engineering from Bharathiyar University, Coimbatore, India, in 1999, and the ME and PhD degrees in aerospace engineering from the Indian Institute of Science, Bengaluru, India, in 2001 and 2005, respectively. He was a post-doctoral researcher with the School of Electrical Engineering, Nanyang Technological University, Singapore from 2005 to 2007. From 2010 to late 2018, he was an associate professor with the School of Computer Science and Engineering, Nanyang Technological University (NTU). At present he is an associate professor with the Department of Aerospace, Indian Institute of Science, Bangalore, India. His research interests include flight control, unmanned aerial vehicle design, machine learning and optimization as well as computer vision.

Chee Keong Kwoh received the bachelor of engineering and MSc degrees from the National University of Singapore, in 1987 and 1991 respectively, and the PhD degree from the Univeristy of London, Imperial College, in 1995. He is currently an Associate professor with the Department of Computer Science and Engineering, Nanyang Technological University. His research interests focus on the inference of new information not explicitly present in a single data source, which includes data analytics and mining, deep learning, soft computing and graph-based inference; application areas include structural bioinformatics, health informatics, and biomedical engineering.