IMPROVING PROTEIN STRUCTURE PREDICTION BY DEEP LEARNING AND COMPUTATIONAL OPTIMIZATION

A Dissertation

presented to

the Faculty of the Graduate School

at the University of Missouri-Columbia

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

by

JIE HOU

Professor Jianlin Cheng, Dissertation Supervisor

JULY 2019

The undersigned, appointed by the Dean of the Graduate School, have examined the dissertation entitled:

IMPROVING PROTEIN STRUCTURE PREDICTION BY DEEP LEARNING AND COMPUTATIONAL OPTIMIZATION

presented by Jie Hou,

a candidate for the degree of Doctor of Philosophy and hereby certify that, in their opinion, it is worthy of acceptance.

Professor Jianlin Cheng

Professor Jeffrey Uhlmann

Professor Dong Xu

Professor Yunxin Zhao

Professor James A. Birchler

DEDICATION

This dissertation is dedicated to my parents. I want to thank them for their constant support, encouragement, and love throughout my life.

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to my advisor and committee chair Dr.Jianlin Cheng. Dr.Cheng constantly offered me great support and excellent suggestions during the period of my Ph.D. study. He made me passionate about my research work and encouraged me to pursue my academic goal. Without his mentoring, I could not have completed this dissertation. Dr.Cheng's enthusiasm towards research and teaching lays the foundation for my future career.

I would like to thank my committee members for all of their guidance and suggestions through this long-term process, including: Dr. Dong Xu, Dr. Jeffrey Uhlmann, Dr. Yunxin Zhao and Dr. James A. Birchler.

I greatly appreciate the great support from the current and previous members in the Bioinformatics and Data Mining (BDM) lab. Their hard work and scientific contribution provided the foundation for the development of my methods. Furthermore, they created an excellent environment for research in our lab. I appreciate Dr.Jilong Li, Dr.Renzhi Cao, Dr.Badri Adhikari, and Dr.Debswapna Bhattacharya for their direct guidance in my research work. I also enjoyed working with group members including Dr.Oluwatosin Oluwadare, Meshal Alfarhood, Tianqi Wu, Chen (Chris) Chen, Anes Quadou, Adil Al-Azzawi, Zhiye Guo, and Farhan Quadir.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

ABSTRACT

Protein structure prediction is one of the most important scientific problems in the field of bioinformatics and computational biology. The availability of protein three-dimensional (3D) structure is crucial for studying biological and cellular functions of proteins. The importance of four major sub-problems in protein structure prediction have been clearly recognized. Those include, first, protein secondary structure prediction, second, protein fold recognition, third, protein quality assessment, and fourth, multi-domain assembly. In recent years, deep learning techniques have proved to be a highly effective machine learning method, which has brought revolutionary advances in computer vision, speech recognition and bioinformatics.

In this dissertation, five contributions are described. First, DNSS2, a method for protein secondary structure prediction using one-dimensional deep convolution network. Second, DeepSF, a method of applying deep convolutional network to classify protein sequence into one of thousands known folds. Third, CNNQA & DeepRank, two deep neural network approaches to systematically evaluate the quality of predicted protein structures and select the most accurate model as the final protein structure prediction. Fourth, MULTICOM, a protein structure prediction system empowered by deep learning and protein contact prediction. Finally, SAXSDOM, a data-assisted method for protein domain assembly using small-angle X-ray scattering data. All the methods are available as software tools or web servers which are freely available to the scientific community.

Chapter 1

Introduction

Three-dimensional (3D) structure information of proteins is vital for studying their function involved in the cellular processes. The uniquely folded three-dimensional (3D) conformation (tertiary structure) of a protein is primarily determined by its amino acid sequence. Over the past decade, the advancement of high-throughput DNA sequencing technology has drastically reduced the cost and time of genome sequencing and produced tens of millions of protein sequences [1]. However, determining 3D protein structure through experimental techniques (i.e., X-ray crystallography, or NMR spectroscopy) is still time-consuming, labor-intensive and rather expensive, leaving most proteins without solved structures. The gap between the number of protein sequences and experimentally determined structures is exponentially enlarged [2]. Therefore, developing effective and accurate computational tools that can predict protein structure from its amino acid sequence is one of the most important tasks in bioinformatics and computational biology.

Computational methods for protein structure prediction can be classified as template-based and template-free (ab initio). Template-based modeling methods (TBM) attempt to build the tertiary structure of a target protein by using the known structures of its homologous proteins as template [3, 4, 5]. It is also known as homology modeling or comparative modeling. These methods are able to generate accurate three-dimensional structures if the homologous proteins with known structures can be accurately detected and well aligned with a target protein. Otherwise, it cannot predict the correct structure. Ab initio protein structure prediction is to predict the 3D structure from protein sequence without using known structures as template. Fragment-assembly based modeling is one of the representative ab initio methods for structure prediction [6]. Even though it can predict correct structures for some small proteins, it often fails to build the structures of medium to large proteins with complicated topology. Ab initio protein structure prediction has achieved major breakthroughs in the recent years due to the drastic improvement of the accuracy of residue-residue contact distance prediction based on the co-evolutionary analysis and deep learning [7, 8, 9, 10]. The distance-geometry based ab initio modeling using predicted contact distances as restraints is able to build correct structures of proteins of large size and with complicated topologies on various benchmarks and the recent Critical Assessments of Techniques for Protein Structure Prediction (CASP) [4, 10, 11]. In addition to model construction by template-based modeling or ab initio modeling, model quality assessment and model refinement are also two integral parts of a protein structure prediction system [12, 13, 14].

Figure 1.1 is an overview of our protein structure prediction system [4]. Given a target protein sequence, our method first generates the multiple sequence alignments (MSA) by searching the sequence against the non-redundant sequence database to build sequence profiles (i.e. position specific scoring matrix (PSSM) and hidden Markov model (HMM)) for protein templates identification [15] and multiple sequence alignments for co-evolutionary analysis and two-dimensional(2D) residue-residue contact predictions at multiple distance thresholds (i.e. 6 Å, 7.5 Å, 8 Å, 8.5 Å, and 10 Å) [8]. The sequence profiles are also used to predict several important one-dimensional(1D) protein features including secondary structure, solvent accessibility and disorder regions [16, 17]. The profile or sequence of the target was searched against the template profile/sequence library by a number of sequence alignment tools and classifying protein sequences into folds using deep learning to identify protein templates whose structures were known. The sequence alignments between the target and the identified templates are also used to predict domain boundaries. The regions of the target not aligned with any significant template are modeled by template-free (ab initio) methods with contacts (i.e. CONFOLD2, ROSETTA, UniCON3D and FUSION) [6, 10, 18, 19], and the regions covered by templates are modeled by the multi-template combination modeling approach [3, 20]. Both the fragment-assembly and distance-geometry based ab initio modeling methods are used with predicted contacts to make 3D structure prediction when the target sequence does not have significant templates. A number of structures (i.e. generally more than 100 structures) are generated from various target-template alignments produced by a variety of sequence alignment algorithms or their combinations [21]. Model evaluation plays an important role in protein structure prediction, which evaluates the quality of a protein model without knowing its true structure. We use a deep-learning-based quality assessment method to select the presumably most accurate structural models from all these predicted models. The structure of the selected model is then refined using the model refinement techniques [14].

In this dissertation, I mainly focus on my research of applying deep learning and computational optimization methods for protein structure modeling and model quality assessment, which are two principal problems in bioinformatics. Five contributions are described − (a) DNSS2, a method for protein secondary structure prediction using one-dimensional deep convolution network, (b) DeepSF, a method of applying deep convolutional network to classify protein sequence into one of thousands known folds, (c) CNNQA & DeepRank, two deep neural network approaches to systematically evaluate the quality of predicted protein structures and select the most accurate

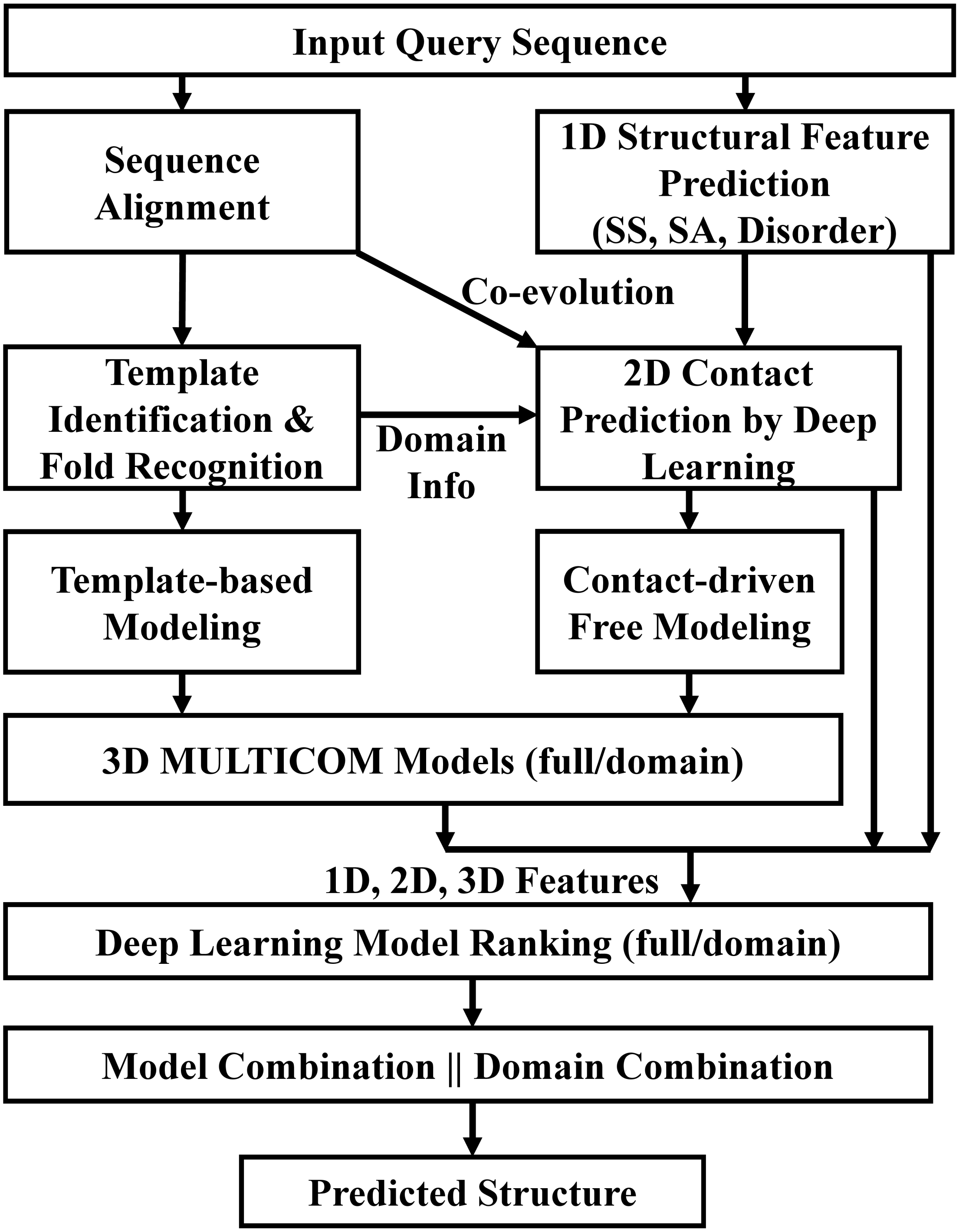

Input Query Sequence
Sequence Alignment
1D Structural Feature Prediction (SS, SA, Disorder)
Co-evolution
Template Identification & Fold Recognition
Domain Info
2D Contact Prediction by Deep Learning
Template-based Modeling
Contact-driven Free Modeling
3D MULTICOM Models (full/domain)
1D, 2D, 3D Features
Deep Learning Model Ranking (full/domain)
Model Combination || Domain Combination
Predicted Structure

Figure 1.1: The MULTICOM protein tertiary structure prediction system

model as the final protein structure prediction, (d) MULTICOM, a protein structure prediction system empowered by deep learning and protein contact prediction, (e) SAXSDOM, a data-assisted method for protein domain assembly using small-angle X-ray scattering data.

Chapter 2 of this dissertation mainly describes the deep learning application in protein secondary structure prediction. We designed several advanced one-dimensional deep convolution networks to predict secondary structures (e.g., deep convolutional/ recurrent/residual/memory/fractal/inception networks). The main content is from the following deposited paper:

Hou, J., *Guo, Z., & Cheng, J. (2019). DNSS2: improved ab initio protein secondary structure prediction using advanced deep learning architectures. bioRxiv, 639021. [22]*

Chapter 3 will describe the deep learning application for protein fold recognition. We developed a new deep-learning-based methods to improve template identification for hard proteins that have little sequence similarity with known structures. Instead of using traditional approaches that identify protein pairs with the same fold based on their pairwise sequence/profile similarities, we utilized the learning power of deep learning to directly classify the target protein to one of thousands of folds. This improved the sensitivity of detecting remote homologous proteins that share the same fold. This chapter is mainly from the content of published paper as follows:

Hou, J., *Adhikari, B., & Cheng, J. (2017). DeepSF: deep convolutional neural network for mapping protein sequences to folds. Bioinformatics, 34(8), 1295-1303.[23]*

Chapter 4 describes a novel single-model quality assessment (QA) method CNNQA, which predicts the absolute local quality of a single protein model based on a deep one-dimensional convolutional neural network (1DCNN). The main content of this chapter is from the following deposited paper:

***Hou, J.**, Cao, R., & Cheng, J. (2019). Deep convolutional neural networks for predicting the quality of single protein structural models. bioRxiv, 590620.[24]*

Chapter 5 describes a new deep-learning-based consensus method (DeepRank) for protein quality assessment that integrates multiple QA methods and residue−residue contact predictions for predicting the global quality of models. The method shows a significant improvement compared to the individual QA methods used to generate input features and is more consistent in selecting models of better quality. This method was officially ranked No.1 in ranking protein structural models in the 13^{th} Critical Assessment of Techniques for Protein Structure Prediction (CASP13).

In chapter 5, we also describe the method of our protein structure prediction system (MULTICOM) which is driven by deep learning and contact prediction. The method was officially ranked 3^{rd} out of all 98 human and server predictors in CASP13 (2018). The main content of this chapter comes from the following publication:

***Hou, J.**, Wu, T., Cao, R., & Cheng, J. (2019). Protein tertiary structure modeling driven by deep learning and contact distance prediction in CASP13. Proteins: Structure, Function, and Bioinformatics. [4]*

Chapter 6 describes a novel framework of applying machine learning and computational optimization approaches to improve the protein domain assembly by incorporating experimental restraints from small-angle X-ray scattering (SAXS) data. The main content is from the following deposited paper:

***Hou, J.**, Adhikari, B., Tanner, J. J., & Cheng, J. (2019). SAXSDom: Modeling multi-domain protein structures using small-angle X-ray scattering data. bioRxiv, 559617. [25]*

Chapter 2

DNSS2: improved ab initio protein secondary structure prediction using advanced deep learning architectures

2.1 Abstract

Accurate prediction of protein secondary structure (alpha-helix, beta-strand and coil) is a crucial step for protein inter-residue contact prediction and ab initio tertiary structure prediction. In a previous study, we developed a deep belief network-based protein secondary structure method (DNSS1) and successfully advanced the prediction accuracy beyond 80%. In this work, we developed multiple advanced deep learning architectures (DNSS2) to further improve secondary structure prediction. The major improvements over the DNSS1 method include (i) designing and integrating six advanced one-dimensional deep convolutional/recurrent/residual/memory/fractal/inception networks to predict secondary structure, and (ii) using more sensitive profile features inferred from Hidden Markov model (HMM) and multiple sequence alignment (MSA).

Most of the deep learning architectures are novel for protein secondary structure prediction. DNSS2 was systematically benchmarked on two independent test datasets with eight state-of-art tools and consistently ranked as one of the best methods. Particularly, DNSS2 was tested on the 82 protein targets of 2018 CASP13 experiment and achieved the best Q3 score of 83.74% and SOV score of 72.46%. DNSS2 is freely available at: `https://github.com/multicom-toolbox/DNSS2`.

2.2 Introduction

Three major types of protein secondary structure are alpha-helix (H), beta-strand (E) and coil state (C) [26], each of which represents the local structure state of an amino acid in a folded polypeptide chain. The predicted information of protein secondary structure is useful for many applications in computational biology, such as protein residue-residue contact prediction [8, 9, 27], protein folding [23, 28, 29], ab-initio protein structure modeling [6, 10, 30] and protein model quality assessment [31, 32]. For instance, secondary structure prediction was widely utilized in the template-based structure modeling through threading or comparative modeling on those proteins that have structurally determined homologs [3, 5, 30], and in ab-initio modeling for those proteins whose sequences share few sequential similarities with known solved structures [33, 34].

The progress in protein secondary structure prediction over the past few decades can be generally summarized from two aspects: the discovery of novel features that are useful for prediction and the development of effective machine learning algorithms [35, 36]. The early attempts utilized statistical propensities of single amino acid observed from known structures to identify secondary structures in proteins [37]. The subsequent improvements came from the inclusion of sequence evolutionary profile features inferred from multiple sequence alignment (MSA) such as position-specific scoring matrices (PSSM) [16, 38, 39, 40, 41, 42]. In addition to the PSSM, the Hidden Markov model (HMM) profiles derived from HHblits [15] was proposed for predicting protein structural properties [43]. Atchley's factors were also included in some studies to capture the similarity between the types of amino acids [44, 45].

Meanwhile, the machine learning algorithms for protein secondary structure prediction also continued to improve. Several early approaches applied shallow neural networks [46, 47], information theory and Bayesian analysis [48, 49, 50] to secondary structure prediction. PSIPRED [40] method proposed a two-stage neural network to predict the secondary structure from the PSI-BLAST sequence profiles. SSpro [42] used bi-directional recurrent neural networks to capture the long-range interactions between amino acids. Deep learning techniques recently achieved significant success in secondary structure prediction [39, 51, 52, 53, 45, 54]. DNSS [45] applied an ensemble of deep belief networks to predict 3-state secondary structure. SPIDER2 [55] employed stacked sparse auto-encoder neural networks to predict the several structural properties iteratively, and this method was further advanced by bidirectional long- and short-term memory (LSTM) neural networks to capture the long-range interactions [53]. DeepCNF [54] integrated the convolutional neural networks with conditional random-field to learn the complex sequence-structure relationship and interdependence between sequence and secondary structure. Porter 5.0 [56] ensembled seven bidirectional recurrent neural networks to improve the protein structure prediction. Assisted with the power of deep learning, the accuracy of 3-state secondary structure prediction has been successfully improved above 84% [51, 53, 54] on some benchmark datasets.

In this work, we developed an improved version of our ab initio secondary structure method using multiple advanced deep learning architectures (DNSS2). Three major improvements have been made over the original DNSS method. Firstly, besides the PSSM profile features and Atchley's factors used in DNSS, we incorporated several novel features such as the emission and transition probabilities derived from Hidden Markov model (HMM) profile [15], and profile probabilities inferred from multiple sequence alignment (MSA) [16]. All the three new features represent the evolutionary conservation information for amino acids in sequence. Secondly, we designed and integrated six types of advanced one-dimensional deep networks for protein secondary structure prediction, including traditional convolutional neural network (CNN) [57], recurrent convolutional neural network (RCNN) [58], residual neural network (ResNet) [59], convolutional residual memory networks (CRMN) [60], fractal networks [61], and Inception network [62]. The ensemble of six networks from DNSS2 significantly improved the secondary structure prediction. Finally, DNSS2 was trained on a large dataset, including 4,872 non-redundant protein structures with less than 25% pairwise sequence identity and 2.5 Å resolution. Our method was extensively tested on the independent dataset and the latest CASP13 dataset with other state-of-art methods and delivered the state-of-the-art performance.

2.3 Materials and Methods

2.3.1 Experimental design

In this work, the main objective was to improve the secondary structure prediction by developing more advanced deep learning architectures and introducing more useful features. In the process, we have developed a systematic framework to effectively build deep learning architectures and obtain features to improve secondary structure prediction. **Figure 2.1** provides an overview of our experimental design. **Figure 2.1(A)** lists the six major steps of designing, training and testing deep learning architectures. **Figure 2.1(B)** illustrates the process of creating training and validation datasets. The key analysis is to design appropriate architectures and investigate if they can improve prediction accuracy. Six different deep neural network architectures were evaluated in the study, including convolutional neural network (CNN) [57], recurrent convolutional neural network (RCNN) [58], ResNet [59], convolutional recurrent memory network (CRMN) [60], FractalNet [61], and Inception network [62]. Most of these architectures were applied to secondary structure prediction for the first time. The detailed description of each network is included in Section 2.3.4. To ensure a fair comparison, each network was optimized using the original feature profiles of training proteins and evaluated on the same validation set of DNSS1. The network that achieved the best Q3 accuracy was selected to explore the feature space on the profiles derived from multiple sequence alignments (MSA) generated by PSI-BLAST [38] and HHblits [15], Atchley factors, and emission/transition probabilities inferred from the Hidden Markov model (HMM) profile. The optimal feature set was determined according to the highest Q3 accuracy on the validation datasets. The networks were then re-trained using the optimal input profiles to obtain the best models.

Since combining predictors generally improved the prediction accuracy, the different combinations of networks were also evaluated. Finally, after the optimal sets of deep learning architectures and feature profiles were determined, all networks were re-trained on the large dataset that was manually curated including the non-redundant proteins whose structures have been released publicly before 2018. The final networks were used to predict the secondary structure for the test proteins. The probabilities of the three states (i.e., helix, sheet, and coil) for each residue predicted by six networks were averaged to make the final secondary structure prediction. Our method was then benchmarked with other state-of-art methods on the two independent test datasets.

2.3.2 Datasets and evaluation metric

As described in section 2.3.1, two training datasets were used in our experiment. In the first stage, the original DNSS dataset [45] that included 1,230 training proteins

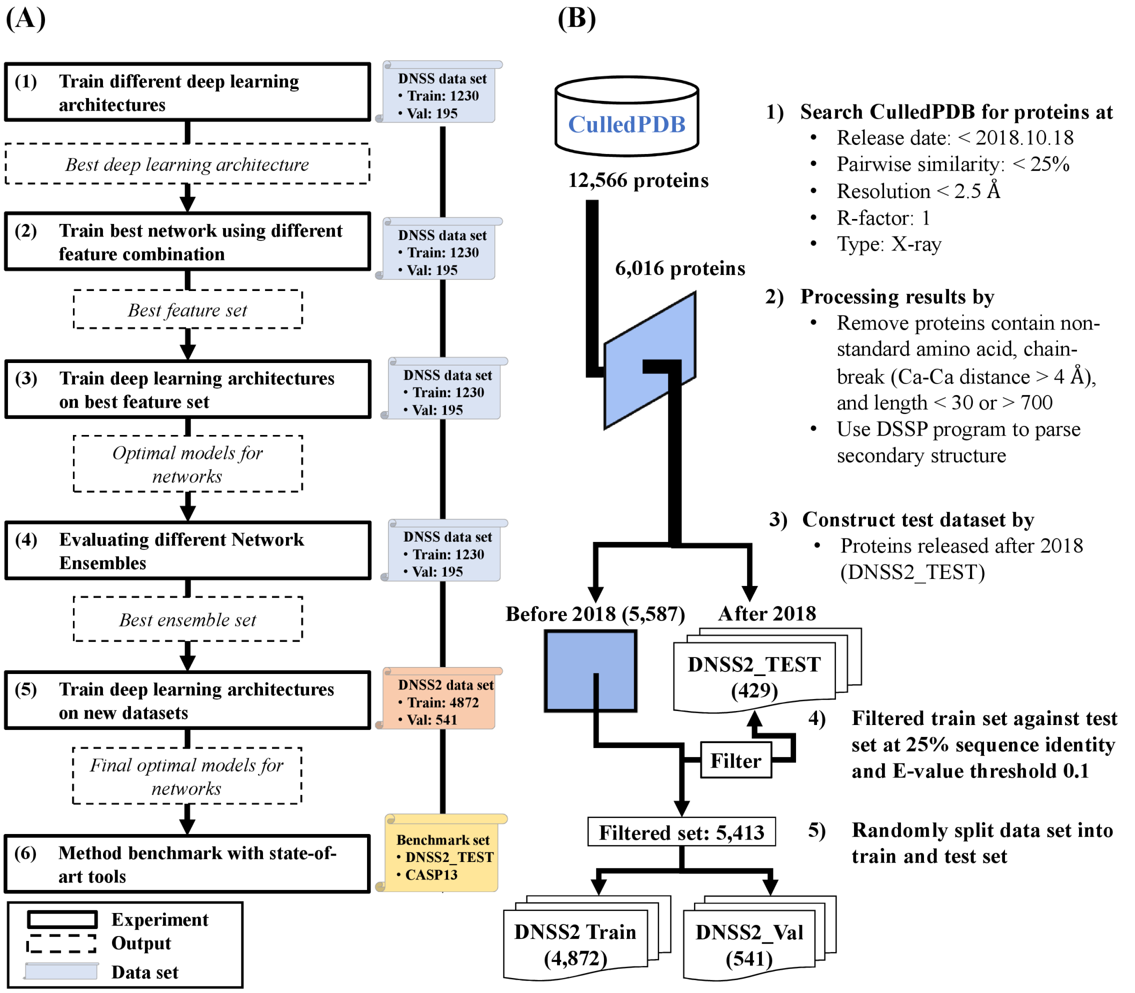

(A)
(1) Train different deep learning architectures
DNSS data set
• Train: 1230
• Val: 195
Best deep learning architecture
(2) Train best network using different feature combination
DNSS data set
• Train: 1230
• Val: 195
Best feature set
(3) Train deep learning architectures on best feature set
DNSS data set
• Train: 1230
• Val: 195
Optimal models for networks
(4) Evaluating different Network Ensembles
DNSS data set
• Train: 1230
• Val: 195
Best ensemble set
(5) Train deep learning architectures on new datasets
DNSS2 data set
• Train: 4872
• Val: 541
Final optimal models for networks
(6) Method benchmark with state-of-art tools
Benchmark set
• DNSS2_TEST
• CASP13
Experiment
Output
Data set
(B)
CulledPDB
12,566 proteins
6,016 proteins
1) Search CulledPDB for proteins at
• Release date: < 2018.10.18
• Pairwise similarity: < 25%
• Resolution < 2.5 Å
• R-factor: 1
• Type: X-ray
2) Processing results by
• Remove proteins contain non-standard amino acid, chain-break (Ca-Ca distance > 4 Å), and length < 30 or > 700
• Use DSSP program to parse secondary structure
3) Construct test dataset by
• Proteins released after 2018 (DNSS2_TEST)
Before 2018 (5,587)
After 2018
DNSS2_TEST (429)
Filter
4) Filtered train set against test set at 25% sequence identity and E-value threshold 0.1
Filtered set: 5,413
5) Randomly split data set into train and test set
DNSS2 Train (4,872)
DNSS2_Val (541)

Figure 2.1: Overview of the experimental workflow for improving secondary structure prediction. (A) Six principal steps are conducted to construct and train deep networks. The solid box represents an analysis step. The dashed box represents the output from the previous step. The scroll represents the dataset used in each step. (B) Dataset generation and filtering process.

and 195 validation proteins was utilized to investigate whether the deep learning architectures and novel features can boost the prediction accuracy.

To utilize more data available since DNSS1 was published, a new, larger training set of DNSS2 was constructed from CullPDB [63] curated on 18 October 2018 (**Figure 2.1(B)**). The dataset consists of 12,566 proteins that share less than 25% sequence identity with 2.5 Å resolution cutoff and R-factor cutoff 1. The structures of all the proteins were determined by X-ray crystallography. The dataset was then filtered by removing proteins with non-standard amino acids, chain-break (i.e., distance of adjacent Ca-Ca atoms is larger than 4 Å), and sequence length shorter than 30 or longer than 700 amino acids. Considering all external methods benchmarked in this work were developed prior to year 2018, the proteins that were released after Jan 1st, 2018 were extracted as independent test set (DNSS2_TEST). The resulting set of proteins was further filtered against DNSS2_TEST set using CD-HIT suite [64] with criteria of 25% sequence identity cutoff and e-value threshold 0.1. Finally, 5,413 proteins released prior to Jan 1st, 2018 were obtained as our training set, in which 4,872 proteins were used for network training (DNSS2_TRAIN) and 547 proteins were used for model selection (DNSS2_VAL). In addition, the proteins of the CASP13 (2018) experiment were collected and the ones with at least 25% sequence identity with training proteins were removed, which results in a set of 82 test proteins. The proteins were also classified into template-based (TBM) and free-modeling (FM) targets based on the official CASP definition (CASP 13, 2018, `http://www.predictioncenter.org/casp13/index.cgi`). In summary, the final test set contain 429 proteins from DNSS2_TEST and 82 proteins from CASP13.

We evaluated our secondary structure prediction based on two primary metrics: Q3 accuracy and Segment Overlap measure (SOV). Q3 score represents the percent of correctly predicted secondary structure states in a protein. SOV score measures the similarity between the predicted segments of continuous structure states and those in the experimental structure [45, 65]. The Q3 and SOV scores are complementary with each other for secondary structure evaluation. All training and testing proteins' structure files were parsed by DSSP program [66] to obtain the real secondary structure classification for each amino acid for training and evaluation.

2.3.3 Input features

The profile of each amino acid is represented by 21 numbers from PSI-BLAST-based position specific scoring matrix (PSSM), 20 emission probabilities and 7 transition probabilities extracted from Hidden Markov Model (HMM) profile, 20 probabilities of standard amino acid calculated from the multiple sequence alignment (MSA) and 5 numbers derived from Atchley's factor. These features (73 numbers in total) represent the evolutionary conservation and physicochemical properties for residues in a protein sequence.

PSI-BLAST was run to generate multiple sequence alignment and PSSM profile through searching a sequence against filtered UniProt sequence database at 90% sequence identity (UniRef90) [67] with three iterations and an e-value cutoff 0.001 ('-evalue .001 -inclusion_ethresh .002'). Less stringent threshold was used ('-evalue 10 -inclusion_ethresh 10') in case some proteins did not have homologous sequences returned. In a PSSM profile, each position is represented by 20 numbers related to the probabilities for 20 standard amino acids appearing at the position in the multiple sequence alignment. In addition, the sequence information in the second to the last column in PSI-BLAST profile is given for each residue. HMM profile was generated by running three iterations of 'HHblits' against the uniclust30 database (version: October 2017) [68]. Two types of probabilities were associated with each residue in a HMM profile: emission probability and transition probability. Emission probability represents the probability of a given amino acid occurring at the position in the multiple sequence alignment. The transition probability represents the probability transiting from an alignment state (i.e., match, insertion, and deletion) to another. Similar to PSSM, the emission frequencies of the 20 standard amino acid for each residue were reported in the HMM profile, and the probabilities were calculated according to formula:

$$p_{ik} = 2^{(\frac{-Freq_{ik}}{1000})}$$

where i is the i-th residue in sequence and k is the k-th standard amino acid. And the probability is set to 0 if the frequency is denoted as '*' in the HMM profile. The transition probabilities for each amino acid were also derived in the same fashion. In total, 20 emission probabilities and 7 transition probabilities for each amino acid were collected to represent the residue conservation inferred from HMM.

Since HHblits was more sensitive to identify distant homologous sequences than PSI-BLAST, the probability matrix of amino acids was also calculated from the multiple sequence alignment (MSA) generated by HHblits. The conversion from MSA to a probability matrix follows the same calculation as SSpro [16].

2.3.4 Deep learning architectures

A widely used deep learning architecture in bioinformatics is deep convolutional neural networks (CNN). Convolutional neural networks have some distinctive advantages over the traditional neural networks for the bioinformatics problems in several ways: (1) it can learn informative representation directly from sequence features without requiring segmentation (e.g., sliding window) or dimension reduction (e.g., principal component analysis) techniques; (2) the convolutional network can learn both local and global features to discover complex patterns; and (3) the architecture is independent of input size (i.e., length or volume). In this work, we design a standard CNN and five advanced deep learning architectures based on both convolutional and other useful operations as in **Figure 2.2**.

Figure 2.2(A) illustrates our standard convolutional neural network (CNN) for secondary structure prediction, consisting of a sequence of convolutional blocks, each of which contains a convolutional layer, a batch-normalization layer, and an activation layer. The original input is a $L \times K$ vector (X), where L is sequence length and K is the number of features per residue position in the sequence. For each convolution block, the feature maps are obtained after the convolution operation is applied by multiplying the weight matrices (called filters, W) with a window of local features on the previous input layer and adding bias vectors (b) according to the formula: $X^{l+1} = W^{l+1} * X^{l} + b^{l+1}$, where l is the layer number. The batch normalization layer is added to obtain a Gaussian normalization of convolved features coming out of each convolutional layer. Then an activation function such as rectified linear function (i.e., ReLU) is applied to extract non-linear patterns of the normalized hidden features. To avoid overfitting, regularization approaches such as dropout [69] can be applied in the hidden layers. The final output node (also a filter) in the output cell uses the softmax function to classify the input of each residue position from its previous layer into one of three secondary structure states. The output is a $L \times 3$ vector, holding the predicted probability of three secondary structure states for each of L positions in a sequence. The final optimal CNN architecture includes 6 convolutional blocks, in which the filter size (window size) for each convolutional layer is 6, and the number of filters (feature maps) in each convolution layer is 40.

The residual network (ResNet) was designed to make traditional convolutional neural network deeper without gradient vanishing. The architecture constructs many residual blocks and stacked up them to form a deeper network, as shown in **Figure 2.2(B)**. In each residual block, the input X^{l} is fed into a few convolutional layers to obtain the non-linear transformation output $G(X^{l+1})$. In order to make the network deeper, an extra skip connection (i.e., short-cut) is added to copy the input X^{l} to the output of non-linear transformation layer, where $X^{(l+1)*}$ can be represented as $X^{(l+1)*} = X^l + G(X^{l+1})$ before applying another ReLU non-linearity. This process makes neural network deeper by adding shortcuts to facilitate gradient back-propagation during training and achieve better performance. The residual blocks with different configuration can be stacked to achieve higher accuracy. For instance, the final best architecture in DNSS2 is made up of 13 residual blocks, each of which includes 3 convolutional layers with filter size 1, 3, 1 respectively. The first three residual blocks used 37 filters to learn features, while the middle four blocks used 74 filters for each convolution layer, and the last six residual blocks used 148 filters. In total, 39 convolutional layers are included in the final residual network. In the network, the dropout and batch normalization were also added to prevent network from overfitting.

Inception network is an advanced architecture for building deeper networks by repeating a bunch of inception modules, as shown in **Figure 2.2(C)**. Instead of trying to determine the best values for certain hyper-parameters (i.e., number of filter size, number of layers, inclusion of pooling layer), inception network proposes to concatenate outputs of hidden layers with different configuration through an inception module and trains the network to learn patterns from the combination of diverse hyper-parameters. Despite its high computation cost, inception network has performed remarkably well in many applications [51, 62]. For secondary structure prediction, a combination of three filter sizes $1 \times K$, $3 \times K$ and $5 \times K$ was applied to convolve feature input, where K is the number of original input features for each residue position. The concatenation of the convolution outputs is fed into an activation layer for non-linear activation calculation. This kind of inception module is repeated to make a deeper network. After the parameter tuning, the optimal inception network is comprised of three inception blocks with 24 convolution layers included.

In addition, we designed three more deep learning architectures: recurrent convolutional neural network (RCNN) [58], convolutional residual memory networks

(CRMN) [60], and fractal network for secondary structure prediction. The recurrent convolutional neural network (RCNN) was designed to model sequential dependency hidden inside the sequential features (**Figure 2.2(D)**), It firstly extracts the higher-level feature maps by a convolution block, and then uses a recurrent neural network (i.e., bi-directional Long-Short-Term Memory (LSTM) network) for modeling the inter-dependence among the convolved features. Such a recurrent convolutional block with 4 convolutional layers included is repeated 5 times to build a deep recurrent convolutional neural network for secondary structure prediction in this work. The CRMN network augmented the architectures by integrating convolutional residual networks with LSTM (**Figure 2.2(E)**) (e.g., 2 residual blocks and 2 LSTM in the network). Both methods advanced the convolutional neural network by introducing the memory mechanisms of recurrent neural network (RNN). Moreover, inspired by ResNet and Inception Network, we built a Fractal network stacking up different number of convolution blocks in both parallel and hierarchical fashion by adding several shortcut paths to connect lower-level layers and higher-level layers, as shown in **Figure 2.2(F)**. After tuning, the fractal network was assembled with 16 convolution layers for one fractal block.

2.3.5 Training and evaluation procedure

Deeper networks with complex architectures are generally difficult to train effectively due to the high-dimensional hyper-parameter space. To obtain good performance on specific feature sets within a reasonable amount of time for each deep network, we developed an efficient heuristic random sampling approach for model hyperparameter optimization. Specifically, based on the several trials on network training, we first determined heuristically a reasonable range for each type of the network hyperparameters, including the number of filters from 20 to 50, the number of convolution blocks from 3 to 7, and the filter size from 3 to 7. For each subsequent trial, the values

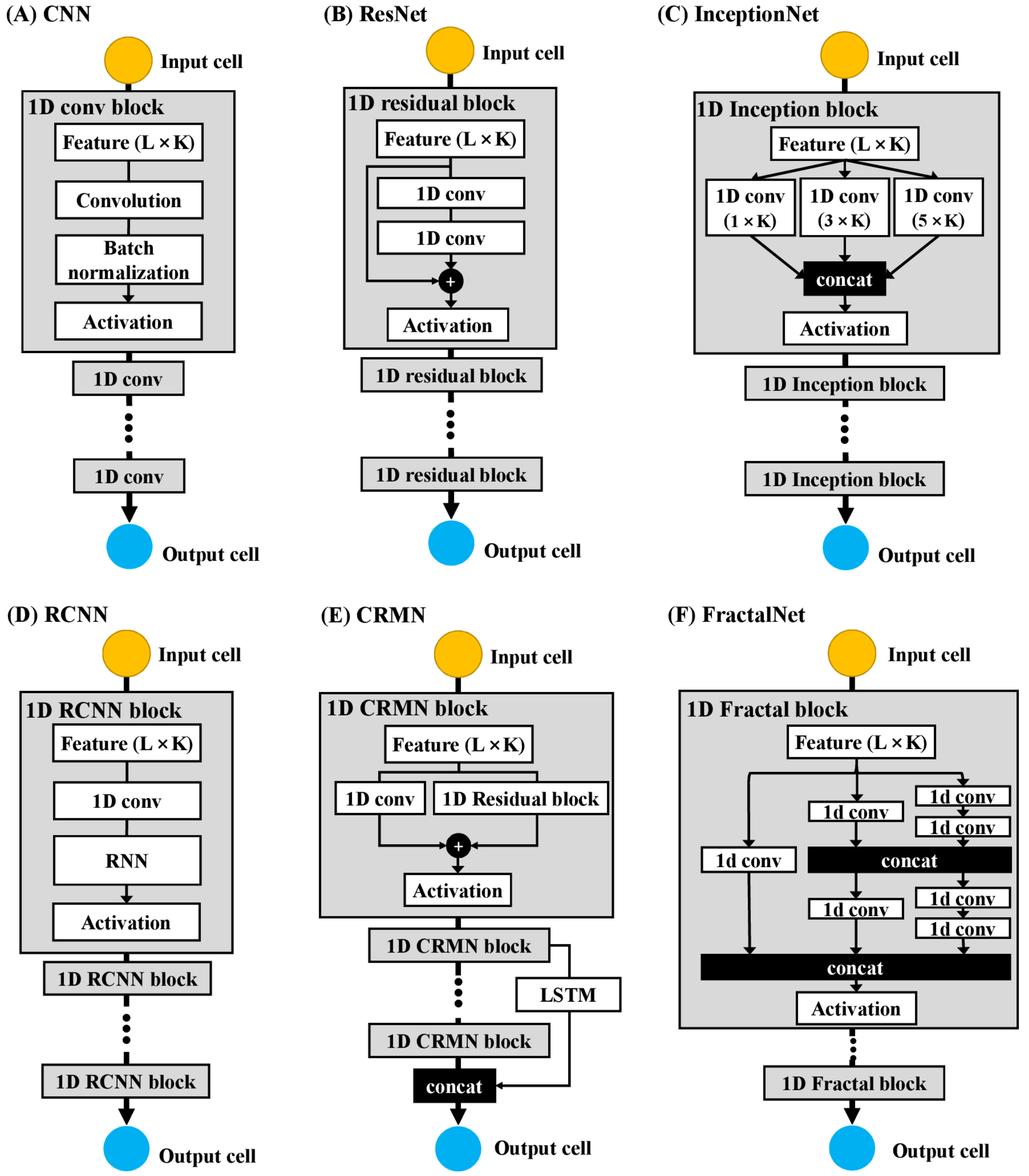


Figure 2.2: Six deep learning architectures: (A) CNN, (B) ResNet, (C) InceptionNet, (D) RCNN, (E) CRNN, (F) FractalNet for secondary structure prediction. L: sequence length; K: number of features per position.

of hyper-parameters were randomly sampled from their specified range and the Q3 accuracy of the network on the validation dataset under the specific parameter combination was assessed. For each deep network, the best parameter set was determined after 100 trials were evaluated. We found that using the random sampling technique was able to generate better models in most cases and was also more efficient than the traditional grid search or greedy search.

The performance of different deep architectures and different feature profiles on the secondary structure prediction were rigorously examined using the training and validation set from original DNSS method. After the parameters and input features were determined, we trained each deep network on the latest curated dataset (DNSS2_TRAIN) and selected best models using the Q3 accuracy on the independent validation dataset (DNSS2_VAL). We used the Keras library (`http://keras.io/`) along with Tensorflow as a backend to train all networks. The performance of DNSS2 was evaluated on the two independent datasets and compared with a variety of the state-of-art secondary structure prediction tools, including SSpro5.2 [16], PSSpred [70], MUFOLD-SS [51], DeepCNF [54], PSIPRED [71], SPIDER3 [53], Porter 5 [56] and our previous method DNSS1 [45]. All the methods were assessed according to the Q3 and SOV scores on each dataset.

2.4 Results and Discussion

2.4.1 Benchmarking different deep architectures of DNSS2 with DNSS1

The first evaluation was to investigate whether the new deep architectures networks (DNSS2) outperform the deep belief network (DNSS1) for the secondary structure prediction. In order to fairly compare them, we trained and validated the six deep networks on the original input features of the same 1,230 training and 195 validation proteins used to train and test DNSS1. **Table 2.1** compares the Q3 and Sov scores of DNSS1 and DNSS2 architectures on the validation set. The results show that five out of six new advanced deep networks (RCNN, ResNet, CRMN, FractalNet, and InceptionNet) except the standard CNN network obtain higher Q3 scores than the deep belief network that used in DNSS1. InceptionNet worked best among individual deep architectures. The ensemble of the six deep architectures (DNSS2) achieved the highest Q3 score of 83.04%, better than all the six individual deep architectures and 79.1% Q3 score of DNSS1.

Method	**Q3(%)**	**Sov(%)**
DNSS1	79.1	72.38
DNSS2_CNN	77.86	68.42
DNSS2_RCNN	79.87	72.34
DNSS2_ResNet	79.61	69.94
DNSS2_CRMN	79.32	69.21
DNSS2_FractalNet	79.85	72.82
DNSS2_InceptionNet	80.68	72.74
DNSS2	83.04	72.74

Table 2.1: Performance of the six different deep architectures and their ensemble on the DNSS1 validation dataset. DNSS2 represents the ensemble of six deep architectures (CNN, RCNN, ResNet, CRMN, FractalNet and InceptionNet).

2.4.2 Impact of different input features

After the best deep learning architecture (i.e., InceptionNet) was determined, it was utilized to examine the impact of the different input features including PSSM, Atchley factor (FAC), Emission probabilities (Em), Transition probabilities (Tr), and amino acids probabilities from HHblits alignments (HHblitsMSA). In this analysis, the protein sequence databases required for alignment generation were updated to latest and all the input features for DNSS1 datasets were regenerated. Specifically, the Uniref90 database that was released at October 2018 was used to generate PSSM profiles by PSI-BLAST, and the latest version of Uniclust30 database (October 2017) was used to generate HMM profiles by HHblits. The Inception network was then trained on the 1,230 proteins using the combination of five kinds of features. We tested six feature combinations shown in **Table 2.2**. Hyper-parameter optimization was applied to obtain the best model on each feature combination. **Table 2.2** shows the performance of different input feature combinations with the inception network on the validation dataset of 195 proteins. Adding the emission profile inferred from HMM model on top of PSSM and Atchley factor features increased the Q3 score from 79.81% to 82.31%. Integrating all the five kinds of features will yield the highest Q3 score (i.e., 82.72%) and Sov score (75.89%).

The performance of the six deep architectures and their ensemble on the latest features (the combination of all five kinds of features) of the DNSS1 validation dataset was also reported in **Table 2.3**. All six architectures were re-trained on the 1,230 proteins and evaluated on the validation dataset. Compared to the results in **Table 2.1**, the prediction accuracy of all the networks on the validation set was improved. The Q3 and SOV scores of the ensemble (DNSS2) were increased to 83.84% and 75.5%, respectively. The results indicate that the update of the protein sequence databases helps improve prediction accuracy.

2.4.3 Comparison of DNSS2 with eight state-of-the-art tools on two independent test datasets

DNSS2 was compared with eight state-of-art methods including SSPro5.2, DNSS1, PSSpred, MUFOLD-SS, DeepCNF, PSIPRED, SPIDER3, and Porter 5 on the DNSS2_TEST dataset. The test dataset contains non-redundant proteins released after Jan 1st, 2018. All the tools were downloaded and configured based on their

Rank	Feature Name	Q3(%)	SOV(%)
1	PSSM + FAC + Em + Tr + HHblitsMSA	82.72	75.89
2	PSSM + FAC + Em + Tr	82.36	76.03
3	PSSM + FAC + Em	82.31	74.15
4	PSSM + FAC + HHblitMSA	81.98	74.67
5	PSSM + FAC + Tr	80.13	71.61
6	PSSM + FAC	79.81	71.43

Table 2.2: Performance of different input feature combinations on the validation dataset of 195 proteins. PSSM, FAC, Em, Tr, HHblitsMSA denote five kinds of features: PSSM, Atchley factor, Emission probabilities, Transition probabilities, amino acid probabilities from HHblits alignments.

Method	**Q3(%)**	**Sov(%)**
DNSS2_CNN	80.29	72.1
DNSS2_RCNN	81.83	73.97
DNSS2_ResNet	81.53	73.71
DNSS2_CRMN	81.91	73.37
DNSS2_FractalNet	82.02	73.8
DNSS2_InceptionNet	82.74	75.3
DNSS2	83.84	75.5

Table 2.3: Performance of the six different deep learning architectures (CNN, RCNN, ResNet, CRMN, FractalNet, and InceptionNet) and their ensemble (DNSS2) on DNSS1 validation dataset and the updated protein sequence database.

instructions. The sequence databases that the tools require were updated to the latest version.

The Q3 score of each tool on the test dataset was reported in **Table 2.4**. In general, DNSS2 is comparable to the two predictors (Porter 5 and SPIDER3) on this dataset and outperforms the other six methods. Specifically, DNSS2 achieved a Q3 accuracy of 85.02% and SOV accuracy of 76.01% on the DNSS2_TEST dataset, which was significantly better than DNSS 1.0 on the DNSS2_test dataset with p-value equal to 2.2E-16.

In addition to the DNSS2_test dataset, we also compared these methods on the 82 protein targets of 2018 CASP13 experiment, which share less than 25% sequence identity with the training proteins of DNSS2. Both template-based (TBM) and free-modeling (FM) protein targets were used to evaluate the methods and the results are summarized in the **Table 2.5**. Consistent with the performance on the DNSS2_test dataset shown in **Table 2.4**, DNSS2, SPIDER3 and Porter 5 performed best, while DNSS2 achieved slightly better performance than SPIDER3 and Porter 5. **Figure 2.3** plots the distribution of the Q3 scores for all CASP13 targets obtained by DNSS2 and the other eight methods. In general, the distribution of DNSS2 consistently shifts to higher Q3 score compared with other methods, even though the distribution of DNSS2 largely overlaps with that of SPIDER3 and Porter 5.

Table 2.6 summarized the confusion matrix of predictions of three kinds of secondary structures (helix, sheet, coil) by DNSS2 on the CASP13 dataset. DNSS2 yields the highest accuracy for helical prediction (87.91%), followed by the coil prediction (80.21%) and the sheet prediction (76.45%). The prediction errors between helix, sheet, and coil was also reported. The error rate of misclassifying helix as sheet is the lowest (0.57%) and sheet as coil is the highest (22.46%).

Method	**Q3** (%)	**SOV** (%)
SSPro5.2	79.26	70.78
PSSpred	81.86	71.65
MUFOLD	81.85	73.56
DeepCNF	82.85	70.57
PSIPRED	83.94	74.49
SPIDER3	85.34	77.61
Porter 5	85.07	76.79
DNSS1	80.14	73.63
DNSS2	85.02	76.01

Table 2.4: Q3 scores of 9 secondary structure prediction methods on DNSS2_test dataset. Three methods (SPIDER3, Porter5, DNSS2) have Q3 score higher than 85%.

Method	**All**		**TBM**		**FM**	
	Q3 (%)	**SOV** (%)	**Q3** (%)	**SOV** (%)	**Q3** (%)	**SOV** (%)
SSPro5.2	76.73	69.94	78.16	71.32	76.12	70.88
PSSpred	78.8	67.85	81.32	72.11	76.99	64.55
MUFOLD	79.58	71.74	79.71	74.13	79.8	70.79
DeepCNF	80.24	69.5	82.34	73.68	78.36	65.55
PSIPRED	80.7	72	83.67	76.72	78.41	68.14
SPIDER3	81.73	74.39	84.84	78.31	78.89	71.1
Porter5	82.07	74.61	84.79	78.98	79.42	70.3
DNSS1	77.06	70.4	79.48	73.58	75.46	68.79
DNSS2	82.2	73.03	85.37	76.98	79.82	70.56

Table 2.5: Comparison of methods on the CASP13 dataset in terms of all CASP13 targets, template-based targets, and template-free targets.

	C pred	E pred	H pred
Coil (C)	80.21%	9.51%	10.28%
Sheet (E)	22.46%	76.45%	1.10%
Helix (H)	11.52%	0.57%	87.91%

Table 2.6: Confusion matrix of helix, sheet and coil predicted by DNSS2 on CASP13 dataset.

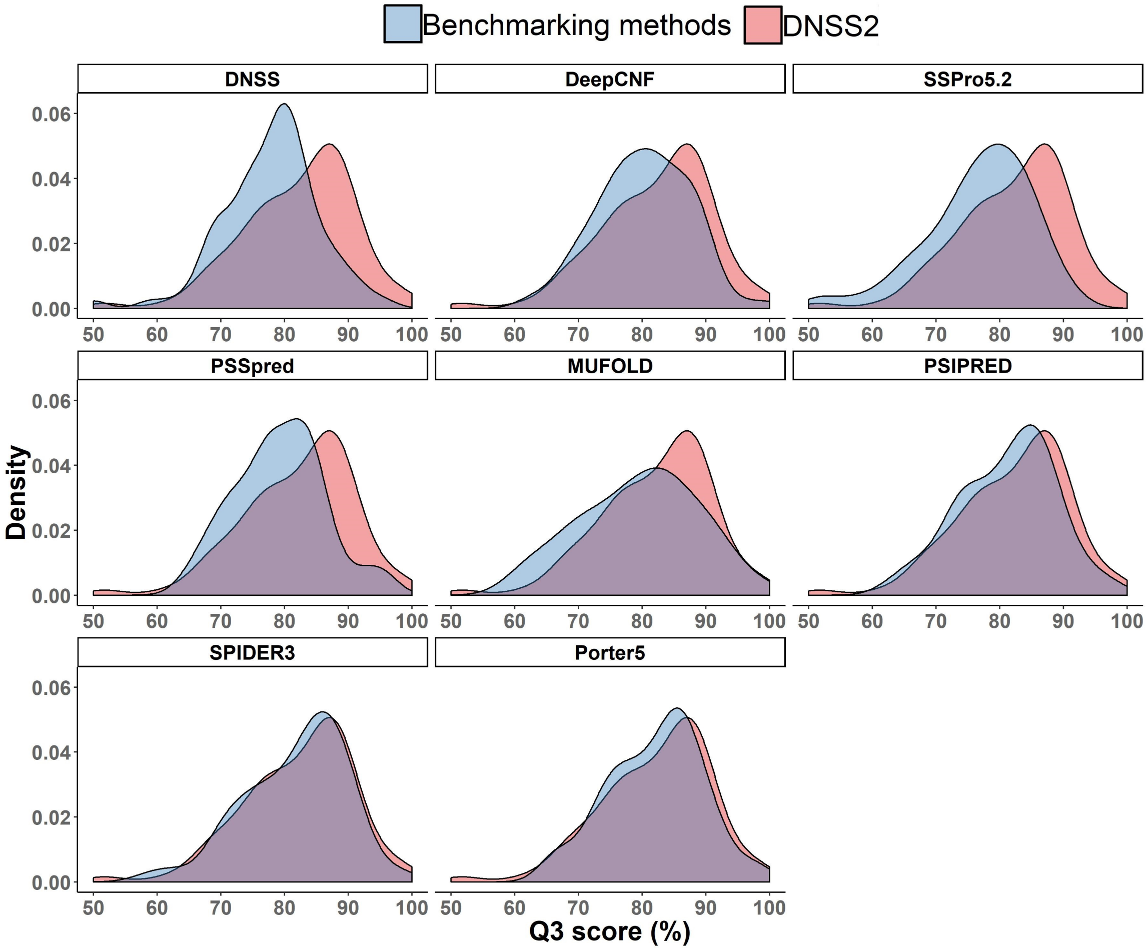


Figure 2.3: Comparison of the distribution of Q3 scores of eight existing methods and that of DNSS2 on all CASP13 targets.

2.4.4 Conclusion

In this work, we developed several advanced deep learning architectures and their ensemble to improve secondary structure prediction. We investigated six advanced deep learning architectures and five kinds of input features on secondary structure prediction. Several deep learning architectures such as inception network, fractal network, and recurrent convolutional memory network are novel for protein secondary structure prediction and performed better than the deep belief network. The performance of the deep learning method is comparable to or better than seven external state-of-the-art methods on the two independent test datasets. Our experiment also demonstrated that emission/transition probabilities extracted from hidden Markov model profiles are useful for secondary structure prediction.

Chapter 3

DeepSF: deep convolutional neural network for mapping protein sequences to folds

3.1 Abstract

Protein fold recognition is an important problem in structural bioinformatics. Almost all traditional fold recognition methods use sequence (homology) comparison to indirectly predict the fold of a target protein based on the fold of a template protein with known structure, which cannot explain the relationship between sequence and fold. Only a few methods had been developed to classify protein sequences into a small number of folds due to methodological limitations, which are not generally useful in practice. We develop a deep 1D-convolution neural network (DeepSF) to directly classify any protein sequence into one of 1,195 known folds, which is useful for both fold recognition and the study of sequence-structure relationship. Different from traditional sequence alignment (comparison) based methods, our method automatically extracts fold-related features from a protein sequence of any length and maps it to the fold space. We train and test our method on the datasets curated from SCOP1.75,

yielding an average classification accuracy of 75.3%. On the independent testing dataset curated from SCOP2.06, the classification accuracy is 73.0%. We compare our method with a top profile-profile alignment method - HHSearch on hard template-based and template-free modeling targets of CASP9-12 in terms of fold recognition accuracy. The accuracy of our method is 12.63%-26.32% higher than HHSearch on template-free modeling targets and 3.39%-17.09% higher on hard template-based modeling targets for top 1, 5, and 10 predicted folds. The hidden features extracted from sequence by our method is robust against sequence mutation, insertion, deletion and truncation, and can be used for other protein pattern recognition problems such as protein clustering, comparison and ranking. The web server of the method is available at: `http://iris.rnet.missouri.edu/DeepSF/`. The supplemental material can be found at: `https://doi.org/10.1093/bioinformatics/btx780`

3.2 Introduction

Protein folding reveals the evolutionary process between the protein amino acid sequence and its atomic tertiary structure [72]. Folds represent the main characteristics of protein structures, which describe the unique arrangement of secondary structure elements in the infinite conformation space [73, 74]. Several fold classification databases such as SCOP [74], CATH [75], FSSP [76], ECOD [77] have been developed to summarize the structural relationship between proteins. With the substantial investment in protein structure determination in the past decades, the number of experimentally determined protein structures has substantially increased to more than 100,000 in the Protein Data Bank (PDB) [2, 74]. However, due to the conservation of protein structures, the number of unique folds has been rather stable. For example, the SCOP 1.75 curated in 2009 has 1,195 unique folds, whereas SCOP 2.06 only has 26 more folds identified from the recent PDB [78]. Generally, determining the folds of a protein can be accomplished by comparing its structure with those of other proteins whose folds are known. However, because the structures of most (>99%) proteins are not known, the development of sequence-based computational fold detection method is necessary and essential to automatically assign proteins into fold. And identifying protein homologs sharing the same fold is a crucial step for computational protein structure predictions [79, 80] and protein function prediction [81].

Sequence-based methods for protein fold recognition can be summarized into two categories: (1) sequence alignment methods and (2) machine learning methods. The sequence alignment methods [82, 83] align the sequence of a target protein against the sequences of template proteins whose folds are known to generate alignment scores. If the score between a target and a template is significantly higher than that of two random sequences, the fold of the template is considered to be the fold of the target. In order to improve the sensitivity of detecting remote homologous sequences that share the same fold, sequence alignment methods were extended to align the profiles of two proteins. Profile-sequence alignment methods [38] and profile-profile alignment methods based hidden Markov model (HMM) [80] or Markov random fields (MRFs) [84] are more sensitive in recognize proteins that have the same fold, but little sequence similarity, than sequence-sequence alignment methods. Despite the success, the sequence alignment methods are essentially an indirect fold recognition approach that transfers the fold of the nearest sequence neighbors to a target protein, which cannot explain the sequence-structure relationship of the protein.

Machine learning methods have been developed to directly classify proteins into different fold categories [85, 86, 87, 88]. Multi-layer perception and support vector machine have been used to construct a single classifier to recognize fold pattern in an early work [85]. Ensemble classifiers were proposed to improve fold recognition [89]. In order to better use sequence features, kernel-based learning was designed to classify protein folds [86]. A recent ensemble-based method combined template-based search and support vector machine classification to recognize protein folds [90]. However, because traditional machine learning methods cannot classify data into a large number of categories (e.g., thousands of folds), these methods can only classify proteins into a small number (e.g., dozens) of pre-selected fold categories, which cannot be generally applied to predict the fold of an arbitrary protein and therefore is not practically useful for protein structure prediction. To work around the problem, another kind of machine learning methods [91, 92, 79] converts a multi-fold classification problem into a binary classification problem to predict if a target protein and a template protein share the same fold based on their pairwise similarity features, which is still an indirect approach that cannot directly explain how a protein sequence is mapped to one of thousands of folds in the fold space.

In this work, we utilize the enormous learning power of deep learning to directly classify any protein into one of 1,195 known folds. Deep learning techniques have achieved significant success in computer vision, speech recognition and natural language processing [93, 57]. The application of deep learning in bioinformatics has also gained the traction since 2012. Deep belief networks [94] were developed to predict protein residue-residue contacts. Recently a deep residual convolutional neural network was designed to further improve the accuracy of contact prediction [9]. Deep learning methods have also been applied to predict protein secondary structures [45, 54] and identify protein pairs that have the same fold [79, 84].

Here, we design a one-dimensional (1D) deep convolution neural network method (DeepSF) to classify proteins of variable-length into all 1,195 known folds defined in SCOP 1.75 database. DeepSF can directly extract hidden features from any protein sequence of any length through convolution transformation, and then classify it into one of thousands of folds accurately. The method is the first method that can map all protein sequences in the sequence space directly into all the folds in the fold space without relying on pairwise sequence comparison (alignment). The hidden fold-related features generated from sequences can be used to measure the similarity between proteins, cluster proteins, and select template proteins for tertiary structure prediction.

We rigorously evaluated our method on three test datasets: new proteins in SCOP 2.06 database, template-based targets in the past CASP experiments, and template-free targets in the past CASP experiments. Our method (DeepSF) is more sensitive than a state-of-the-art profile-profile alignment method - HHSearch in predicting the fold of a protein, and it is also much faster than HHSearch because it directly classifies a protein into folds without searching a template database. We also demonstrate that the hidden features extracted from protein sequences by DeepSF is robust against residue mutation, insertion, deletion and truncation. To generalize the application of our method, we also applied our deep convolutional neural network to classify proteins based on ECOD domain classification database [77], which focuses on distant evolutionary relationships between proteins.

3.3 Methods

3.3.1 Training, validation and test datasets

The main dataset that we used for training, validation and test was downloaded from the SCOP 1.75 genetic domain sequence subsets with less than 95% pairwise identity released in 2009. The protein sequences for each SCOP domain were cleaned according to the observed residues in the atomic structures [74]. The dataset contains 16,712 proteins covering 7 major structural classes with total 1,195 identified folds. The number of proteins in each fold is very uneven, with 5% (i.e., 61/1,195) folds each having $>$ 50 proteins, 26% (i.e., 314/1,195) folds each having 6 to 50 proteins, and 69% (820/1,195) each having $\leq$ 5 proteins, making it challenging to train a classifier accurately predicting all the folds, especially small folds with few protein sequences.

The proteins in all 1,195 folds have sequence length ranging from 9 to 1,419 (**Figure 3.1(a)**), and most of them have length in the range of 9 to 600 (**Figure 3.1(b)**). In order to remove the homologous sequence redundancy between test datasets and training datasets, we adopted two different strategies for homology reduction: three-level redundancy removal at fold/superfamily/family levels and sequence identity reduction. The three-level redundancy removal started with fold-level reduction that split proteins into a fold-level training dataset and a fold-level test dataset based on superfamilies, i.e., no proteins from the same superfamily will be included in both training and test datasets. The fold-level training dataset was split into a superfamily-level training dataset and a superfamily-level test dataset based on families, i.e., no proteins from the same family existed in both the training and test datasets. Finally, the superfamily-level training dataset was split into a family-level training dataset and a family level test dataset by sampling 80% of proteins in the same family for training and using the remaining 20% for test. After the three-level reduction, the 80% of proteins sampled from the fold-level, superfamily-level, and family-level test datasets, respectively, were combined into one test dataset. The remaining 20 percent of proteins from the fold-level, superfamily-level, and family-level test datasets were combined a validation dataset. We further removed the proteins in the validation dataset whose E-value of sequence similarity with proteins in the training dataset is less than ”1e-4”. More detailed description about three-level homology removal and how to tune hyper parameters on the validation dataset can found in Section 1.1 in the supplemental document. The distribution of E-value of best hits for proteins in the validation and test datasets in terms of family, superfamily and fold level is shown in Figure S7 in the supplemental document (`https://doi.org/10.1093/bioinformatics/btx780`). The three-level test datasets can validate the performance of the method at fold, superfamily, and family level on SCOP 1.75 database, respectively.

In order to validate the performance on two independent datasets: SCOP 2.06 and

CASP dataset, the SCOP 1.75 dataset with less than or equal to 95% sequence identity was split into a training dataset and a validation set according 8/2 ratio for each fold. The validation dataset was further filtered to at most 70%, 40%, 25% pairwise similarity with the training dataset according to the sequence identity reduction (see details for sequence similarity reduction in Section 1.2 in the supplemental document).

Independent SCOP 2.06 test dataset

In order to independently test the performance of our method, we collected the protein sequences in the latest SCOP 2.06 [78], but not in SCOP 1.75. The sequences with similarity greater than 40% with SCOP 1.75 dataset were further removed. And the remaining proteins were filtered to less than or equal to 25% pairwise similarity with e-value cutoff "1e-4" by CD-Hit suite [64]. The parameter setting for CD-HIT is described in Section 8.1 in the supplementary document. Finally, this independent SCOP test dataset contains 2,533 domains, covering 550 folds, which were split into three sub test datasets (37 proteins in the fold-level test dataset, 1,754 in the superfamily level test dataset, and 742 in the family-level test dataset).

Independent CASP test dataset

Besides classifying the proteins with known folds in the SCOP, we tested our methods on a protein dataset consisting of template-free and template-based targets used in the 9^{th}, 10^{th}, 11^{th}, and 12^{th} Critical Assessments of Structure Prediction (CASP) experiments from 2010 to 2016 [95, 96]. These are new proteins available after SCOP 1.75 was created in 2009. The complete CASP dataset contains 431 domains. The sequences in the CASP dataset with sequence identity $>$ 10% against the SCOP training dataset are removed. To assign the folds to these CASP targets, we compare each CASP target against all domains in SCOP 1.75 using the structural similarity metric - TM-score [97]. Based on the evaluation of domains from each fold, referred

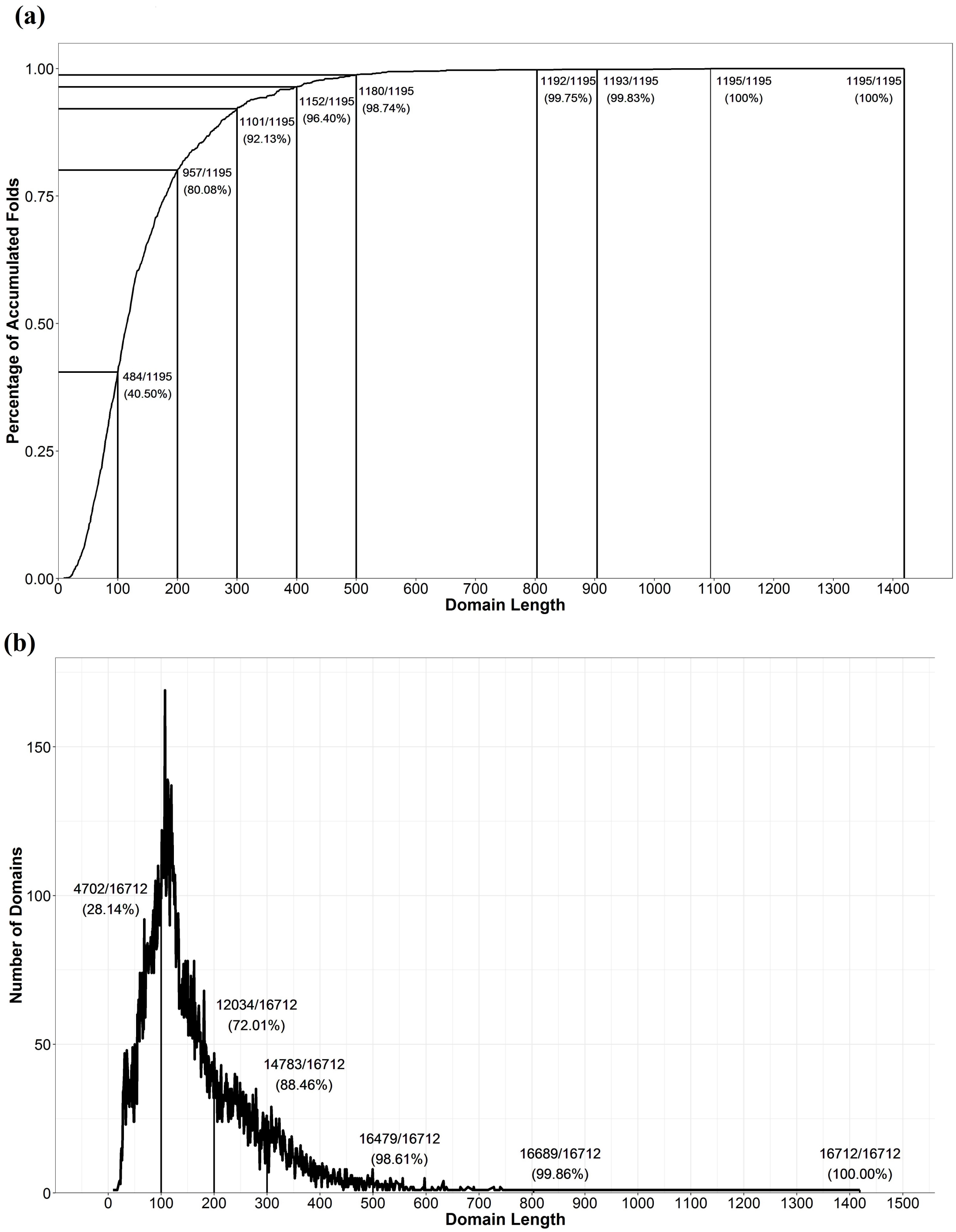


Figure 3.1: (a) The percentage of accumulated folds against length of proteins in the SCOP 1.75 dataset. In this plot, all the proteins with length less than 1,419 contains all 1,195 folds. (b) The distribution of the number of domains versus length of proteins in the SCOP 1.75 dataset. The proteins in SCOP 1.75 dataset with sequence similarity at most 95% have sequence length ranging from 9 to 1,419.

to supplemental Section 2 (`https://doi.org/10.1093/bioinformatics/btx780`), if a CASP target has TM-score above 0.5 with a SCOP domain, with 0.67 percentage alignment and RMSD < 3.57, suggesting they have the same fold, the fold of the SCOP domain is transferred to the CASP target [98]. If the CASP target does not have the same fold with any SCOP domain, it is removed from the dataset. After preprocessing, the dataset has 184 protein targets with fold assignment, which include 95 template-free (FM) or seemly template-free (FM/TBM) targets and 88 template-based (TBM) targets, where the categories of targets were defined by CASP experiments [96].

3.3.2 Input feature generation and label assignment

We generated four kinds of input features representing the (1) sequence, (2) profile, (3) predicted secondary structure, and (4) predicted solvent accessibility of each protein. Each residue in a sequence is represented as a 20-dimension zero-one vector in which only the value at the residue index is marked as 1 and all others are marked as 0. The position-specific scoring matrix (PSSM) for each sequence is calculated by using PSI-BLAST to search the sequence against the 'nr90' database. The 20 numbers in the PSSM corresponding to each position in the protein sequence is used as features to represent the profile of amino acids at the position. We predicted 3-class secondary structure (Helix, Strand, Loop) and two-class solvent accessibility (Exposed, Buried) for each protein sequence using SCRATCH [16]. The secondary structure of each position is represented by 3 binary numbers with one of them as 1, indicating which secondary structure it is. Similarly, the solvent accessibility at each position is denoted by two binary numbers. In total, each position of a protein sequence is represented by a vector of 45 numbers. The whole protein is encoded by $L \times 45$ numbers. It is worth noting that these input features have been used in protein fold recognition. [86, 79, 90]. Each sequence is assigned to a pre-defined fold index in the range of $0 \sim 1{,}194$ denoting its fold according to SCOP 1.75 definition, which is the class label of the protein.

3.3.3 Deep convolutional neural network for fold classification

The architecture of the deep convolutional neural network for mapping protein sequences to folds (DeepSF) is shown in **Figure 3.2**. It contains 15 layers including input layer, 10 convolutional layers, one K-max pooling layer, one flattening layer, one fully-connected hidden layer and an output layer. The softmax function is applied to the nodes in the output layer to predict the probability of 1,195 folds. The input layer has L × 45 input numbers representing the positional information of a protein sequence of variable length L. Each of 10 filters in the first convolution layer is applied to the windows in the input layer to generate L × 1 hidden features (feature map) through the convolution operation, batch-normalization and non-linear transformation of its inputs with the rectified-linear unit (ReLU) activation function [57], resulting 10 × L hidden features. Different window sizes (i.e., filter size) in the 1D convolution layer are tested and finally two window sizes (6 and 10) are chosen, which are close to the average length of beta-sheet and alpha-helix in a protein. The hidden features generated by 10 filters with two window sizes (i.e., 10 × L × 2) in the first convolution layer are as input to be transformed by the second convolution layer in the same way. The depth of convolution layers is set to 10. Inspired by the work [99], the K-max pooling layer is added to transform the hidden features of variable length in the last convolution layer to the fixed number of features, where K is set to 30. That is the 30 highest values (30 most active features) of each L × 1 feature map generated by a filter with a window size are extracted and combined. The extracted features learned from both window sizes (i.e., 6, 10) are merged into one single vector consisting of 10 × 30 × 2 numbers, which is fed into a fully-connected hidden layer consisting of with 500 nodes. These nodes are fully connected to 1,195 nodes in the output layer to predict the probability of 1,195 folds. The node in the output layer uses the softmax activation function. To prevent the over-fitting, the dropout [69] technique is applied in the hidden layer (i.e., the 14^{th} layer in **Figure 3.2**).

3.3.4 Model training and validation

We trained the one-dimensional deep convolutional neural network (DeepSF) on variable-length sequences in 1,195 folds. Considering the proteins in the training dataset have very different length of up to 1,419 residues, we split the proteins into multiple mini-batches (bins) based on fixed-length interval (bin size). The proteins in the same bin have similar length in a specific range. The zero-padding is applied to the sequences whose length is smaller than the maximum length in the bin. All the mini-batches are trained for 100 epochs, and the proteins in each bin are used to train for a small number of epochs (i.e., 3 epochs for bin with size of 15) in order to avoid over-training on the proteins in a specific bin. We evaluated the performance of different bin sizes (see the Result section 3.4) to choose a good bin size. The DeepSF with different parameters is trained on the training dataset with less than or equal to 95% pairwise similarity, and is then evaluated on the validation sets with different sequence similarity levels (95%, 70%, 40%, 25%) or at three hierarchical levels (family/superfamily/fold) with the training dataset. The model with the best average accuracy on the validation datasets is selected as final model for further testing and evaluation. A video demonstrating how DeepSF learns to classify a protein into a correct fold during training is available `http://iris.rnet.missouri.edu/DeepSF/`.

3.3.5 Model evaluation and benchmarking

We tested our method on the two independent test datasets: SCOP 2.06 and CASP dataset (see Section 3.3.1). Since the number of proteins in different folds are extremely

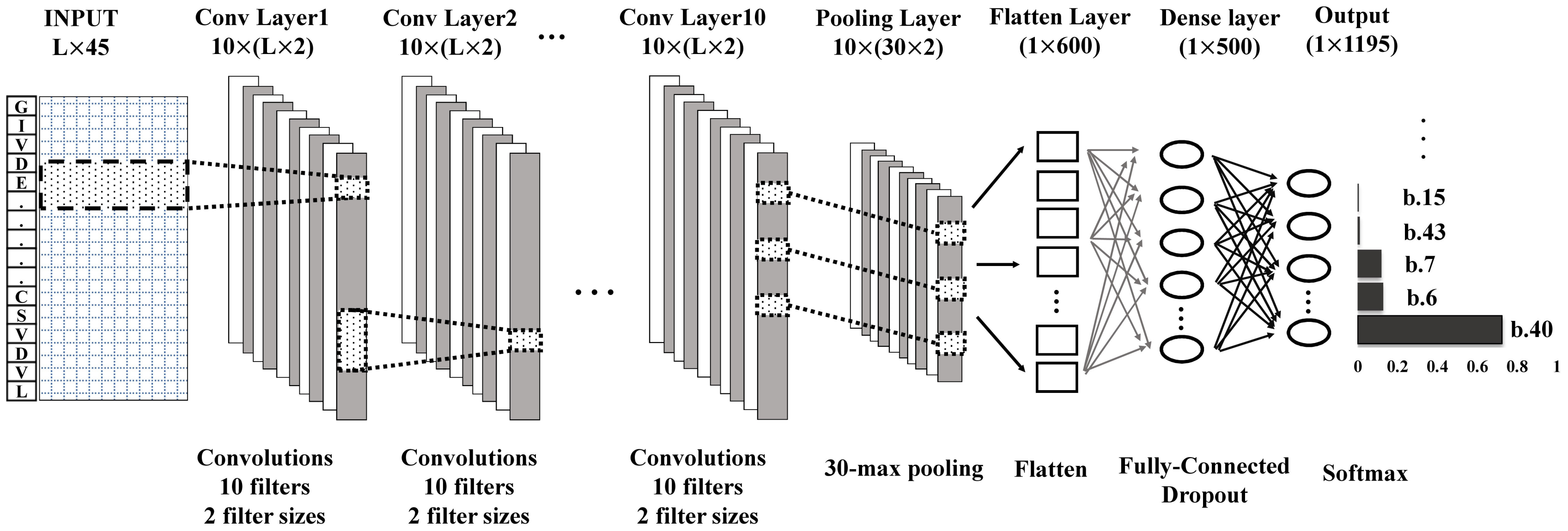


Figure 3.2: The architecture of 1D deep convolutional neural network for fold classification. The network accepts the features of proteins of variable sequence length (L) as input, which are transformed into hidden features by 10 hidden layers of convolutions. Each convolution layer applies 10 filters to the windows of previous layers to generate L hidden features. Two window sizes (6 and 10) are used. The 30 maximum values of hidden values of each filter of the 10^{th} convolution layer are selected by max pooling, which are joined together into one vector by flattening. The hidden features in this vector are fully connected to a hidden layer of 500 nodes, which are fully connected to 1,195 output nodes to predict the probability of each of 1,195 folds. The output node uses softmax function as activation function, whereas all the nodes in the other layers use rectified linear function max(x, f(x)) as activation function. The features in the convolution layers are normalized by batches.

unbalanced, we split the 1,195 folds into three groups based on the number of proteins within each fold (i.e., small, medium, large). A fold is defined as 'small' if the number of proteins in the fold is less than 5, 'medium' if the number of proteins is in the range between 6 and 50, and 'large' if the number of proteins is larger than 50. We evaluated DeepSF on the proteins of all folds and those of each category in the test dataset separately. We compared DeepSF with the baseline majority-assignment method, which assigns the most frequent folds to the test proteins. Moreover, we compared DeepSF with a state-of-the-art profile-profile alignment method - HHSearch and PSI-BLAST on the CASP dataset based on top1, top5, top10 predictions, respectively.

3.3.6 Hidden fold-related feature extraction and template ranking

The outputs of the 14^{th} layer of DeepSF (the hidden layer in fully connected layers) used to predict the folds can be considered as the hidden, fold-related features of an input protein, referred to as SF-Feature. The hidden features bridges between the protein sequence space and the protein fold space as the embedded word features connect a natural language sentence to its semantic meaning in natural language processing. Therefore, the hidden features extracted for proteins by DeepSF can be used to assess the similarity between proteins and can be used to rank template proteins for a target protein.

In our experiment, we evaluated the following four different distance (or similarity) metrics to measure the similarity between the fold-related features:

(1) Euclidean distance

$$Euclid - D : (Q, T) = \sqrt{\Sigma_{i=1}^{N}(Q_i - T_i)^2} \qquad (3.1)$$

(2) Manhattan distance:

$$Manh - D : (Q, T) = \Sigma_{i=1}^{N} |Q_i - T_i| \tag{3.2}$$

(3) Pearson's Correlation score:

$$Corr - D : (Q, T) = log(1 - Corr(Q, T)) \tag{3.3}$$

(4) KL-Divergence:

$$KL - D : (Q, T) = \Sigma_{i=1}^{N} (Q_i log \frac{Q_i}{T_i} + T_i log \frac{T_i}{Q_i}) \tag{3.4}$$

where Q, T is the SF-feature for query protein and template protein.

We randomly sampled 5 folds from the training dataset and sampled at most 100 proteins from the 5 folds to test the four metrics above. We use hierarchical clustering to cluster the proteins into 5 clusters, where the distance between any two proteins is calculated from their fold-related feature vectors by the four metrics, respectively. This process is repeated 1,000 times and the accuracy of clustering based on the four distance metrics are calculated and compared (see Results Section 3.4). To select the best template for a target protein, the fold-related features of the target protein is compared with those of the proteins in the fold that the target protein is predicted to belong to. The templates in the fold are ranked in terms of their distance with the target protein.

3.4 Results and discussion

3.4.1 Training and validation on SCOP 1.75 dataset

We trained the deep convolutional neural network (DeepSF) on SCOP 1.75 dataset in the mini-batch mode, where the proteins in each mini-batch (bin) have similar length. We evaluated the effects of different bin sizes: 500, 200, 50, 30, 15 and size ranging from 1 to 15. The classification accuracy on the validation dataset with different bin sizes for each epoch of training is shown in **Figure 3.3**. Bin size of 15 has the fastest convergence and highest accuracy on both training (see **Figure 3.3(a)**) and validation datasets (see **Figure 3.3(b)** and **Figure 3.4**), and therefore is chosen taking accuracy and running time into account. For the test dataset of SCOP 1.75, we evaluated the performance of DeepSF at family, superfamily and fold level against training datasets. As shown in **Table 3.1**, at the family level, DeepSF achieves the accuracy of 76.18% for top prediction, which is worse than a standard sequence alignment method - PSI-BLAST. At the superfamily level, for top 1 (or top 5) prediction, the accuracy of DeepSF is 50.71% (or 77.67%), which is much higher than 42.20% (or 51.40%) of PSI-BLAST. At the fold level, for top 1 (or top 5) prediction, the accuracy of DeepSF is 40.95% (or 70.47%), which is many times better than 5.60% (or 11.60%) of PSI-BLAST. It is worth noting that the accuracy of PSI-BLAST is calculated based on the top folds from the ranked templates. The results show that DeepSF recognizes folds much better than PSI-BLAST for hard cases when sequence identify is very low.

On the validation datasets whose redundancy is reduced to at most 95%, 70%, 40% and 25% sequence similarity with the training dataset, DeepSF achieves the accuracy of 80.4% (or 93.7%) for top 1 (or top 5) predictions at the 95% similarity level. The average accuracy on all the four validation datasets (95%/70%/40%/25%) is 75.3% (or 90.9%) for top 1 (or top 5) predictions. The detailed results on these validation datasets are reported in **Table 3.2**.

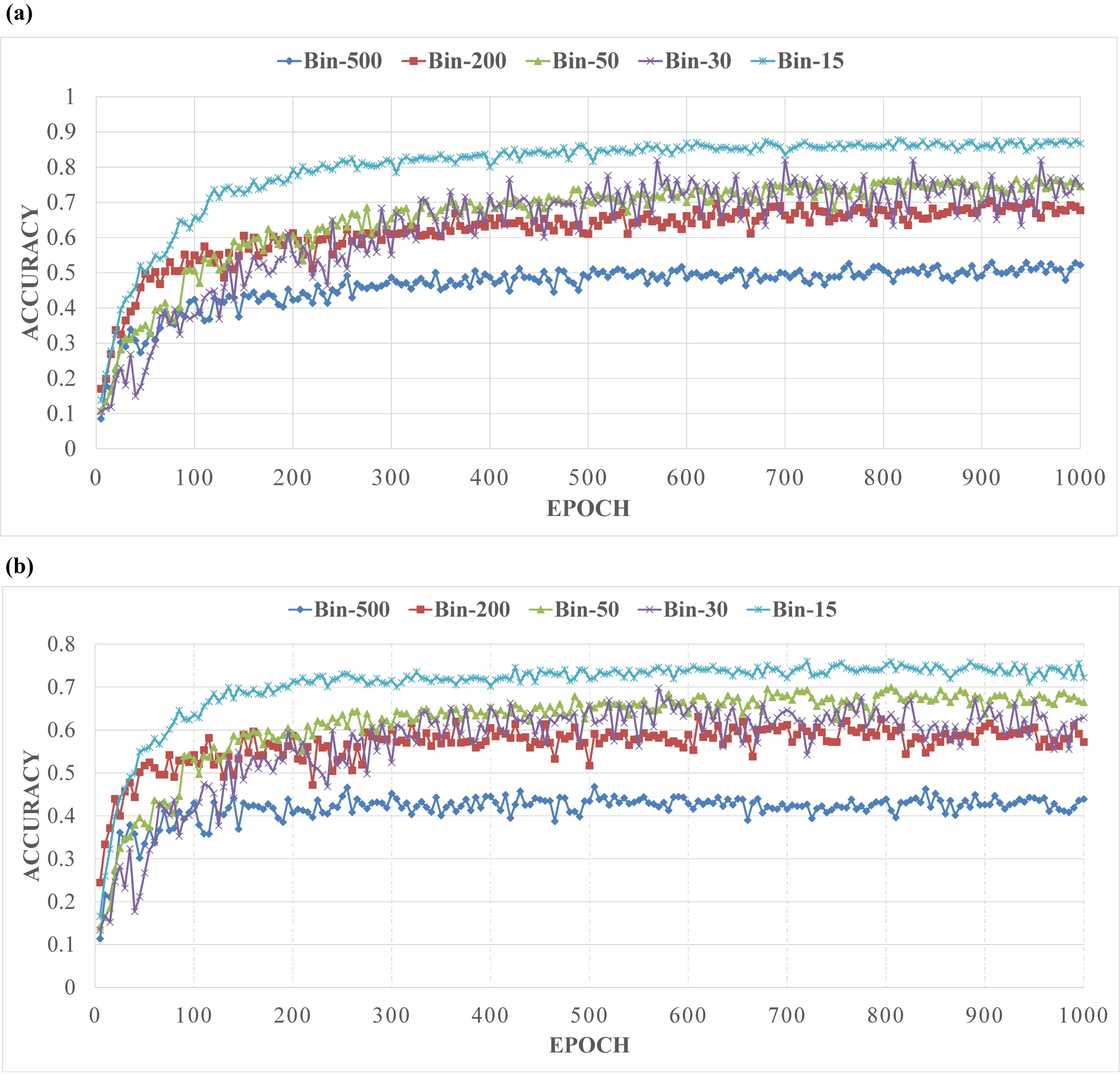


Figure 3.3: (a). The classification accuracy of training dataset against the number of training epochs for 5 different bin size. (b). The classification accuracy of validation dataset against the number of training epochs for 5 different bin size.

3.4.2 Performance on SCOP 2.06 dataset

We evaluated DeepSF on the independent SCOP 2.06 dataset, which contains 2,533 proteins belonging to 550 folds. 60 folds with 1,326 proteins are considered as "Large" fold, 249 folds with 898 proteins as "Medium" fold and 241 folds with 307 proteins as

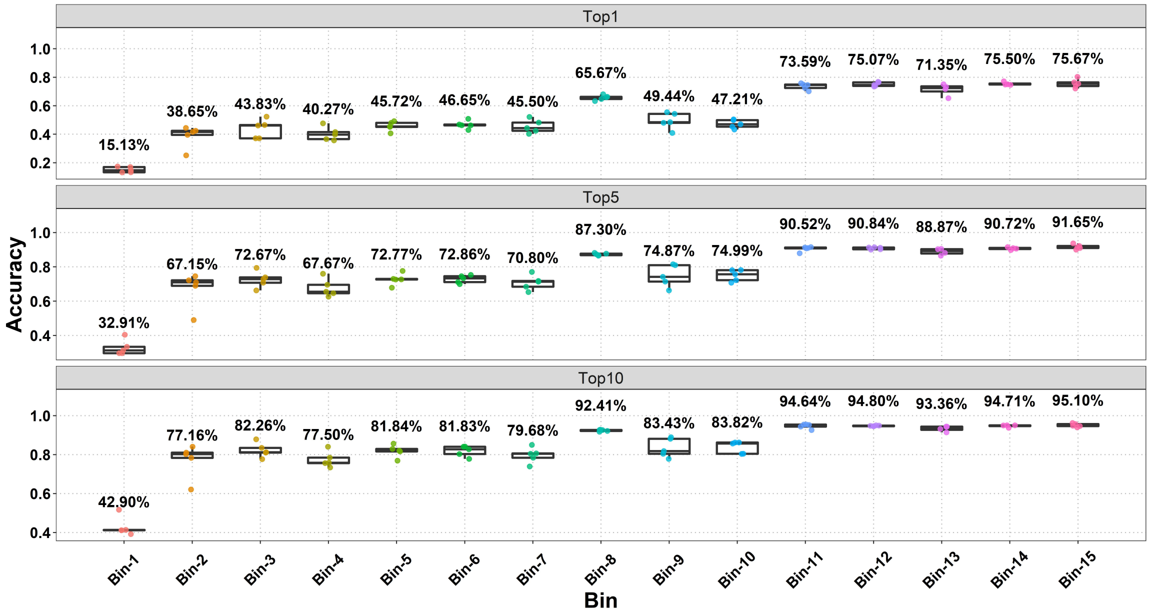


Figure 3.4: The effects of bin size between 1 and 15 on the model training. Accuracy was calculated based on the sequence identity reduction based dataset from SCOP 1.75. Training process was repeated and visualized as points. The averaged accuracy on the validation dataset based on each bin size was annotated.

Level	**Methods**	**Top1**	**Top5**	**Top10**
Family	DeepSF	76.18%	94.50%	97.56%
(1,272 proteins)	PSI-BLAST	96.80%	97.40%	97.60%
Superfamily	DeepSF	50.71%	77.67%	77.67%
(1,254 proteins)	PSI-BLAST	42.20%	51.40%	54.60%
Fold	DeepSF	40.95%	70.47%	82.45%
(718 proteins)	PSI-BLAST	5.60%	11.60%	16.20%

Table 3.1: The prediction accuracy at family/superfamily/fold levels for top 1, top 5, and top 10 predictions of DeepSF and PSI-BLAST, on SCOP 1.75 test dataset

	ID < 95%	**ID < 70%**	**ID < 40%**	**ID < 25%**	**Average**
Top 1	**80.40%**	**78.20%**	**75.80%**	**66.90%**	**75.30%**
Top 5	93.70%	92.40%	90.00%	87.60%	90.90%
Top 10	96.20%	95.40%	93.60%	92.10%	94.30%

Table 3.2: The prediction accuracy on four validation sets with different sequence similarity to training dataset for top 1, top 5, and top 10 predictions.

"Small" fold. The classification accuracy of DeepSF on all the folds and each kind of fold is reported in **Table 3.3**. The accuracy on the entire dataset is 73.0% and 90.25% for top 1 prediction and top 5 predictions, respectively. The model also achieves accuracy of 79.64%, 74.16% and 67.93% for top 1 prediction on "Large", "Medium", and "Small" folds, respectively. The higher accuracy on larger folds suggests that more training data in a fold leads to the better prediction accuracy. The classification accuracy of DeepSF on SCOP 2.06 dataset at family, superfamily and fold level against training dataset is reported in **Table 3.4**.

DeepSF	**Top1**	**Top5**	**Top10**
SCOP2.06 dataset	73.00%	90.25%	94.51%
"Large" folds	79.64%	94.87%	97.81%
"Medium" folds	74.16%	75.61%	76.06%
"Small" folds	67.93%	86.86%	94.74%

Table 3.3: The accuracy of DeepSF on SCOP 2.06 dataset and its subsets

3.4.3 Performance on CASP dataset

We evaluated our method on the CASP dataset, including 95 template-free proteins and 88 template-based proteins. We compared our method with the two widely used alignment methods (HHSearch and PSI-BLAST). Our method predicts the fold for each CASP target from its sequence directly. HHSearch and PSI-BLAST search each

Type	**Methods**	**Top1**	**Top5**	**Top10**
Family	DeepSF	75.87%	91.77%	95.14%
(742 proteins)	PSI-BLAST	82.20%	84.50%	85.30%
Superfamily	DeepSF	72.23%	90.08%	94.70%
(1,754 proteins)	PSI-BLAST	86.90%	88.40%	89.30%
Fold	DeepSF	51.35%	67.57%	72.97%
(37 proteins)	PSI-BLAST	18.90%	35.10%	35.10%

Table 3.4: The prediction accuracy at family/superfamily/fold level for top 1, top 5, and top 10 predictions, on SCOP 2.06 test dataset.

CASP target against the proteins in the training dataset to find the homologs to recognize its fold, where the accuracy of PSI-BLAST/HHSearch is calculated based on the top ranked folds from the identified templates.

As shown in the **Table 3.5** and **Table 3.6**, DeepSF achieved better accuracy on both template-based targets and template-free targets than HHSearch, PSI-BLAST in all situations. On the template-based targets that have little similarity with training proteins, the accuracy of DeepSF for top 1, 5, 10 predictions are 46.59%, 73.86%, 84.09% (see **Table 3.5**), which is 3.39%, 12.46%, 17.09% higher than HHSearch. And interestingly, the consensus ranking of HHSearch and DeepSF (Cons_HH_DeepSF) is better than both DeepSF and HHSearch, particularly for top 1 prediction, suggesting that the two methods are complementary on template-based targets. Because CASP targets has very low sequence similarity (<10%) with the training dataset, which is difficult for profile-sequence alignment methods to recognize, PSI-BLAST has the lowest prediction accuracy. On the hardest template-free targets that presumably have no sequence similarity with the training dataset, the accuracy of DeepSF for top 1, 5 and 10 predictions are 24.21%, 51.58%, and 70.53% (see **Table 3.6**), 12.63%, 16.84% and 26.32% higher than HHSearch that performs better than PSI-BLAST. The consensus (Cons_HH_DeepSF) of DeepSF and HHSearch is only slightly better than DeepSF, which is different from its effect on template-based modeling targets.

Method	**Top1**	**Top5**	**Top10**
DeepSF	46.59%	73.86%	84.09%
HHSearch	43.20%	61.40%	67.00%
Cons_HH_DeepSF	59.10%	77.30%	85.20%
PSI-BLAST	15.90%	31.80%	47.70%

Table 3.5: The performance of the methods on 88 template-based proteins in the CASP dataset

Method	**Top1**	**Top5**	**Top10**
DeepSF	24.21%	51.58%	70.53%
HHSearch	11.58%	34.74%	44.21%
Cons_HH_DeepSF	23.16%	56.84%	70.53%
PSI-BLAST	8.42%	15.79%	32.63%

Table 3.6: The performance of the methods on 95 template-free proteins in the CASP dataset

3.4.4 Evaluation of four distance metrics for comparing fold-related hidden features

We evaluated the four distance metrics by using hierarchical clustering to cluster proteins with known folds based on their hidden fold-related features (see Method Section 3.3.6). The boxplot in **Figure 3.5(a)** shows the clustering accuracy of 4 different distance metrics. While Euclid-D, Manh-D and Corr-D achieve accuracy of 86.3%, 80.4%, and 88.0%, KL-D performs the best with accuracy of 89.3%. **Figure 3.5(b)** shows an example that using KL-D as distance metric to cluster the fold-level features of proteins in five SCOP2.06 folds that are randomly sampled. The proteins are perfectly clustered into 5 groups with the same folds. The visualized heat map

(**Figure 3.5(b)**) shows that proteins in the same cluster (fold) has the similar hidden feature values.

3.4.5 Fold-classification assisted protein structure prediction

Since applying a distance metric such as KL-D to the fold-related hidden features of two proteins can be used to measure their structural similarity, we explored the possibility of using it to rank template proteins for a target protein to assist tertiary structure prediction. Using the DeepSF model, we can generate fold-related features (SF-features) for any protein in a template protein database. In our experiment, we use DeepSF to generate SF-features for all the proteins in the training dataset as the template database. Given a target protein, we first extracted its SF-features and predicted the top 5 folds for it. We selected top 5 folds because top 5 predictions generally provided the high accuracy of fold prediction. Then we collected the template proteins that belong to the predicted top 5 folds and compare their SF-features with that of the target protein using KL-D metric. The templates are then ranked by KL-D scores from smallest to largest, and the top ranked 10 templates are selected to build the protein structures for the target proteins [100]. This method contrasts with the approach of HHSearch, where the target sequence is searched against the template database, and the top ranked 10 templates with smallest e-value are selected as candidate templates for protein structure prediction.

After the templates are detected by DeepSF or HHSearch, the sequence alignment between the target protein and each template are generated using HHalign [80]. Each alignment and its corresponding template structure are fed into Modeller [5] to build the tertiary structures. The predicted structural model with highest TMscore among all the models generated by top templates is selected for comparison. The quality of best predicted models from DeepSF and HHSearch is evaluated against the native structure in terms of TM-score and RMSD [97].

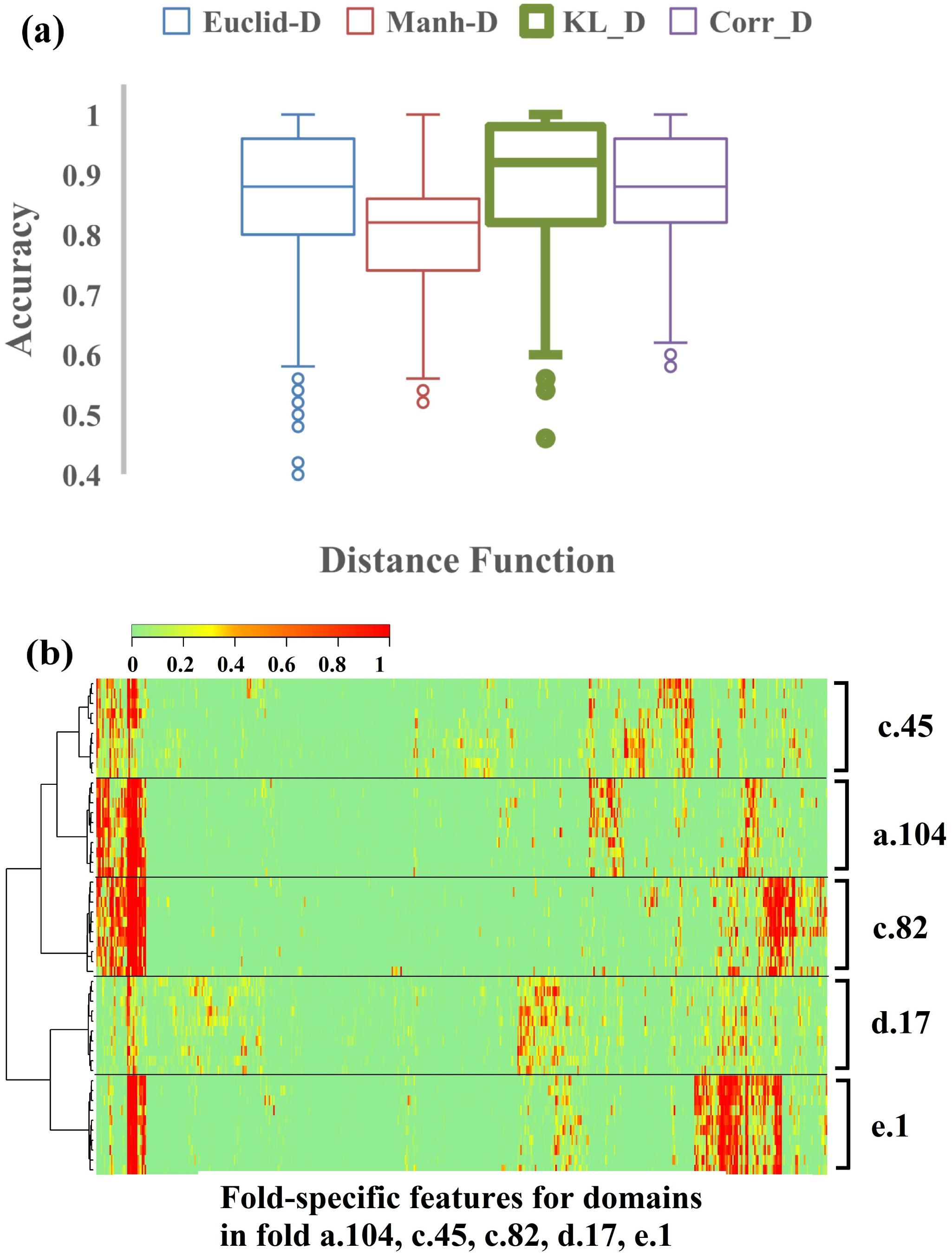


Figure 3.5: (a) The accuracy of 4 distance metrics in clustering proteins based on fold-related features. The clustering accuracy is average over 1000 clustering processes. (b) A hierarchical clustering of proteins from 5 folds in the SCOP 2.06 dataset using KL-D as metric. Each row in the heat map visualizes a vector of fold-related hidden features of a protein. The feature vectors of the proteins of the same fold are similar and clustered into the same group.

Here, we mainly evaluated template ranking and protein structure prediction on the 95 template-free CASP targets assuming that our method is more useful for detecting structural similarity for hard targets without sequence similarity with known templates. **Table 3.7** reports the average, min, max and standard deviation (std) of TMscore of the best models predicted for 95 template-free targets by DeepSF and HHSearch. DeepSF achieved a higher average TMscore (0.27) than that (0.25) of HHSearch. And the p-value of the difference using Wilcoxon paired test is 0.019.

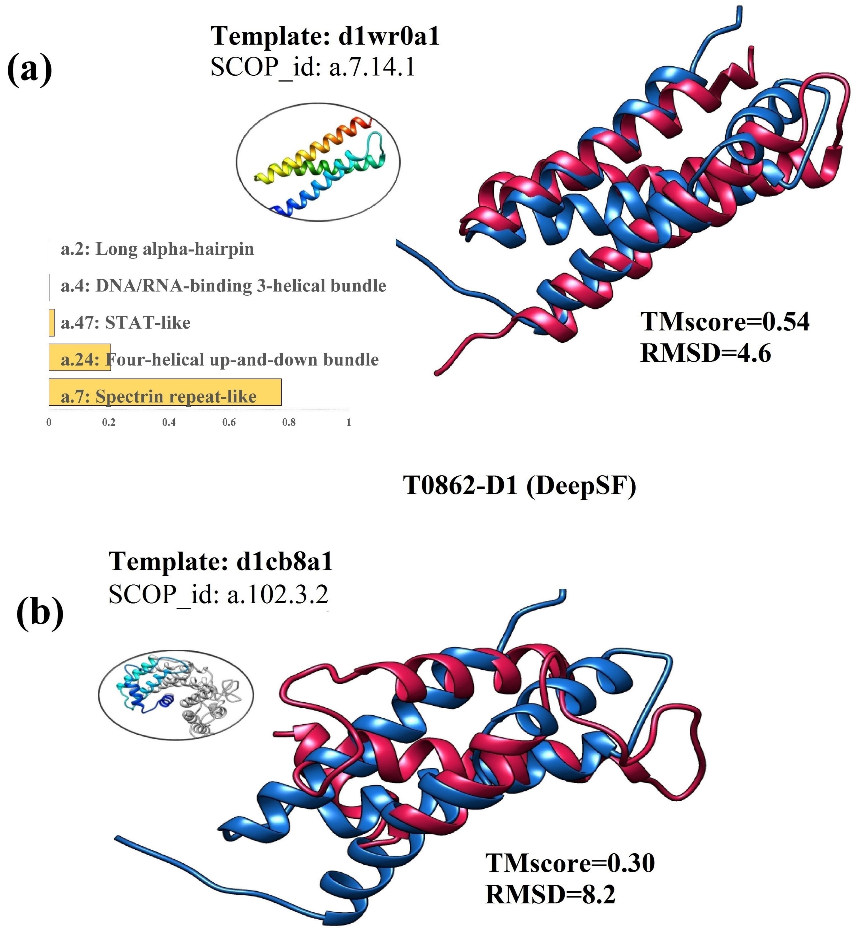
Figure 3.6(b) shows an example on which DeepSF performed well. T0862-D1 is a template-free target in CASP 12, which contains multiple helices. DeepSF firstly classifies T0862-D1 into fold 'a.7' with probability 0.77 which is a 3-helix bundle. And among the top 10 ranked templates with smallest KL-D score in the fold 'a.7', the domain 'd1wr0a1' (SCOP id: a.7.14.1) was used to generate the best structural model with TMscore = 0.54 and RMSD = 4.6 Angstrom. In contrast, among the top 10 predicted structural models from HHSearch, the best model was constructed from a segment (residues 5-93) of a large template 'd1cb8a1' (SCOP id: a.102.3.2), which has TMscore of 0.30 and RMSD of 8.2.

Methods	**TM-score**			
	Min	Max	Mean	Std
DeepSF	0.15	0.54	0.27	0.07
HHSearch	0.11	0.52	0.25	0.08

Table 3.7: Accuracy of protein structure predictions on 95 template-free targets.

3.4.6 Robustness of fold-related features against sequence mutation, insertion, deletion and truncation

In the evolutionary process of proteins, amino acid insertion, deletion or mutations mostly modifies protein sequences without changing the structural fold. Protein



T0862-D1 (DeepSF)

T0862-D1 (HHSearch)

Figure 3.6: Tertiary structure prediction for CASP12 target T0862-D1 based on templates identified by DeepSF and HHSearch. (a) DeepSF predictions: a top template, five predicted folds and the supposition between the best model and the template structure; (b) HHSearch predictions: top template, and superposition of the best model and the template structure.

truncation that shortens the protein sequences at either N-terminal or C-terminal sometimes still retains the structural fold [101]. A good method of extracting fold-related features from sequences should capture the consistent patterns despite of the evolutionary changes. Therefore, we simulated these four residue changes to check if the fold-related features extract from protein sequences by DeepSF are robust against mutation, insertion, deletion and even truncation. To analyze the effects of mutation, insertion, and deletion, we selected some proteins that have 100 residues, and randomly selected the positions for insertion, deletion, or substitution with one or more residues randomly sampled from 20 standard amino acids. And at most 20 residues in total are deleted from or inserted into sequences. Each change was repeated 50 times, and the exactly same sequences were removed after sampling. For example, for domain 'd1lk3h2' we generated 44 sequences with at least one residue deleted, and 44 sequences with at least one residue insertion, and 18 sequences with at least one residue mutation. The SF-Features for these mutated sequences are generated and compared to the SF-Feature of the original wild-type sequence. We also randomly sampled 500 sequences with length in the range of 80 to 120 residues from the SCOP 1.75 dataset as control, and compare their SF-features with those of the original sequence. The distribution of KL-D divergences between the SF features of these sequences and the original sequence are shown in **Figure 3.7**. The divergence of the sequences with mutations, insertions, and deletions from the original sequence is much smaller than that of random sequences. The p-value of difference according to Wilcoxon rank sum test is $< 2.2\text{e-}16$. The same analysis is applied to the other two proteins: 'd1foka3' and 'd1ipaa2', and the same phenomena has been observed (see **Figure 3.8**). The results suggest that the feature extraction of DeepSF is robust against the perturbation of sequences.

For the truncation analysis, we simulated residue truncations on C-terminus of 4,188 proteins in the SCOP 2.06 datasets (identity 40% against SCOP1.75) by letting

DeepSF read each protein's sequence from N-terminal to C-terminal to predict its fold. DeepSF needs to read 67.1% of the original sequences from N- to C-terminal on average in order to predict the same fold as using the entire sequences. This may suggest that the feature extraction is robust against the truncation of residues at C-terminal. A video demonstrating how DeepSF reads a protein sequence from N- to C-terminal to predict fold is available at `http://iris.rnet.missouri.edu/DeepSF/`.

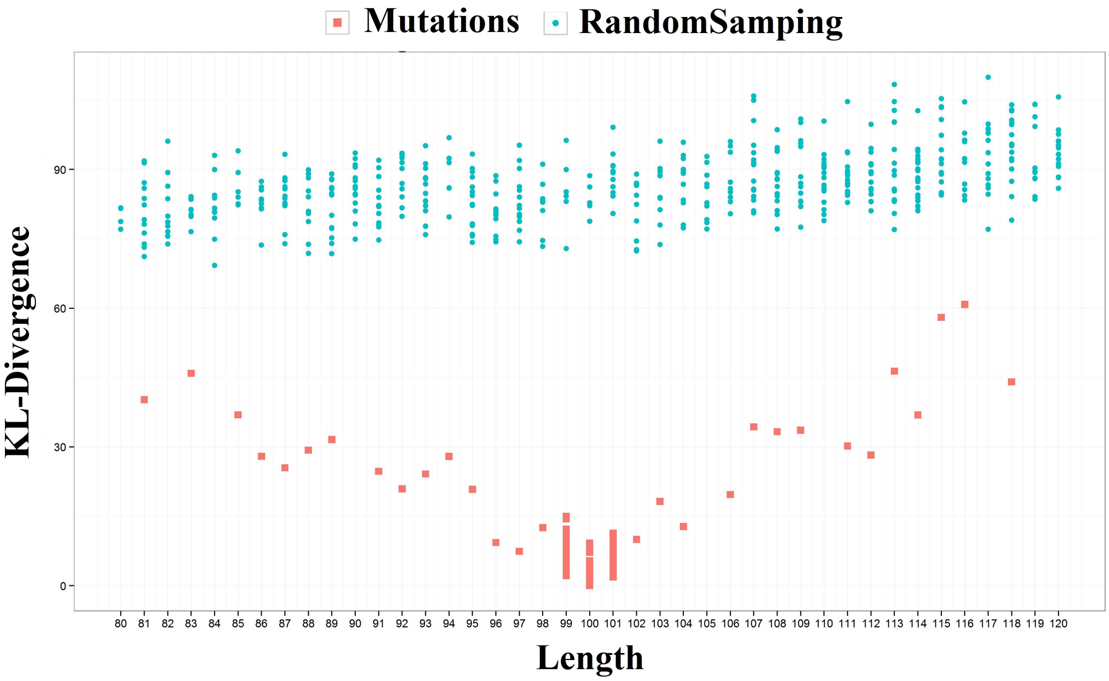


Figure 3.7: The KL-D divergences of fold-related features of 106 modified sequences of protein 'd1lk3h2' from the wild-type sequence (red dots) and those of 500 random sequences from the wild-type sequence (blue dots).

3.4.7 Generalization of deep convolutional neural network for family classification on SCOP database and fold classification on ECOD database

We generalized our method to the family level classification involving 3,901 families in the SCOP1.75 database. On the test dataset, the prediction was 61.21% (or

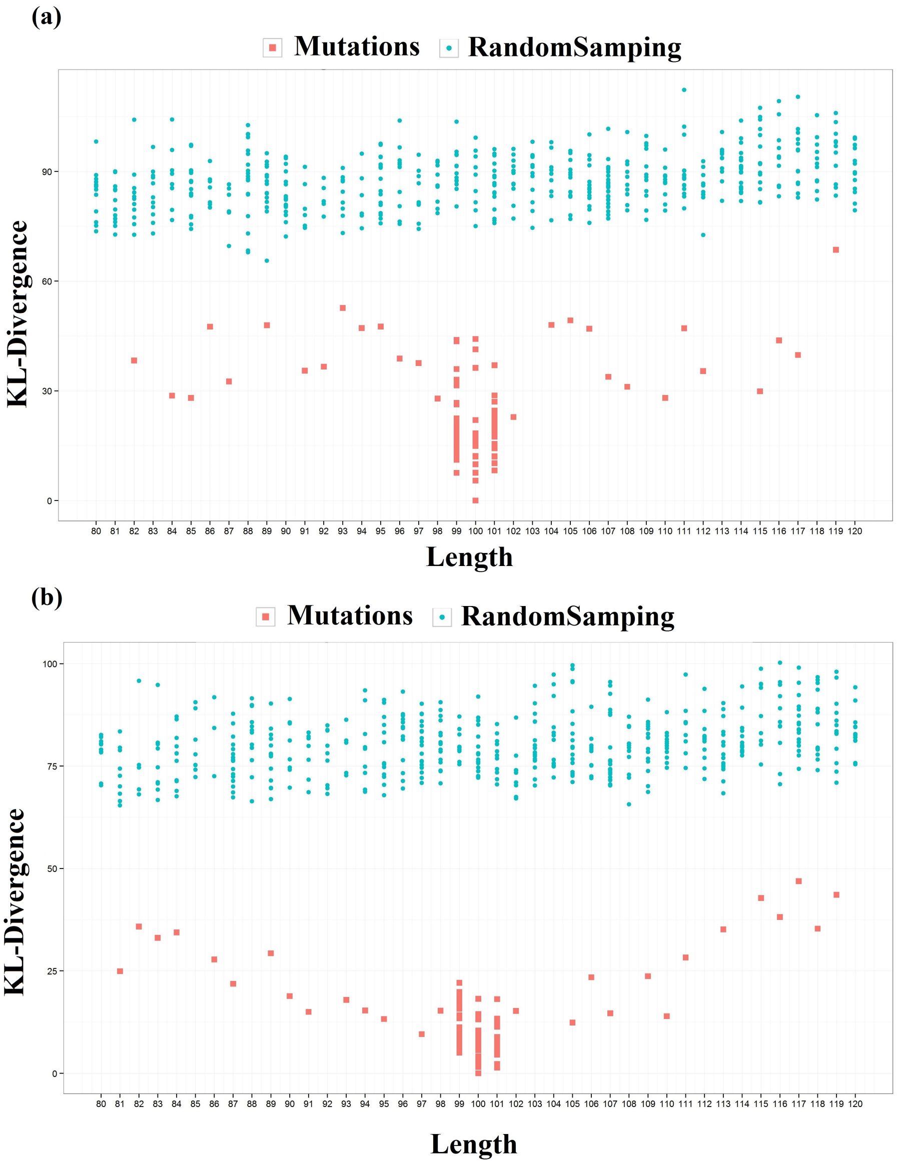


Figure 3.8: (a). The KL-D divergences of fold-related features of 102 modified sequences of protein 'd1foka3' from the wild-type sequence (red dots) and those of 500 random sequences from the wild-type sequence (blue dots). We generated 46 sequences with at least one residue deleted, and 40 sequences with at least one residue insertion, and 16 sequences with at least one residue mutation. (b). The KL-D divergences of fold-related features of 106 modified sequences of protein 'd1ipaa2' from the wild-type sequence (red dots) and those of 500 random sequences from the wild-type sequence (blue dots). We generated 45 sequences with at least one residue deleted, and 41 sequences with at least one residue insertion, and 20 sequences with at least one residue mutation.

79.42%) for top1 (or top 5) prediction. Detailed results are described in the Section 3 in the supplementary document (https://doi.org/10.1093/bioinformatics/btx780). Moreover, we trained our method on the ECOD database [77], which is a hierarchical domain classification data-base based on the distant evolutionary relationships between proteins. We designed two architectures to classify 2,186 possible homologous groups (sharing similar structure but lack a convincing argument for homology) with an accuracy of 50.95% (or 78.23%) for top 1 (or top 5) prediction and 3,459 homologous groups with an accuracy of 47.46% (or 71.52%) for top 1 (or top 5) prediction. The detailed analysis of the results is reported in Section 4 in the supplementary document (https://doi.org/10.1093/bioinformatics/btx780).

3.4.8 Feature importance analysis for fold classification

In this study, four kinds of sequence and structure features were generated to represent the protein sequence. It is worth analyzing the importance of the four features to the fold classification. The input features of proteins from total 15 different feature combination sets were fed into 1D-convolutional neural network for fold classification, and the classification accuracy were evaluated. The results are summarized in the **Figure 3.9**. Secondary structure information make most significant contribution to the fold classification with at least 6.48% higher accuracy in top 1 predictions compared to the rest three features. And including all 4 features will lead to best performance. Due to the significant effect of secondary structure features, we also analyzed how different quality of predicted secondary structure will influence the fold prediction, which is useful in real practice. In this study, we generated predicted secondary structure by SCRATCH [16], DeepCNF [102], DNSS [45], and PSIPRED [71], which were used for fold classification in CASP dataset. The quality of predicted secondary structures (Q3, SOV) were calculated based on that in the native structure. More details are described in the section S7 in the supplementary file (https://doi.

org/10.1093/bioinformatics/btx780).

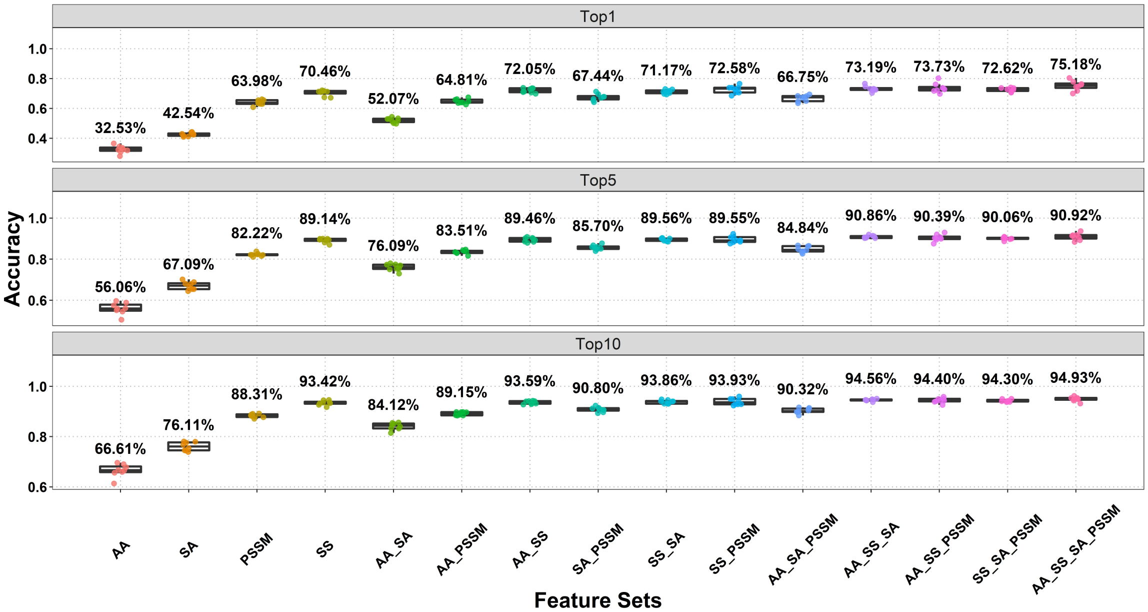


Figure 3.9: The feature importance analysis on fold classification. Accuracy was calculated based on the sequence identity reduction based dataset from SCOP 1.75. Training process was repeated and visualized as points. The averaged accuracy on the validation dataset based on each feature set was annotated.

3.5 Conclusion

We presented a deep convolution neural network to directly classify a protein sequence into one of all 1,195 folds defined in SCOP 1.75. To our knowledge, this is the first system that can directly classify proteins from the sequence space to the entire fold space rather accurately without using sequence comparison. Our method can automatically extract a set of fold-related hidden features from protein sequence of any length by deep convolution, which is different from previous machine learning methods relying on a window of fixed size or human expertise for feature extraction. The automatically extracted features are robust against sequence perturbation and can be used for various protein data analysis such as protein comparison, clustering, template ranking and structure prediction. And on the independent test datasets, our method is more accurate in recognizing folds of target proteins that have little or no sequence similarity with the proteins having known structures than widely used profile-profile alignment methods. Moreover, our method of directly assigning a protein sequence to a fold is not only complementary with traditional sequence-alignment methods based on pairwise comparison, but also provides a new way to study the protein sequence-structure relationship.

Chapter 4

Deep convolutional neural networks for predicting the quality of single protein structural models

4.1 Abstract

Predicting the global quality and local (residual-specific) quality of a single protein structural model is important for protein structure prediction and application. In this work, we developed a deep one-dimensional convolutional neural network (1DCNN) that predicts the absolute local quality of a single protein model as well as two 1DCNNs to predict both local and global quality simultaneously through a novel multi-task learning framework. The networks accept sequential and structural features (i.e., amino acid sequence, agreement of secondary structure and solvent accessibilities, residual disorder properties and Rosetta energies) of a protein model of any size as input to predict its quality, which is different from existing methods using a fixed number of hand-crafted features as input. Our three methods (InteractQA-net, JointQA-net and LocalQA-net) were trained on the structural models of the single-domain protein targets of CASP8, 9, 10 and evaluated on the models of CASP11 and CASP12 targets.

The results show that the performance of our deep learning methods is comparable to the state-of-the-art quality assessment methods. Our study also demonstrates that combining local and global quality predictions together improves the global quality prediction accuracy. The source code and executable of our methods are available at: `https://github.com/multicom-toolbox/CNNQA`.

4.2 Introduction

In the past few decades, protein structure prediction had achieved significant progress on both template-based modeling and template-free modeling [18, 103, 104, 105, 23, 106, 3]. As a quality control step of modeling, protein model quality assessment (QA) plays an important role in selecting most accurate models among a massive number of decoys generated by protein structure modeling methods. There are two kinds of model quality assessment methods: local quality assessment [107, 108, 109, 110, 111] and global quality assessment [13, 31, 32, 112, 113, 114, 115, 116, 109, 117, 118]. Local QA methods attempt to predict the spatial deviation of each residue in a model from the native structure (e.g., the absolute distance between the position of Ca atom of a residue in a model and that in the native structure), while global QA methods aim to predict the overall similarity (e.g., GDT-TS score [119]) between a model and its native structure. One kind of QA methods require a pool of models as input, which are often called consensus (or multi-model) methods [115, 120, 121, 122]. Consensus QA methods evaluate a protein model by comparing it against the other models in the pool and calculating the average structural similarity as an indicator of the quality. Another kind of QA method only takes a single model as input to predict its quality, which are called single-model QA methods. These methods utilize the sequence and structural information of a single model itself to assess its quality. Consensus QA methods usually achieve good performance if a significant portion of models in the model pool are of good quality. However, it tends to fail if most models are of poor quality and is time-consuming if the size of the model pool is large. In contrast, the performance of single-model QA methods can be more consistent and more independent of the distribution of model quality in a pool because it predicts the quality of a model using only the information about itself. This is particularly useful if there are very few good models in a large model pool, which often happens in template-free protein structure prediction.

Recent top-ranked single-model QA methods generally start with generating structure-related features from a model followed by applying machine learning methods to estimate its local or global quality score. Several features have been proved to be effective, such as sequence/profile alignment, predicted secondary structure and solvent accessibility of residues [108], residue-residue contact potential [112], torsion angle of main chain [123], physicochemical properties [113], and energy-based environment of residues and models [113, 110, 32]. Methods such as support vector machine [108, 124, 32], neural network [31], and linear combination [13, 12] are commonly used for quality estimation. Many top QA methods have been largely tested and assessed in the Critical Assessment of Techniques for Protein Structure Prediction (CASP) [107]. ProQ2 [109] had the good performance on local quality assessment by using machine learning on the features including secondary structure, surface area, contacts information and so on, and its new version ProQ3 [32] that added Rosetta energies as features further improved quality assessment. DeepQA [31] integrated energy-based potential scores with other structural information (i.e., RWplus [125], OPUS [126] and DFIRE [127]) derived from structures and improved the global quality prediction. Qprob [113] combined structural/sequence features, including energy and physicochemical properties of a model, to evaluate its quality. All the methods predict local or global quality separately. No methods tried to predict both quality measurements at the same time, even though some methods derived the global score converted from predicted local quality score of residues [108, 128]. Moreover, traditional machine learning based quality assessment methods used a fixed-size sliding window approach to estimate the local deviation of each residue, in which the features of neighboring residues within a window of a specific size (e.g., 5, 11 and 21 residues) that is centered on a target residue are combined by machine learning approaches to predict the local quality of the residue. Recently, deep learning techniques that can handle input of varied size have achieved significant success in the bioinformatics field [31, 129, 79]. Especially, the application of deep convolutional neural network (CNN) [8, 23, 9] (e.g., 1DCNN for sequential data and 2DCNN for image-like inputs) has achieved the promising performance and becomes one of the best machine learning methods for solving bioinformatics problems [8, 23, 9]. The convolutional neural networks can learn longer-range sequential and structural information from the input features of arbitrary length, which cannot be utilized traditional sliding window approaches.

In this study, we designed novel deep convolutional networks to predict the local and global quality of a protein model consisting of any number of residues, leveraging their capability of handling input of any length. Furthermore, we used a novel multi-task learning framework to study whether global and local quality predictions can synergistically interact to improve prediction performance. Specifically, we developed three novel single-model predictors, InteractQA-net, JointQA-net and LocalQA-net, which use sequence information, structural features, residue-specific Rosetta energies, and other energy scores as input to predict local quality or both global and local quality of a model. We also combined the three predictors to further improve prediction accuracy.

4.3 Methods

4.3.1 Datasets

The dataset for training and validation was downloaded from the 8^{th}, 9^{th} and 10^{th} Critical Assessments of Structure Prediction (CASP) experiments (`http://predictioncenter.org/`), consisting of the models for 322 protein targets whose native structures were officially released. The targets with multiple-domains were removed from dataset because using only single-domain models to train the methods worked better (see the Results and Discussions Section for details), and the remaining protein models for single-domain targets were used for training and validation, leading to 48,574 structural models for 236 single-domain targets. Specifically, the final dataset contains 15,022 models of 82 CASP8 targets, 19,926 models of 87 CASP9 targets, and 13,626 models of 67 CASP10 targets.

The 236 targets were randomly split into the two sets according to the 80% / 20% ratio. The models of 80% targets were used for training and the rest for validation and parameter tuning. Specifically, the final training dataset contains 38,832 models and the validation dataset contains 9,742 models. The independent test datasets include the models of all the single-domain and multi-domain targets of CASP11 and CASP12 experiments. Specifically, 14,076 models of 84 CASP11 targets and 6,008 models from 40 CASP12 targets whose native structure were released to date were included into the test dataset. The true local and global quality scores of the models in the datasets above were obtained by comparing them with the corresponding native structures. The local quality and global scores predicted by other CASP QA methods for the models were downloaded directly from CASP data repository (`http://predictioncenter.org/`) for comparison with our methods.

4.3.2 Feature Extraction

Our one-dimensional deep convolutional networks (1DCNN) take the following residue-wise raw features and several global features as input, which include (1) amino acid encoding of each residue, (2) position specific scoring matrix (PSSM) profile of each residue derived from the multiple sequence alignment of the protein, (3) predicted secondary structure of each residue, (4) predicted solvent accessibility of each residue, (5) predicted disorder state of each residue, (6) the agreement between the secondary structure of each residue in the model and the predicted one and, (7) the agreement of solvent accessibility of each residue in the model and the predicted one, (8) Rosetta energies of each residue as in the ProQ3 [32], which is calculated from Van der Waals, side-chains, Hydrogen bonds, and Backbone information, and (9) six global knowledge-based potentials or features of the entire model produced by ModelEvaluator [118], Dope [117], RWplus [125], Qprob [113], GOAP [130], and Surface score . The amino acid encoding is a vector of 20 binary numbers where the value at the index of the residue index is labeled as 1, otherwise as 0. The PSSM profile is generated by PSI-BLAST [38] searching the sequence against 'nr90' sequence database. SSPro [16] was run to generate the predicted secondary structure and solvent accessibility for each residue in the model. The disorder state of each residue was predicted by PreDisorder [17]. The features of a model of L residues are stored in a vector of length L. Each element of the vector contains all the local features of each residue as well as several global features, which is the input for the deep convolutional neural network.

We used LGA structural alignment tool [131] to measure the local residue-wise distance error and global structural similarity score between models and their native structures. The local distance error is defined by the distance deviation of each residue in a model and in the native structure after superimposing them together, while the global similarity score is defined by the GDT-TS score [132] - the average percent of residues in the model that are close to their positions in the native structure according to several thresholds. We used a function S-function c applied in the previous studies [108, 109] to scale the local distance deviation of residues into the range of [0,1], where d is the distance deviation of a residue between model and native structure, and d_0 is set to 3.0Å. Lower a distance, higher is the S score. d and S can be converted back and forth.

4.3.3 Deep convolutional neural network for protein model quality prediction

We designed three architectures of deep convolutional neural networks (CNN) for predicting the residue-wise local quality of a protein model and investigating the effect of global quality prediction on the local quality prediction. Our first network (LocalQA-net) is designed for local quality prediction using 1D convolutional neural networks, as shown in **Figure 4.1**. The network has one input layer for each protein structure of any length, multiple hidden convolutional layers and one output layer to predict final residual qualities of the same size. In the hidden convolutional layers, the "rectified-linear unit (ReLU)" activation function [133] and batch normalization [134] were applied during training. Our second network (InteractQA-net) consists of two sub-networks for local quality and global quality prediction separately and a common convolutional sub-network of extracting features from the input layer that are shared by the former two, as shown in **Figure 4.2**. On top of the common convolutional sub-network, the sub-network for predicting local quality, referred to as LocalQA-net, has one convolutional output layer with a sigmoid activation function to predict the local quality score for each residue in a model, resulting in L scores for a model of length L. The sub-network for the global quality prediction, referred to as GlobalQA-net, shares the same common network as LocalQA-net, followed by one K-max pooling layer [99] (default K=30), one standard fully connected layer (default 50 hidden nodes), and one single output node to predict the global quality score of an input model. Given a protein model, the InteractQA-net first optimized the weights of LocalQA-net and the common sub-network based on local quality scores. Then the shared weights in the convolutional layers were transferred to GlobalQA-net and both the shared weights and the weights of GlobalQA-net were optimized by global quality scores. After the weights were updated by training on GlobalQA-net, the shared weights were transferred back to LocalQA-net for further optimization. These steps iterated until training converged or the maximum number of iterations was reached. The network was optimized by the Nesterov Adam (nadam) [135] method with Mean Square Error (MSE) as loss function. In order to optimize the performance, we adjusted three main hyper-parameters of convolutional layers during training, including (1) the depth of the network (from 5 to 10), (2) the filter size of the filters in the convolutional layer (from 5 to 10), and (3) the number of filters in each convolutional layer (from 5 to 20). The c (referred as ASE) [136], a standard measure used in CASP to assess the accuracy of local quality prediction, for each parameter setting on the validation was calculated. ASE is the averaged absolute difference of predicted quality score and real quality score of each residue in a model. ASE is defined as $100 * (1 - \frac{1}{\Sigma_{i=1}^{N}|S(e_i)-S(d_i)|})$, which is (1 - the average difference of predicted residue quality ($S(d_i)$) and real residue quality ($S(e_i)$)) times 100. The higher ASE score, more accurate is the local quality prediction. The parameter setting yielding higher ASE was preferred. Each convolutional layer applies the batch-normalization and uses the rectified-linear unit (ReLU) activation function to convert its activation into its output.

In addition to the architecture above, we also designed another architecture called 'JointQA-net' to integrate global quality prediction with local quality prediction, as shown in **Figure 4.3**. The common sub-network and the sub-network for local quality prediction in JointQA-net are the same as InteractQA-net. But JointQA-net has a much simpler sub-network for global quality prediction, which has only one single output node to predict global quality scores. Moreover, instead of alternately training networks using local quality scores and global similarity scores as 'InteractQA-net', JointQA-net predicts both quality scores simultaneously in its output layer in order to optimize all the weights in the network at the same time. Finally, in order to evaluate the effectiveness of incorporating global quality predictions into local quality learning, both InteractQA-net and JointQA-net were compared to the basic network of local quality prediction - LocalQA-net - whose weights were not adjusted according to global quality scores but according to local quality scores only.

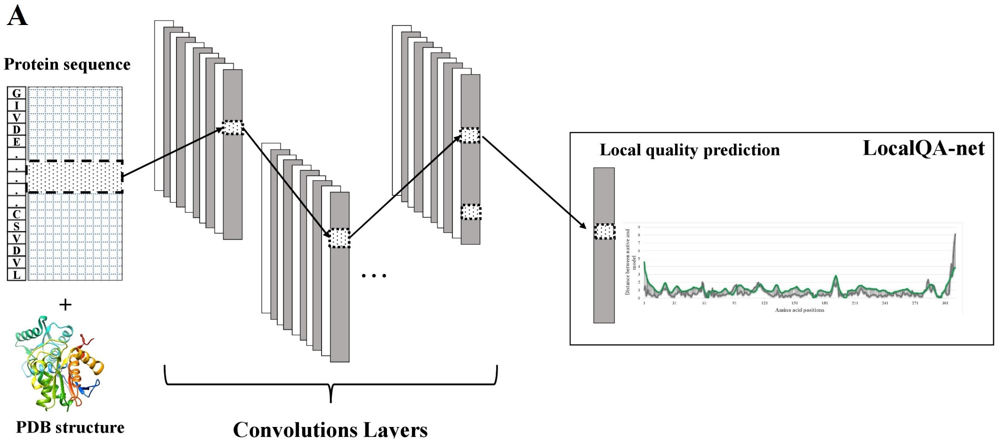


Figure 4.1: The architecture of 1D deep convolutional neural network for protein model quality prediction. (A). The network (LocalQA-net) accepts the raw features of models of proteins of variable sequence length (L) as input, and transforms the features into higher-level hidden features by 5 hidden layers of convolutions. Each convolutional layer applies 5 filters to windows of previous layers to generate L hidden features. The window size for each filter is set to 6. The last output layer adds one convolutional layer with one filter to generate the output of length L representing the local quality for each of L residues.

4.3.4 Evaluation and Benchmarking

We evaluated both local and global quality predictions of our deep learning methods on two sub-sets (1^{st} stage and 2^{nd} stage) of CASP11 and CASP12 datasets, respectively. The local quality predictions were evaluated based on the ASE score [136]. The

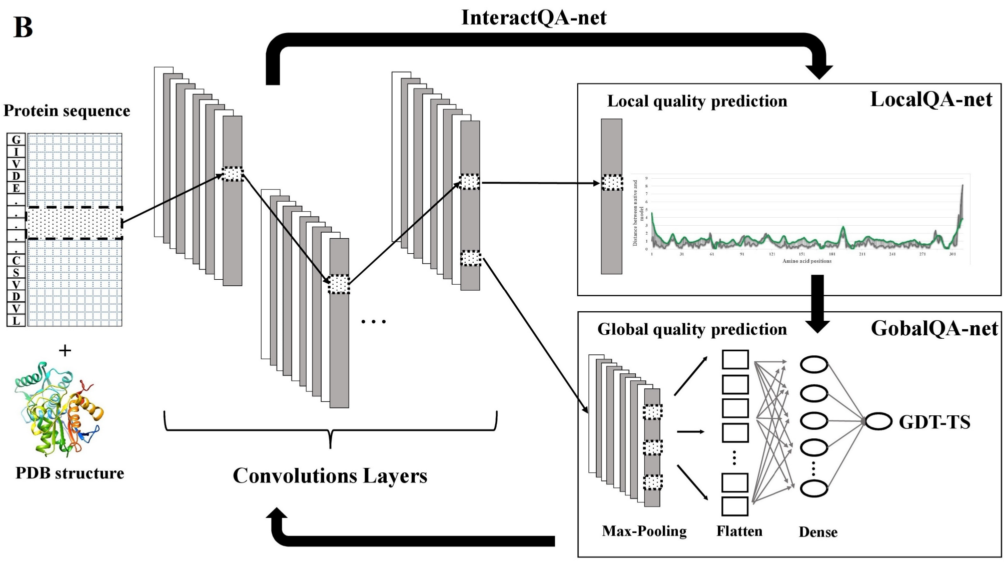


Figure 4.2: The architecture of 1D deep convolutional neural network for protein model quality prediction. (B). The network (InteractQA-net) contains a common sub-network for extracting features from the input layer by convolution and two sub-networks for local quality (LocalQA-net) and global quality (GlobalQA-net) predictions separately. The network accepts the raw features of models of proteins of variable sequence length (L) as input, and transforms the features into higher-level hidden features by 10 hidden layers of convolutions. Each convolutional layer applies 10 filters to windows of previous layers to generate L hidden features. The window size for each filter is set to 15. LocalQA-net adds one convolutional layer with one filter at the top of the common sub-network to generate the output of length L representing the local quality for each of L residues. GlobalQA-net uses one 30-max pooling layer to select 30 maximum values from the output of each filer in the last layer of the common sub-network as features, which are joined together into one vector by a flatten layer. The flatten layer is fully connected to a hidden layer whose output is used by a single output node to predict the global quality score. LocalQA-net and GlobalQA-net are trained by local quality scores and global quality scores alternately.

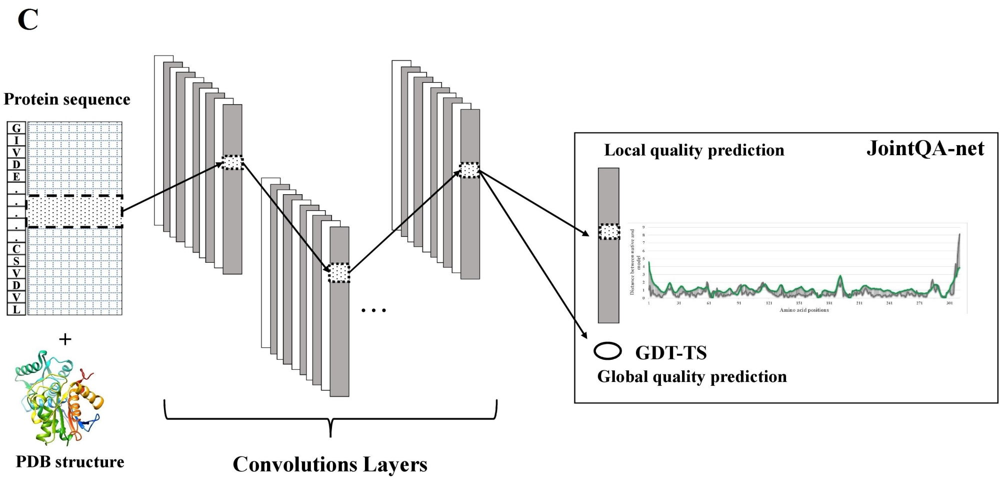


Figure 4.3: The architecture of 1D deep convolutional neural network for protein model quality prediction. (C). JointQA-net accepts the features of protein models of variable sequence length (L) as input and predicts the L local quality scores and one global quality score simultaneously. The weights in the network are optimized by both local and global quality scores at the same time.

global quality predictions were evaluated in terms of (1) Pearson's correlation between predicted global scores of the models of a target and the real global quality scores of the models of the target, and (2) the average loss. The loss is the difference between the real quality score of the no. 1 model selected according to predicted quality scores for a target and the quality score of the real best model of the target. The average loss evaluates the capability of a method to select good models. A loss 0 means the predicted global quality scores can always rank the real best model as no. 1.

We evaluated the performance of our three local quality predictors (InteractQA-net, JointQA and LocalQA-net) on the test datasets and compared them with other QA predictors that participated in CASP 11 and CASP 12. The predictions of CASP QA predictors were directly downloaded from CASP repository (`http://predictioncenter.org/`). In order to evaluate the performance of our global quality predictions, we converted the local quality prediction made by InteractQA-net, JointQA and LocalQA-net into global quality scores by averaging the local quality predictions of residues directly using function $global = \frac{1}{L}\Sigma_{i=1}^{L}\frac{1}{1+(\frac{d_i}{d_0})^2}$. Besides, an ensemble of our three predictors called CNNQA, which uses the average output of the three predictions as its prediction, was evaluated.

4.4 Results and Discussions

4.4.1 Training and parameter optimization

We trained each convolutional network with different parameter setting on our training dataset and selected the best trained model using the ASE metric calculated on the validation dataset. We optimized the following hyper-parameters: the depth of the network (from 5 to 10), the filter size of each convolution layer (from 5 to 15), and the number of filters in each convolutional layer (from 5 to 20). Based on the results on the validation set, the depth of convolutional layers in the InteractQA-net is set to 10, number of filters to 10, and the filter size to 15. For the JointQA-net and LocalQA-net, the final depth of convolution layers is set to 5, number of filters to 5, and filter size to 6 in each convolutional layer. The deep networks trained with these parameters on the training dataset were evaluated on the independent test datasets.

4.4.2 Comparison of local quality predictions with other single-model QA methods on CASP11 and CASP12

We compared InteractQA-net, JointQA-net and LocalQA-net with CASP single-model QA methods on the 1^{st} stage and 2^{nd} stage subsets of CASP11 and CASP12 test datasets. We calculated the average ASE score across all models of each subset for our three predictors and other CASP predictors for comparison (**Table 4.1** and **Table 4.2**). LocalQA-net achieved slightly better performance than InteractQA-net and JointQA-net according to the average ASE scores on the CASP 11 datasets, but it was slightly worse than InteractQA-net and JointQA-net on the CASP 12 datasets, suggesting that including the global quality prediction did not necessarily help with the local quality prediction. However, the accuracy of the ensemble (CNNQA) of the three predictors is higher than each our three predictors, indicating that the three methods are complementary. The performance of LocalQA-net, InteractQA-net, and JointQA-net is comparable to the best performing predictors in CASP11 and CASP12 experiments (e.g., Wang_deep_3, ProQ2, and ProQ3), and CNNQA has slightly higher the average ASE score than all the CASP11 and CASP12 predictors.

Predictor	**Stage1**	**Stage2**	**Average**
CNNQA	81.06	78.43	79.75
LocalQA-net	79.9	78.26	79.08
InteractQA-net	80.27	76.97	78.62
JointQA-net	79.81	77.03	78.42
Wang_deep_3	78.11	74.56	76.34
Wang_deep_2	77.58	74.22	75.9
ProQ2	75.87	75.74	75.81
ProQ2-refine	75.91	75.67	75.79
Wang_deep_1	77.78	73.43	75.6
Wang_SVM	76.79	71.91	74.35
MULTICOM-NOVEL	67.13	67.11	67.12
VoroMQA	62.72	66.45	64.58
MULTICOM-REFINE	62.68	65.26	63.97
MULTICOM-CLUSTER	62.98	64.87	63.92
MULTICOM-CONSTRUCT	63.39	64.35	63.87

Table 4.1: The evaluation results (average ASE scores) of local quality predictions of single-model local quality QA predictors on stage 1 and stage 2 models from CASP 11.

Predictor	Stage1	Stage2	Average
CNNQA	83.22	78.14	80.68
ProQ3	82.19	78.54	80.37
JointQA-net	82.31	77.38	79.85
InteractQA-net	83.13	76.52	79.83
LocalQA-net	80.72	78.09	79.4
Wang4	81.06	76.86	78.96
Wang2	79.62	74.7	77.16
VoroMQAsr	79.32	74.69	77.01
ProQ2	77.73	75.61	76.67
VoroMQA	78.87	74.26	76.56
Wang1	72.22	72.52	72.37
Wang3	53.86	60.55	57.2

Table 4.2: The evaluation results (average ASE scores) of local quality predictions of single-model local quality QA predictors on stage 1 and stage 2 models from CASP12 datasets.

4.4.3 Comparison of global quality predictions with other single-model QA methods on CASP11 and CASP12

In order to evaluate the global quality prediction performance of our methods, we generated the global quality scores for our methods (InteractQA-net, LocalQA-net, LocalQA-net, CNNQA), which were converted from their residue-specific local quality predictions by averaging them using function $global = \frac{1}{L}\Sigma_{i=1}^{L}\frac{1}{1+(\frac{d_i}{d_0})^2}$. We compare them with other QA predictors on the same datasets of CASP 11 and CASP 12 used in the local quality prediction evaluation. We calculated the average Pearson's correlation between predicted global quality scores and real global quality scores as well as the average loss to evaluate the performances of the QA predictors (see the results in **Table 4.3** and **Table 4.4**). According to the Pearson's correlation results on 1^{st} stage and 2^{nd} stage from CASP 11 and CASP 12, InteractQA-net achieved higher correlation and lower loss than LocalQA-net, which showed that integrating the global similarity into local quality prediction improved the global quality prediction derived from the local quality prediction. In terms of average Pearson's correlation, InteractQA-net and CNNQA had the similar performance and both performed better than all other CASP11 and CASP12 predictors. In terms of average loss, InteractQA-net and CNNQA performed better than the other predictors on the CASP11 datasets, but worse than the two top methods (SVMQA [124] and ProQ3 [32]) on the CASP12 datasets.

	Stage1		**Stage2**		**Average**	
Feature	**Corr**	**Loss**	**Corr**	**Loss**	**Corr**	**Loss**
InteractQA-net	0.7243	0.0756	0.4106	0.0596	0.5675	0.0676
CNNQA	0.7126	0.0832	0.3981	0.0594	0.5553	0.0713
LocalQA-net	0.704	0.0839	0.362	0.0639	0.533	0.0739
MULTICOM-CLUSTER	0.6543	0.0965	0.4006	0.0671	0.5275	0.0818
JointQA-net	0.6766	0.0954	0.3695	0.0648	0.523	0.0801
myprotein-me	0.6498	0.0823	0.3875	0.0668	0.5187	0.0745
MULTICOM-NOVEL	0.6517	0.0949	0.3855	0.0623	0.5186	0.0786
Wang_SVM	0.6654	0.1011	0.3676	0.0828	0.5165	0.092
ProQ2-refine	0.664	0.0912	0.353	0.0658	0.5085	0.0785
ProQ2	0.6543	0.0862	0.3535	0.057	0.5039	0.0716
VoroMQA	0.5787	0.1068	0.401	0.0727	0.4899	0.0897
RFMQA	0.6275	0.0956	0.3485	0.0677	0.488	0.0816
Wang_deep_2	0.6395	0.1125	0.3115	0.0832	0.4755	0.0979
Wang_deep_3	0.6338	0.1139	0.305	0.0887	0.4694	0.1013
Wang_deep_1	0.6208	0.1211	0.3071	0.0922	0.464	0.1067

Table 4.3: The evaluation results (Corr. - Pearson's correlation and loss) of global quality predictions of single-model QA predictors on stage 1 and stage 2 models of CASP 11 datasets.

	Stage1		**Stage2**		**Average**	
Feature	**Corr**	**Loss**	**Corr**	**Loss**	**Corr**	**Loss**
CNNQA	0.7283	0.0627	0.627	0.0854	0.6777	0.074
InteractQA-net	0.7208	0.0613	0.625	0.0844	0.6729	0.0728
JointQA-net	0.7124	0.0597	0.623	0.0877	0.6677	0.0737
Wang4	0.7105	0.0689	0.5965	0.113	0.6535	0.0909
LocalQA-net	0.6928	0.0627	0.6112	0.0934	0.652	0.078
MULTICOM-CLUSTER	0.6958	0.0817	0.5895	0.0975	0.6427	0.0896
SVMQA	0.654	0.0353	0.6227	0.0661	0.6383	0.0507
ProQ3	0.647	0.0524	0.6253	0.071	0.6361	0.0617
ProQ2	0.6677	0.0802	0.5981	0.0737	0.6329	0.0769
Wang2	0.6713	0.0766	0.5329	0.144	0.6021	0.1103
VoroMQA	0.6252	0.0798	0.5703	0.1027	0.5978	0.0912
Wang1	0.4517	0.2002	0.2626	0.1486	0.3572	0.1744

Table 4.4: The evaluation results (Corr. - Pearson's correlation and loss) of global quality predictions of single-model QA predictors on stage 1 and stage 2 models of CASP 12 datasets.

4.4.4 Influence of Rosetta energy terms and single-domain targets on the quality predictions

The performance of the three methods with and without Rosetta energies as input and trained on either the single-domain dataset or the full-length dataset was evaluated on Stage 1, Stage 2, and all the models of CASP11 and CASP12 test datasets and were visualized in the **Figure 4.4**. It is worth noting that each network with specific data input was fully tuned to the best performance on the validation dataset before being evaluated on the independent test dataset. As results shown in **Figure 4.4**, adding Rosetta energies improved the local quality prediction in most cases, with an average 1.26 improvement in ASE score. Training the network on the single-domain datasets also generally improved the performance over on the models of all targets (both single-domain and multi-domain targets, with an average 0.49 improvement on the CASP11 and CASP12 datasets in terms of ASE score.

4.4.5 Case study of local quality predictions

Figure 4.5 shows local quality predictions made by our method CNNQA for one model of target T0843 and one model of T0861 from the CASP12 experiment. The **Figure 4.5(A)** plots the real distance (gray) and predicted distance (green) at each residue position of the structural model of T0843, where the two curves overlap well at most positions. **Figure 4.5(B)** shows the superimposition of the native structure (gray) and the structural model(green). The average deviation between actual distances and predicted distances at all residue position in the model is 0.56 angstrom. The red highlighted regions have relatively large deviation (large errors) after the two structures being superimposed. Interestingly, these highlighted regions with a large distance deviation can be captured by the local quality prediction shown in **Figure 4.5(A)**. **Figure 4.5(C)** and **Figure 4.5(D)** show the similar results for the model of T0861, where the average difference between real and predicted distance deviation

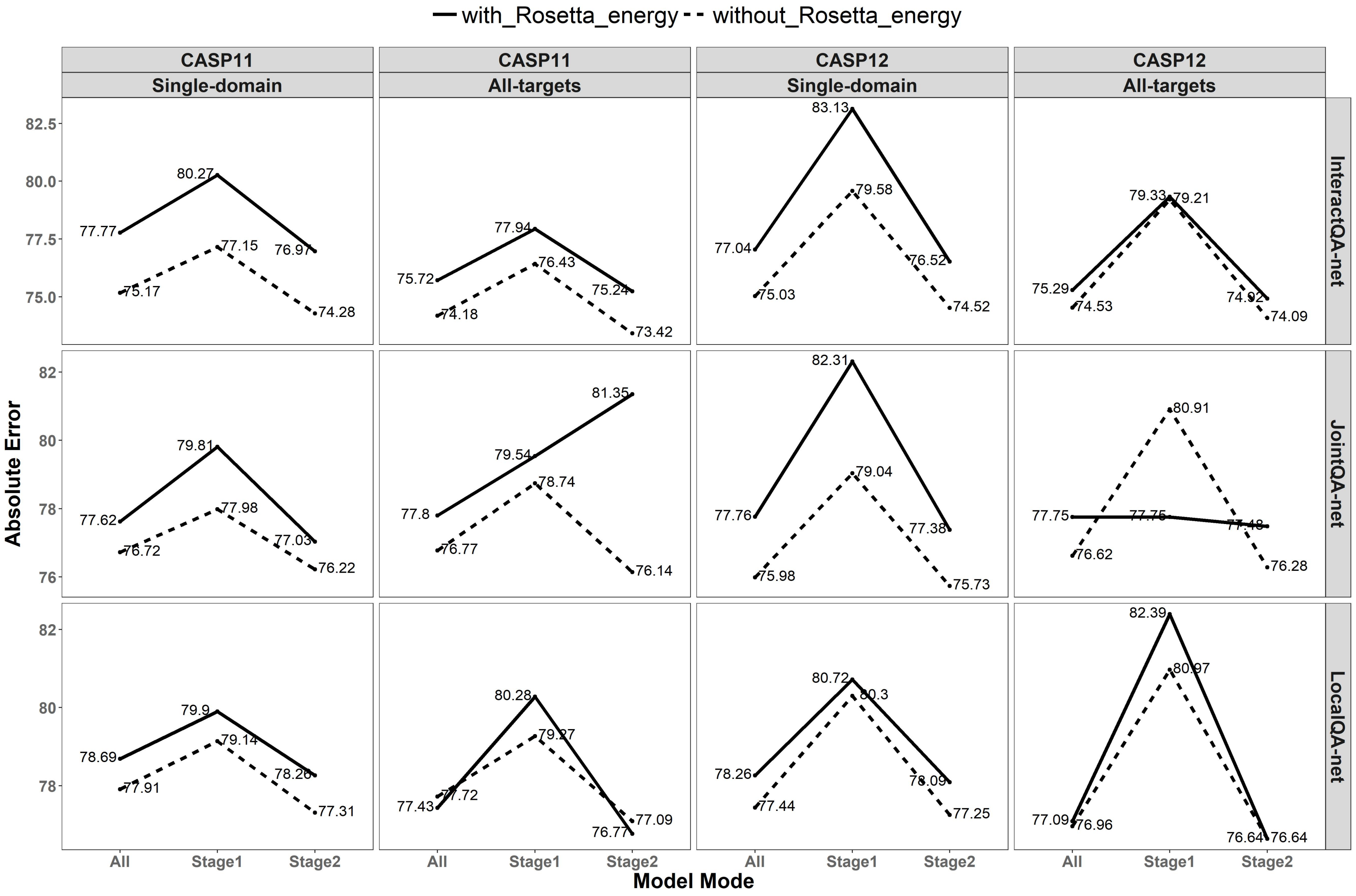


Figure 4.4: The comparison of local quality predictions of the method trained in different situations: (1) with Rosetta energy, (2) without Rosetta energy, (3) using only single-domain models in training, and (4) using both single-domain and multi-domain models (all models of all targets) in training.

is 0.67 angstrom.

4.5 Conclusion

In this work, we presented the novel 1D convolutional neural networks for predicting the quality of a single protein model. Instead of using fixed-size sliding windows to generate features for each residue, our network accepts the input of an entire protein model of arbitrary sequence length and therefore it can access the global structural information that informs the quality of a position of residue. We also designed a new training pipeline to integrate local and global quality prediction together, which improved the accuracy of global quality prediction. Overall, our methods performed comparably to the state-of-the-art methods in the past CASP11 and CASP12 experiments. The results demonstrate that 1D deep convolutional neural networks are promising techniques for protein model quality assessment. In the near future, we will design more advanced deep learning architectures to further advance protein model quality prediction.

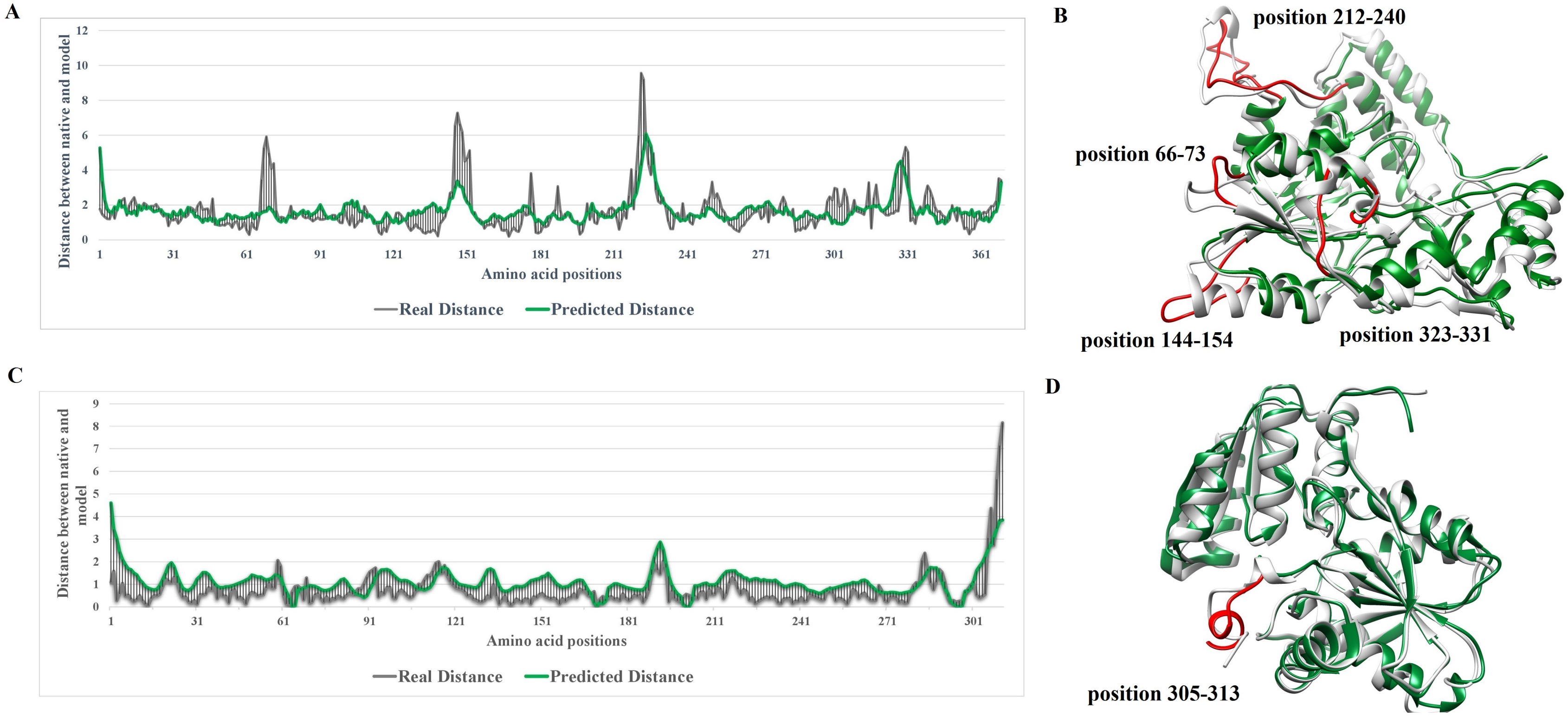


Figure 4.5: Residue-specific distance error predicted by our method (Green) and the real distance error between predicted model and native structure (Gray). (A) The distance error at each amino acid position in the predicted local quality and in the predicted model of T0843. (B) The superimposition of the predicted structure (Green) of the model for target T0843 and its native structure (Gray). The red highlighted regions are the major deviation between predicted model and native structure, matching the large predicted deviation in the local quality prediction. (C) The distance error at each amino acid position in the predicted local quality and in the predicted model of T0861. (D) The superimposition of the predicted structure (Green) of the model for target T0861 and its native structure (Gray).

Chapter 5

Protein tertiary structure modeling driven by deep learning and contact distance prediction in CASP13

5.1 Abstract

Predicting residue-residue distance relationships (e.g., contacts) has become the key direction to advance protein structure prediction since 2014 CASP11 experiment, while deep learning has revolutionized the technology for contact and distance distribution prediction since its debut in 2012 CASP10 experiment. During 2018 CASP13 experiment, we enhanced our MULTICOM protein structure prediction system with three major components: contact distance prediction based on deep convolutional neural networks, distance-driven template-free (ab initio) modeling, and protein model ranking empowered by deep learning and contact prediction. Our experiment demonstrates that contact distance prediction and deep learning methods are the key reasons that MULTICOM was ranked 3rd out of all 98 predictors in both template-free and

template-based structure modeling in CASP13. Deep convolutional neural network can utilize global information in pairwise residue-residue features such as co-evolution scores to substantially improve contact distance prediction, which played a decisive role in correctly folding some free modeling and hard template-based modeling targets. Deep learning also successfully integrated 1D structural features, 2D contact information, and 3D structural quality scores to improve protein model quality assessment, where the contact prediction was demonstrated to consistently enhance ranking of protein models for the first time. The success of MULTICOM system clearly shows that protein contact distance prediction and model selection driven by deep learning holds the key of solving protein structure prediction problem. However, there are still challenges in accurately predicting protein contact distance when there are few homologous sequences, folding proteins from noisy contact distances, and ranking models of hard targets.

The MULTICOM web server is available at: `http://sysbio.rnet.missouri.edu/multicom_cluster/`

The source code of the MULTICOM package is also available at : `https://github.com/multicom-toolbox/multicom`

The supplemental material can be found at: `https://onlinelibrary.wiley.com/doi/full/10.1002/prot.25697`

5.2 Introduction

The major breakthrough in protein structure prediction, particularly template-free (ab initio) prediction, is the drastic improvement of the accuracy of residue-residue contact distance prediction in the recent years, leading to the correct folding of some template-free modeling (FM) targets in CASP11 and CASP12 experiment [103, 11, 137, 138]. The accurate prediction of inter-residue contacts and distances has become a key intermediate step and driving force to predict protein three-dimensional (3D) structure from sequence. The breakthrough in contact distance prediction was driven by two key advances: residue-residue co-evolutionary analysis popularized in [139] and demonstrated in CASP11 and CASP12 experiment [138, 140] and deep learning introduced in [94] and enhanced in [9, 27, 8, 141, 142]. The co-evolutionary analysis is based on the observation that two amino acids in contact (or spatially close according to a distance threshold such as 8Å) must co-evolve in order to maintain the contact relationship during evolution, i.e., if one amino acid is mutated to a positively charged residue, the other one must change to a negatively charged one to be in contact. A number of co-evolutionary methods of calculating direct rather than indirect/accidental correlated mutation scores has been developed and shown to improve contact prediction [7, 143, 144, 145]. Moreover, the co-evolutionary scores can be used as input for machine learning methods to further improve contact prediction. Deep learning, the currently most powerful machine learning method, was introduced into the field in 2012 and demonstrated as the best method for protein contact prediction in 2012 CASP10 experiment [94]. Different variants of deep learning methods - convolutional neural networks and residual networks - were combined with co-evolutionary features to substantially improve contact prediction [9, 9, 8, 141, 142]. The improved contact prediction led to the significant improvement of template-free modeling in CASP12 experiment, in which contact predictions were used with different ab initio modeling methods such as fragment assembly and distance geometry to build protein structural models from scratch [11]. To prepare for 2018 CASP13 experiment, we focused on enhancing our MULTICOM protein structure prediction system [12, 21, 3] with our latest development in contact distance prediction empowered by deep learning and its application to template-free modeling and protein model ranking [112, 13, 12], while having a routine update on its other components such as template library, template identification, and template-based modeling. Our experiment demonstrates that contact distance prediction empowered by the advanced deep learning architecture can accurately predict a large number of contacts for some template-free or hard template-based targets, which are sufficient to fold them correctly by the distance geometry and simulated annealing from scratch without using any template or fragment information. Our experiment also shows that directly translating predicted contacts into tertiary structures by satisfying distance restraints can fold large proteins with complicated topologies better than using contacts indirectly to guide traditional fragment assembly approaches. Moreover, we demonstrate that deep learning can integrate 1D, 2D and 3D structural features to improve protein model ranking. Particularly, we show that, for the first time, improved contact prediction can consistently improve protein model ranking. Therefore, contact distance prediction and deep learning are the key driving force that made our MULTICOM predictor rank third in the CASP13 experiment in both template-based and template-free modeling. The success of MULTICOM human and server predictors (MULTICOM_CLUSTER, MULTICOM-CONSTRUCT and MULTICOM-NOVEL) in CASP13 clearly proves that deep learning holds the key for protein contact distance prediction and folding, even though there are still significant challenges in contact/distance prediction for targets with few homologous sequences, translation of noisy or sparse contact distances into 3D models, and selecting a few good protein structural models from a large pool of low-quality ones for a hard target.

5.3 Materials and Method

In this section, we first provide an overview of the MULTICOM server and human prediction system, followed with the detailed description of several key new components that we added into the MULTICOM system in CASP13, such as the protein contact distance prediction empowered by deep learning, ab initio protein structure prediction driven by predicted contact distances, and large-scale protein quality assessment enhanced by deep learning and contacts.

5.3.1 An overview of the MULTICOM system

Figure 5.1 is an overview of our MULTICOM server and human prediction systems. Once the server received a target protein sequence, MULTICOM searched it against protein sequence databases such as the non-redundant sequence database to collect its homologous sequences to generate multiple sequence alignments, which were used to build sequence profiles such as Position Specific Scoring Matrices (PSSM) [38] and Hidden Markov models (HMM) [15]. The sequence was also used to predict one-dimensional (1D) structural features including secondary structure, solvent accessibility, and disorder regions [17, 16].

The profile or sequence of the target was searched against the template profile/sequence library by a number of sequence alignment tools (e.g., BLAST [82], CSI-BLAST/CS-BLAST [146], PSI-BLAST [38], COMPASS [147], FFAS [148], HHSearch [80], HHblits [15], HMMER [149], JackHMMER [150], SAM [151], PRC [152], RaptorX [153], I-TASSER/MUSTER [30, 154]) to identify protein templates whose structures were known and build pairwise target-template sequence alignments. DeepSF - a deep learning method of classifying protein sequences into folds was also used to identify templates for the target [23].

In parallel to the template identification, the multiple sequence alignments of the target were also used to generate co-evolutionary features by CCMpred [145], FreeContact [155] and PSICOV [144], which were used together with other sequential and structural features such as predicted secondary structure and solvent accessibility as input for DNCON2 [8] to predict residue-residue contacts at multiple distance thresholds (i.e., 6 Å, 7.5 Å, 8 Å, 8.5 Å and 10 Å).

The target-template sequence alignment was used to identify domain boundaries, i.e., the region of the target not aligned with any significantly homologous template was treated as a template-free modeling domain, otherwise a template-based domain. The contact prediction for template-free domains was made by DNCON2 and combined with the contact prediction of the full-length target.

The pairwise target-template alignments were combined into the multi-template alignments between the target and the multiple templates if the structures of the templates were consistent [20]. The alignments and the structures of templates were fed into Modeller [5] to build the structural models for the target. Generally, more than 100 template-based models were constructed for a target.

In parallel to the template-based modeling, predicted contacts were used with several ab initio modeling tools such as CONFOLD2 [10], Rosetta [156], UniCon3D [18] and FUSION [19] to build structural models for a template-free target or domain. Both the template-based models and/or template-free models were added into a model pool for model ranking.

The MULTICOM human predictor also used all CASP13 server models as input. The incomplete server models or highly similar models (e.g., GDT-TS > 0.95) from the same server group were filtered out. The side chains of the remaining models were repacked by SCWRL [157] in order to have the consistent side chain packing before they were evaluated, which was shown to improve the performance of model quality assessment [107]. If the target was identified as multiple-domain protein, the server models were divided into individual domain models.

The structural models from either MULTICOM human predictor or server predictors were compared with 1D structural features (e.g., predicted secondary structure, solvent accessibility) to generate 1D matching scores and with 2D contacts to generate 2D matching scores (i.e., the percentage of predicted contacts existing in a model of the target). The models were also assessed by a number of 3D quality assessment tools to generate 3D quality scores. The 1D, 2D, and 3D quality scores (features) were used by DeepRank - our deep learning-based model quality assessment tool - to predict the accuracy of the models. This quality assessment method was also applied to individual domains if a target had multiple domains. It is worth noting that our three server predictors used different quality assessment methods for model selection. MULTICOM_CLUSTER ranked models primarily based on pairwise similarity scores between models using APOLLO [122], while MULTICOM-CONSTRUCT and MULTICOM-NOVEL selected best five models based on our two new deep learning-based model ranking methods (DeepRank and DeepRank_avg, described in details in Section 5.3.4).

The quality assessment scores were used to rank full-length and/or domain-based models and the top ranked models were selected for model combination and refinement. Each top ranked model was combined with other similar models in the ranked list to generate a consensus model. If the consensus model is not substantially different from the initial model (i.e., GDT-TS > 0.88), it was kept as the final model. Otherwise, it was discarded and 3DRefine [14] was used to refine the top ranked model to generate a refined final model.

In summary, our human predictor differs from the server predictors in several aspects. First, the structural models for a target protein used for model evaluation in the three MULTICOM server predictors were generated by them locally. The MULTICOM human predictor evaluated all server models that were generated by many different CASP13 server predictors, including our three MULTICOM server predictors. Second, the domain boundary determination was somewhat different. Our server predictors used target-template sequence alignments to identify domain boundaries. The MULTICOM human predictor further adjusted the domain boundaries predicted by the servers according to the domain boundaries of the top CASP13 models selected by the model quality assessment method. The domain boundaries of the top CASP13 models were obtained by the domain parsing tools - DomainParser [158] and PDP [159].

Third, the side chains of CASP13 server models were repacked by the MULTICOM human predictor before they were evaluated in order to make the side chains of the models predicted by different CASP13 server predictors consistent, while the MULTICOM server predictors did not have this step. And fourth, the final selected models were further refined by 3Drefine in the MULTICOM human predictor, whereas the MULTICOM server predictors did not use refinement.

5.3.2 Deep convolutional neural network for contact distance prediction

We used DNCON2 to generate the 2D contact map for an input sequence [8]. As shown in **Figure 5.2**, a target sequence was searched against Uniprot20 database (version: 2016˙02) by HHblits [15] to collect homologous sequences and generate multiple sequence alignments. If there is not a sufficient number of homologous sequences (e.g., $<$ 5L sequences; L sequence length), the target was further searched against Uniref90 database (released by April 2018) by JackHMMER [150] to collect more homologous sequences whose multiple sequence alignments were combined with the results of HHblits search. The multiple sequence alignments were used by CCMPred [145], FreeContact [155], and PSICOV [144] to generate residue-residue co-evolution features. The pairwise co-evolution features together with other pairwise information (e.g., secondary structure, solvent accessibility, and mutual information for each pair of residues) were stored in the L×L input matrices (L: sequence length or domain length). The input feature matrices were used by the first-level convolutional neural networks in DNCON2 to predict the contact probability maps (i.e., contact distance distribution) at multiple distance thresholds 6 Å, 7.5 Å, 8 Å, 8.5 Å and 10 Å. The distance distribution and the original input matrices were concatenated as input for the second-level convolutional neural networks to predict a final contact probability map at 8 Å distance threshold.

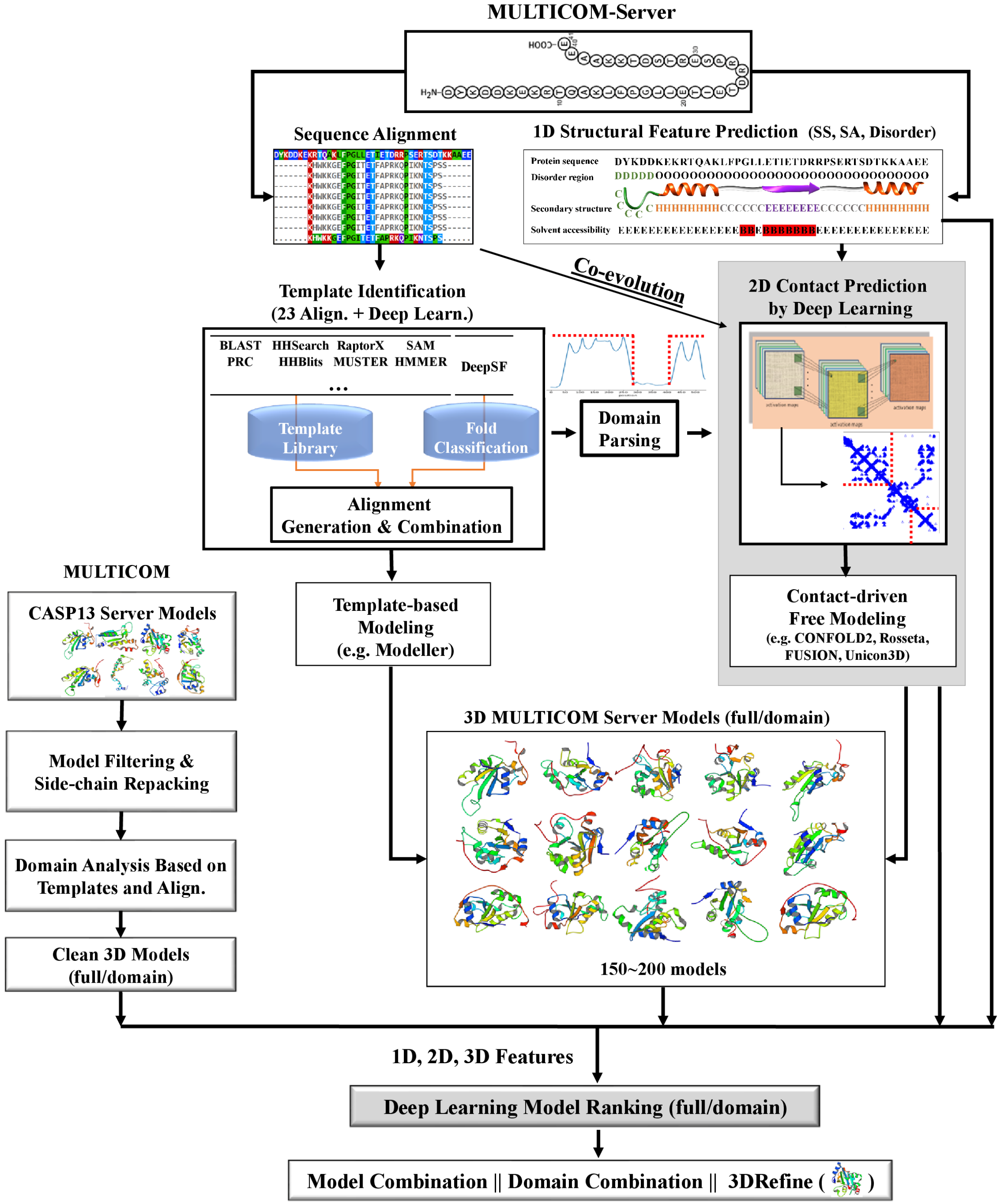

MULTICOM-Server
Sequence Alignment
1D Structural Feature Prediction (SS, SA, Disorder)
Protein sequence
Disorder region
Secondary structure
Solvent accessibility
Co-evolution
Template Identification
(23 Align. + Deep Learn.)
BLAST
PRC
HHSearch
HHBlits
RaptorX
MUSTER
SAM
HMMER
...
DeepSF
Template Library
Fold Classification
Alignment Generation & Combination
Domain Parsing
2D Contact Prediction by Deep Learning
Contact-driven Free Modeling
(e.g. CONFOLD2, Rosseta, FUSION, Unicon3D)
MULTICOM
CASP13 Server Models
Template-based Modeling
(e.g. Modeller)
Model Filtering & Side-chain Repacking
Domain Analysis Based on Templates and Align.
Clean 3D Models (full/domain)
3D MULTICOM Server Models (full/domain)
150~200 models
1D, 2D, 3D Features
Deep Learning Model Ranking (full/domain)
Model Combination || Domain Combination || 3DRefine ( )

Figure 5.1: The pipeline of MULTICOM server and human prediction systems.

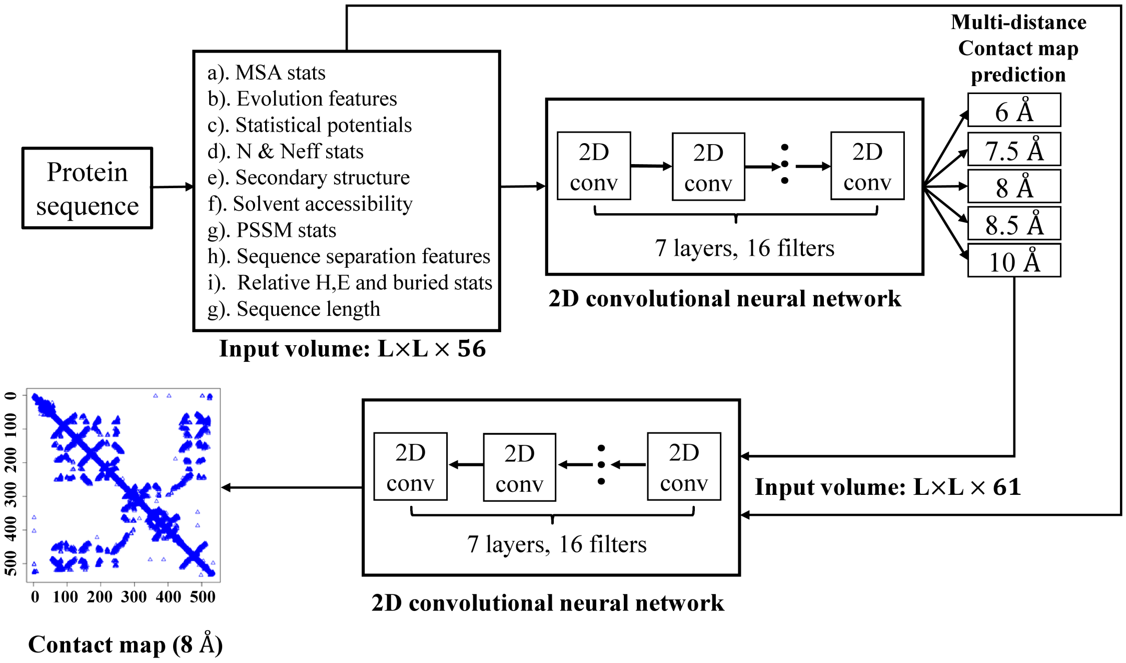


Figure 5.2: The pipeline of DNCON2 for protein residue-residue contact distance prediction. The input volume has 56 channels (matrices) containing various pairwise residue-residue features.

5.3.3 Contact distance-based ab initio folding

We used predicted contacts with a pure contact distance-based ab initio modeling tool - CONFOLD2 and several fragment-assembly tools to build 3D models for targets or domains without significant templates being identified. CONFOLD2 [10] used only predicted contacts and secondary structures to build structural models without leveraging any other information such as structural fragments (**Figure 5.3**). Top x × L contacts (x: a ratio ranging from 0.1 to 4; L: length of the protein) ranked by probabilities were used to generate distance restraints between $C\beta$ atoms (or $C\alpha$ atom for glycine). The predicted secondary structures were used to generate torsion angle restraints, atom-atom distance restraints, and hydrogen-bond restraints [104], which were important for building good local secondary structures in the model. These restraints were used by the distance geometry and simulated annealing optimization implemented in CNS [160] to build tertiary structure models by satisfying the restraints as well as possible. In this round of modeling, some local structures, particularly beta-sheets, are often not well formed due to lack of restraints or noisy restraints. To remedy the problem, the potential beta-sheets were detected in the models generated by the first round of modeling. More angular, hydrogen bond, and atom-atom distance restraints were added in order to improve the pairing between the beta strands. Moreover, the contact distance restraints that were not realized in the models were removed from the list. The new set of restraints were used by the distance geometry again to build 3D models. Usually, a few hundred of models were constructed by using different numbers of contact distance restraints (i.e., 0.1L, 0.2L,..., 3.9L, 4L), which were then clustered. Top models from the clusters were selected as final models. The key feature of this approach is that contacts play a dominant and direct role in building structural models. If there are a sufficient amount of accurate distance restraints, high-quality 3D models can be constructed.

As an alternative, we also used predicted contacts as distance or contact restraints with three fragment assembly methods - Rosetta [156], UniCon3D [18], and FUSION [19] to build models. Contacts were used as a part of the energy function of these methods to guide the assembly of protein structure. Rosetta used the structure fragments drawn from a fragment library to assemble the structure, while UniCon3D and FUSION used hidden Markov models to generate conformations for fragments of variable length. In contrast to the CONFOLD approach [104, 10], extra information such as fragments and energy terms is used in this kind of approach, in which contacts only play an indirect or auxiliary role in structural modeling. Therefore, the fragment assembly approach may fail if its conformation sampling cannot generate correct topologies, which often happens for relatively larger proteins with complicated topologies, even though there is a good amount of accurately predicted contacts. To assist the fragment-assembly with contacts, we selected top L/5 predicted contacts of short-range, medium-range and long-range, which were translated into the distance constraints between pairs of $C\beta$-$C\beta$ as additional energy terms. Rosetta and FUSION used the bounded potential for a distance d, which is defined as follows:

$$f(d) = \begin{cases} (\frac{d-lb}{sd})^2 & \text{if } d < lb \\ 0 & \text{if } lb \leq d \leq ub \\ (\frac{d-ub}{sd})^2 & \text{if } ud < d \leq ub + 0.5 * sd \\ \frac{1}{sd}(d - (ub + 0.5 * sd)) + (\frac{0.5*sd}{sd})^2 & \text{if } d > ub + 0.5 * sd \end{cases} \quad \text{with } sd = 0.5$$

The parameters "lb" and "ub" are lower and upper bounds for atom-atom distance, which had been optimized and set to 3.5 Å and 8 Å in our experiment. Unicon3D adopted a square well function with the exponential decay to account for the contact distance energy and is defined as:

$$f(d) = \begin{cases} -P & \text{if } d < d_0 \\ -P * e^{-(d-d_0)^2} + P * \frac{d-d_0}{d} & \text{if } d > d_0 \end{cases}$$

, where P is the predicted contact probability for a pair of atoms. In CASP13, the contact-based ab initio structure prediction was run for up to two days to generate decoys for model selection.

5.3.4 Protein model ranking by DeepRank integrating 1D, 2D and 3D features

To select most accurate models from a set of predicted structures, we developed a deep learning-based quality assessment (QA) method, DeepRank, by integrating multiple QA methods and contact predictions for predicting the global quality of models. Given a pool of models, it first generated one-dimensional (1D) features representing the

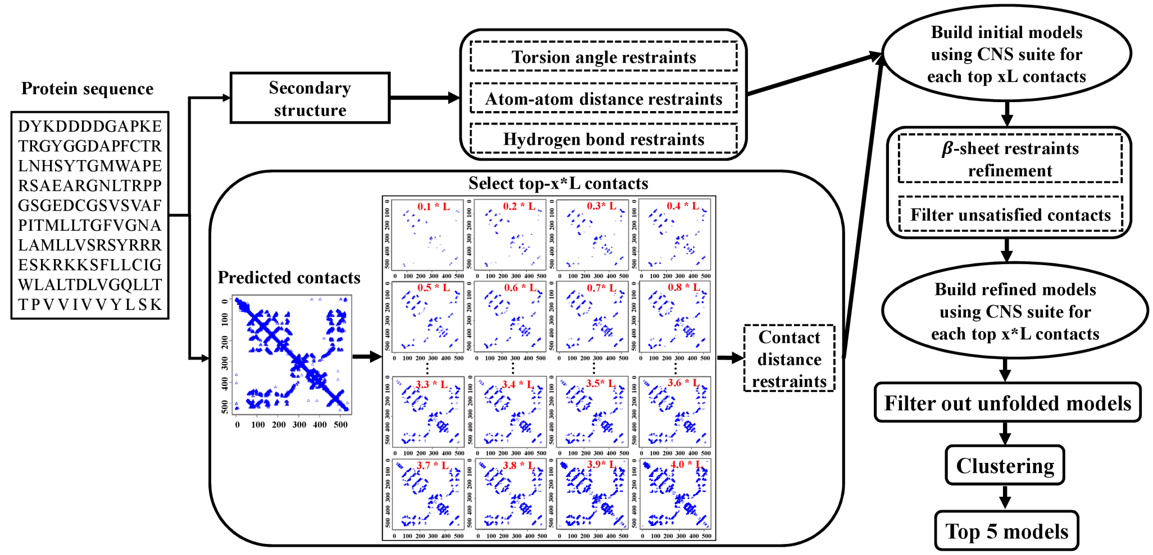


Figure 5.3: Automated contact distance-based ab initio protein structure prediction by CONFOLD2.

similarity between the secondary structure and solvent accessibility predicted from the protein sequence by SSPro [16] and the ones parsed from each protein model by DSSP [66]. The percentage of inter-residue contacts (i.e., top L/5 short-range, medium-range and long-range contacts, respectively) predicted by DNCON2 [8] existing in a model was used as 2D contact features. It also generated 3D quality scores for each model by using 9 single-model QA methods (i.e., SBROD [161], OPUS˙PSP [162], RF˙CB˙SRS˙OD [125], Rwplus [163], DeepQA[31], ProQ2 [109], ProQ3 [32], Dope [117] and Voronota [164]) as well as three multi-model QA methods (i.e., APOLLO [122], Pcons [165], and ModFOLDclust2 [120]). These features were used by two-level neural networks to predict the quality scores of the models (**Figure 5.4**). In the first level, all the 1D, 2D and 3D quality features were fed into 10 pre-trained neural networks to predict the quality (GDT-TS score) of each model. These networks were trained on the models of CASP8-11 experiments and rigorously benchmarked on the CASP12 targets. The ensemble of 10 networks was constructed as in the following steps: (1) All the server models of 425 CASP8-11 targets were randomly split into 10 equal-size subsets by targets (i.e., each subset contained all server models of the targets allocated to it); (2) Each subset was used as the validation data for selecting the network parameters (i.e., the number of layers and hidden nodes), and the remaining 9 subsets were used as training data for network training. The architecture with the lowest average loss (i.e., the difference between the GDT-TS score of the top selected structural model and the GDT-TS score of the best structural model of a target) on the validation subset was selected as the final network for this subset. This process was repeated 10 times (i.e., 10-fold cross-validation), with each of the 10 subsets was used as validation data once, yielding ten pre-trained neural networks. All the input features of each structural model were fed into the 10 trained networks to generate 10 quality scores. In the second level, the 10 predicted quality scores and the initial input features were used together by another deep neural network to predict the final quality score. The second-level network was also trained on the all models of CASP8-11 targets, where the models were randomly split into the training and validation data with ratio 9 to 1. The details of the network configuration are reported in supplemental Table S1. This method was also blindly tested as 'MULTICOM_CLUSTER' in the CASP13 quality assessment category and ranked as one of the best predictors in selecting models and estimating the absolute error (see supplementary Table S2 for details). We also developed a simplified DeepRank method (called DeepRank_avg) by averaging the predictions from the 10 trained networks in the first level as the final quality score.

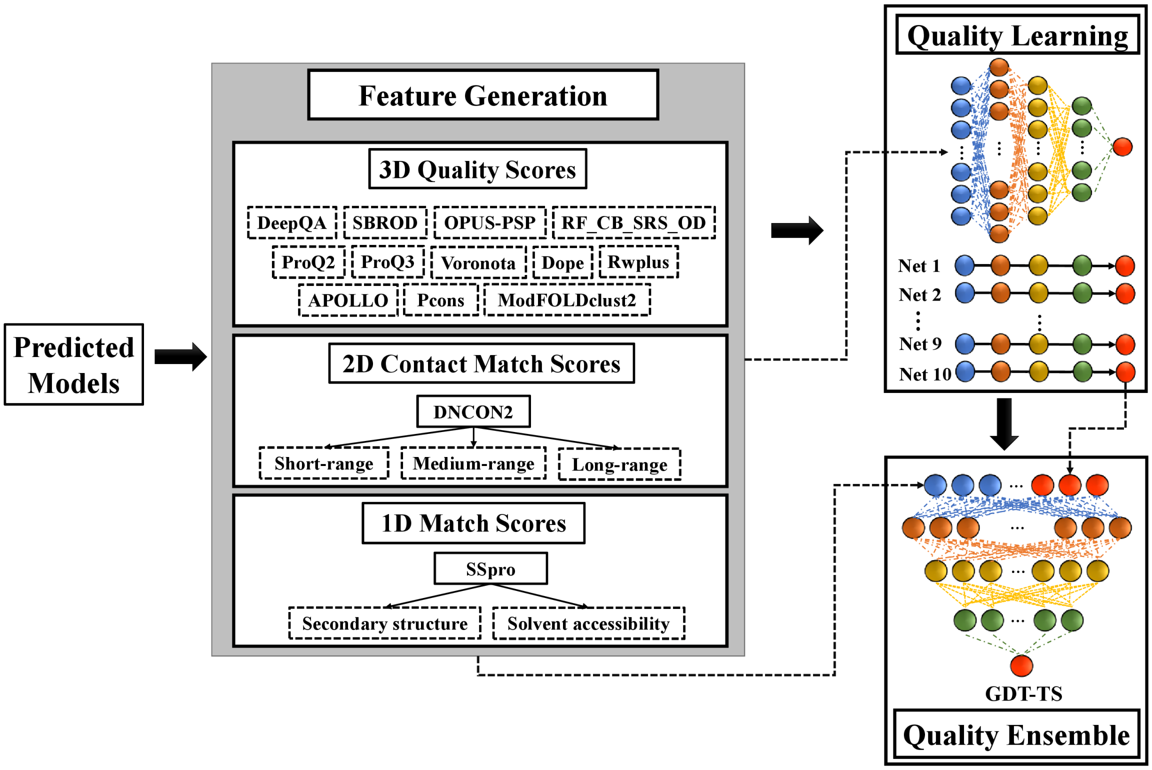


Figure 5.4: The workflow of deep learning-based model quality assessment with contacts (DeepRank).

5.4 Results and Discussions

5.4.1 Performance of MULTICOM human and server predictors in CASP13

We evaluate the performance of MULTICOM methods on 104 "all groups" domains that were used in CASP13 official evaluation. Based on the official domain definition of CASP13, the 104 domains were classified into 31 free-modeling (FM) domains, 40 template-based easy (TBM-easy) domains, 21 template-hard (TBM-hard) domains, and 12 FM-TBM domains.

Figure 5.5 shows the performance of MULTICOM human predictor and our three server predictors based on the TM-score metric [132]. According to the evaluation, as shown in **Figure 5.5(A)**, MULTICOM human predictor outperforms the three server predictors and MULTICOM-CONSTRUCT ranked better than MULTICOM_CLUSTER, followed with MULTICOM-NOVEL in terms of averaged TM-score on 104 domains. On all the domains, the average TM-score of MULTICOM is 0.69, significantly higher than 0.59 of MULTICOM-CONSTRUCT (difference = 0.1; P-value = 4.478E-14), whereas the difference between the two on template-based easy domain (i.e., 0.04) is much smaller and on template-free domains (i.e., 0.19) is much larger. **Figure 5.5(B)** shows the performance of four predictors on the 40 TBM-easy domains. MULTICOM-CONSTRUCT and MULTICOM-NOVEL achieved higher TM-score than MULTICOM_CLUSTER. The major difference among the three servers is the QA methods employed for model selection. The three QA methods: DeepRank, DeepRank_avg and APOLLO (a pairwise model comparison method) were used in the MULTICOM-CONSTRUCT, MULTICOM-NOVEL and MULTICOM_CLUSTER, respectively. As shown in supplemental Figure S5, DeepRank has the higher capability of model selection than APOLLO. Especially for the template-based targets, DeepRank has a much lower loss (GDT-TS score 0.039) compared to the APOLLO's loss (0.059) in model selection. The better ability of model selection in template-based targets led to better tertiary structure prediction for MULTICOM-CONSTRUCT (ΣGDT-TS = 75.83) than MULTICOM_CLUSTER (ΣGDT-TS = 72.91) as shown in supplemental Figure S2. **Figure 5.5(C)** reports the results of the four predictors on the 31 free-modeling domains. MULTICOM human predictor successfully predicted correct fold for 17 out of 31 domains (TM-score >0.5).

Supplemental Figure S1 compares MULTICOM with other top ranked CASP13 groups. MULTICOM (group number: '089') is consistently ranked among the top three predictors according to all metrics on the three domain sets. For instance, it is ranked no. 3 according to z-score on all 104 domains. Figure S2 shows the performance of our three MULTICOM server predictors and other top ranked server groups on the 112 "all groups" and "server only" domains. MULTICOM-CONSTRUCT ranked 7th among all server groups on all the targets, followed by MULTICOM_CLUSTER and MULTICOM-NOVEL. The performance of the global and local quality metrics defined by GDT-TS [132], and LDDT score [166] are also summarized in Figure S3 and Figure S4. We also analyzed the performance of the different alignment tools used by our server predictors. The results are summarized in supplementary Table S3.

5.4.2 Performance of DeepRank and individual QA methods used by MULTICOM

To assess how well the model ranking component of MULTICOM predictors worked, we evaluate the results of DeepRank and the individual QA methods used by DeepRank on the CASP13 targets. The loss of each QA method on the 74 CASP13 "all group" full-length targets whose experimental structures are available was calculated and visualized in **Figure 5.6(A)**. The loss is defined as the difference between the GDT-TS score of the top selected model by each method and the GDT-TS score of the best model of the target. The lower average loss represents the better capability of a

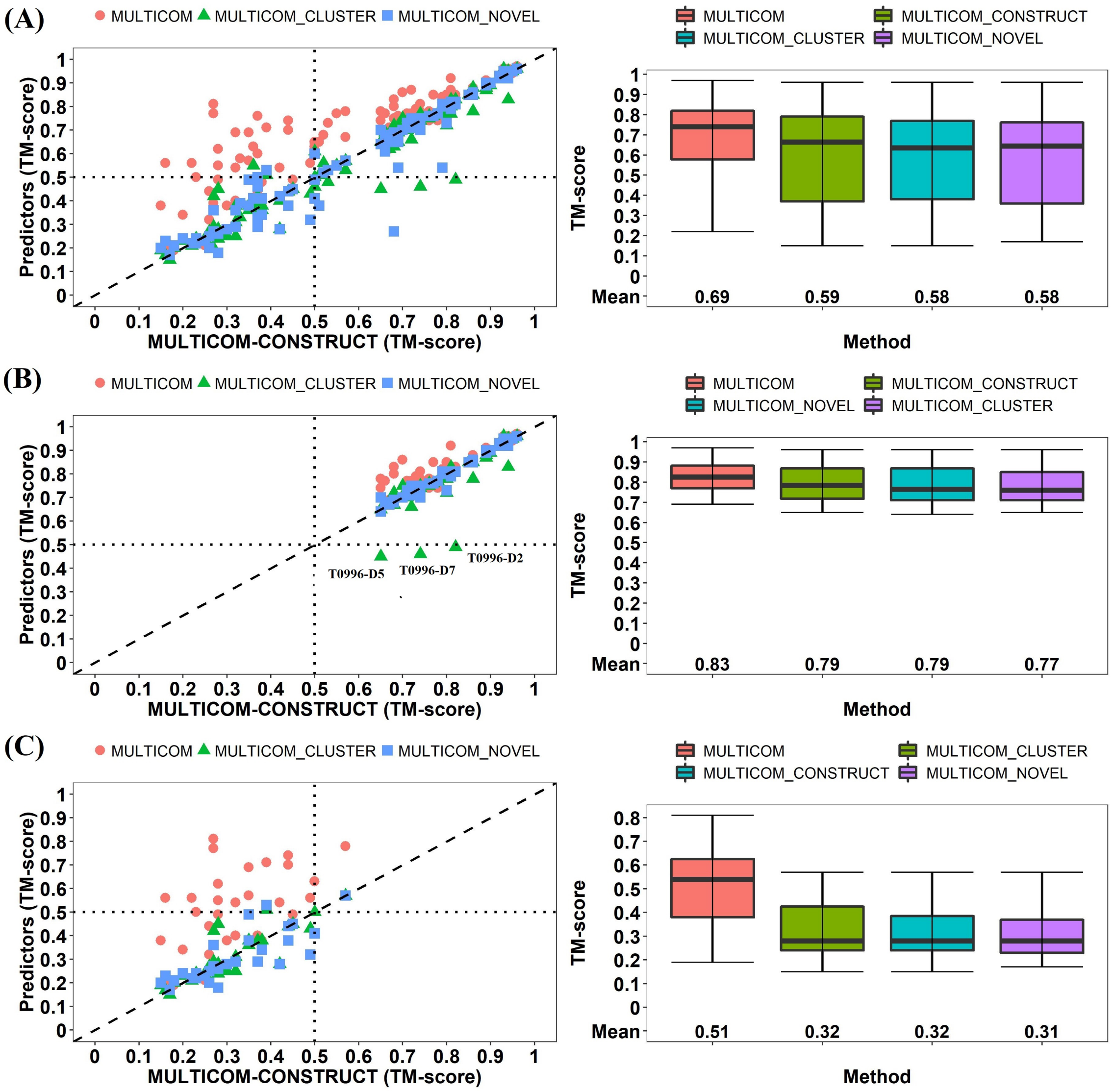


Figure 5.5: Evaluation of four MULTICOM predictors. The methods are ranked by average TM-score of the first (i.e., TS1) submitted models. (A) on 104 domains (Left plot: TM_scores of MULTICOM, MULTICOM_CLUSTER, MULTICOM-NOVEL models versus TM_scores of MULTICOM-CONSTRUCT models; Right plot: mean and variation of the TM-scores of the models of the four methods). (B) on 40 template-based (TBM-easy) domains. (C) on 31 template-free (FM) domains.

QA method for model selection. 24 QA methods are categorized into four groups, including (1) our deep learning integration of diverse quality assessment methods (DeepRank), (2) 3 contact match scores, (3) 3 clustering-based methods, and (4) 17 single-model QA methods. The results show that DeepRank had the lower average loss (0.052) than other individual QA methods on all 74 all-group targets. **Figure 5.6(B)** plots the GDT-TS scores at the 100-point scale of the top models selected by each individual QA method and DeepRank against the GDT-TS scores of MULTICOM's first submitted models. The fitted curve for each method is highlighted in different colors. The larger area under the curve represents the better overall accuracy of model selection. The analysis shows that DeepRank achieves higher GDT-TS scores (Avg. GDT = 54.90 at 100-point scale, i.e., 0.549 at 1-point scale) for model selection than the clustering-based method APOLLO (Avg. GDT = 53.31 at 100-point scale, i.e., 0.5331 at 1-point scale), and also outperforms all other QA methods.

Prior to CASP13, we assessed how much the deep learning and contact prediction improved the quality assessment in CASP12 dataset. After the quality scores were generated using individual QA methods, two baseline combination strategies (e.g., the average score of raw feature scores and Z-scores respectively) were compared with the deep learning. Supplemental Table S4 shows that the Z-score based consensus worked better than the average score consensus, while the deep neural network of integrating all features except contacts further reduced the loss from 0.064 of the z-score based consensus to 0.054. Furthermore, the deep learning with contact features performed best (correlation = 0.853 and loss = 0.048), and the improvement was significant compared to the averaging approach (loss = 0.067) according to the P-value (0.007751). The average loss of the deep learning with contacts is 0.051 on the 74 CASP13 targets, lower than 0.059 of the deep learning without contacts that is lower than both the average score consensus (loss = 0.073) and z-score consensus (loss = 0.057). The improvement is also consistent with the results in the blind CASP13 experiment (supplemental Table S5). This further validated the deep learning and contact prediction's positive contribution to model selection.

Figure 5.7 illustrates how MULTICOM estimated the quality of models for a TBM-hard target T0966 and predicted the final structure. **Figure 5.7(A)** visualized the distribution of the GDT-TS scores of 146 server models for this target. It is a bimodal distribution, where the GDT-TS scores of major models are centered around 0.1 and 0.5. **Figure 5.7(B)** is the plot of the true GDT-TS scores of models against their predicted ranking by DeepRank. It successfully ranked the model with highest GDT-TS score (0.6103) as No.1 (**Figure 5.7(D)**). MULTICOM generated a refined model by combining the top 1 selected model with the other top ranked models, which had a GDT-TS score of 0.6113 (**Figure 5.7(E)**). The ranking of individual QA methods for this target is shown in Figure S6. The other three such successful cases for DeepRank are also reported in Figures S7, S8 and S9.

To assess how contact predictions can help model ranking, we evaluated DeepRank with/without contact features on targets with low contact prediction precision and ones with high contact prediction precision, respectively (Figure S10). The consistent, significant improvement in model selection has been observed when the contact prediction of short-range, medium-range, and long-range has high precision (precision > 0.5). However, the less accurate contact prediction led to the slightly worse performance on model selection than not using contact prediction.

We also analyzed the effect of side-chain repacking on model evaluation. The results show that repacking the side chains of models before they were evaluated reduced the loss of modeling ranking. The detailed results are reported in supplementary Table S6 and Figure S11.

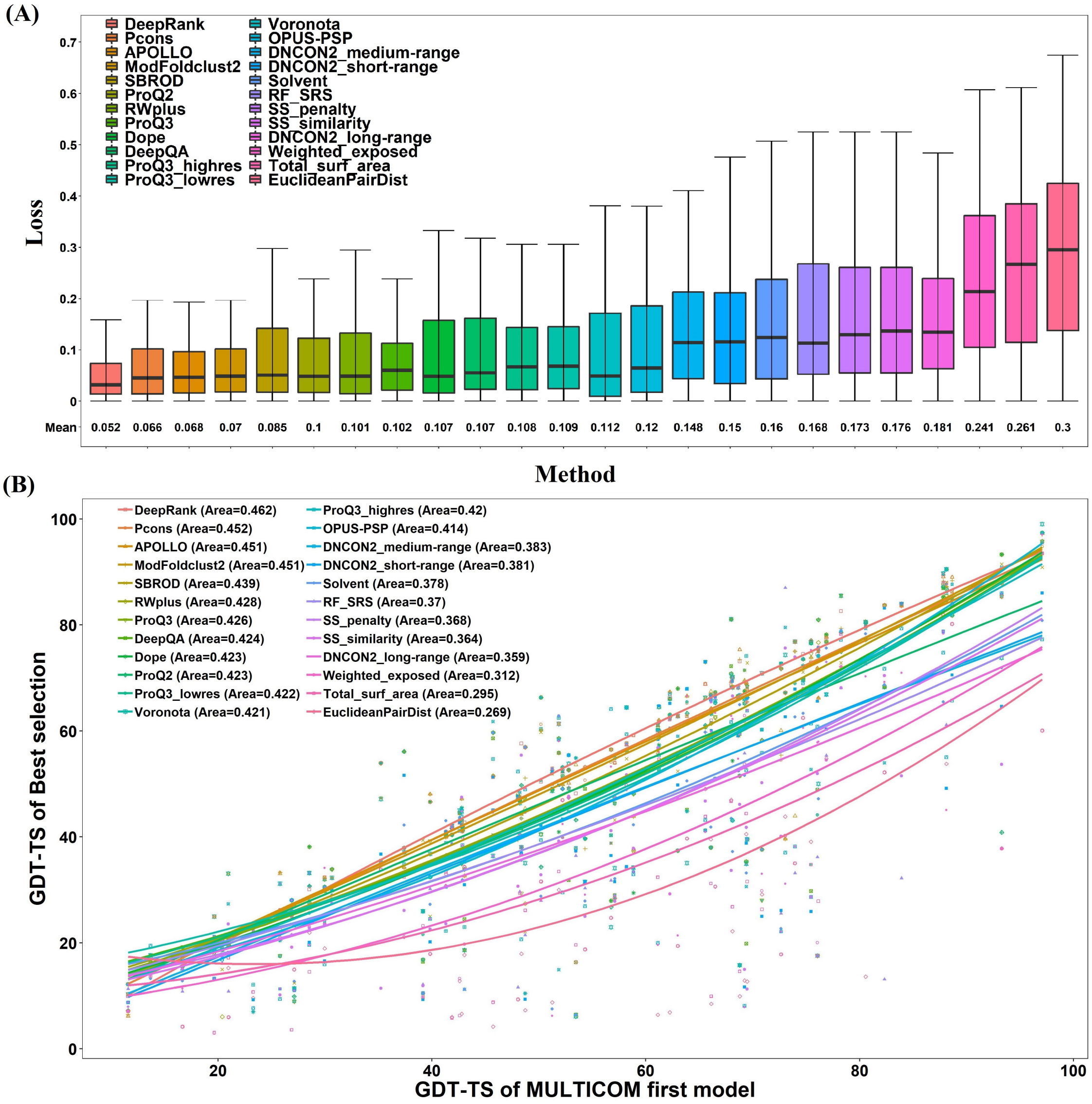


Figure 5.6: Comparison of DeepRank with individual QA methods used in MULTICOM predictors. (A) The box plot of loss of each method. Here the loss is measure at 1-point scale (i.e., the highest/perfect GDT-TS score = 1). (B) The GDT-TS score at the 100-point scale of the top models selected by each individual QA method and DeepRank is plotted against the GDT-TS score of MULTICOM's first submitted models for 74 "all group" full-length targets. The curve for each method is fitted by the second-degree polynomial regression function. The area under the curve for each method is calculated and shown on the top left. The larger area indicates the better capacity of model selection.

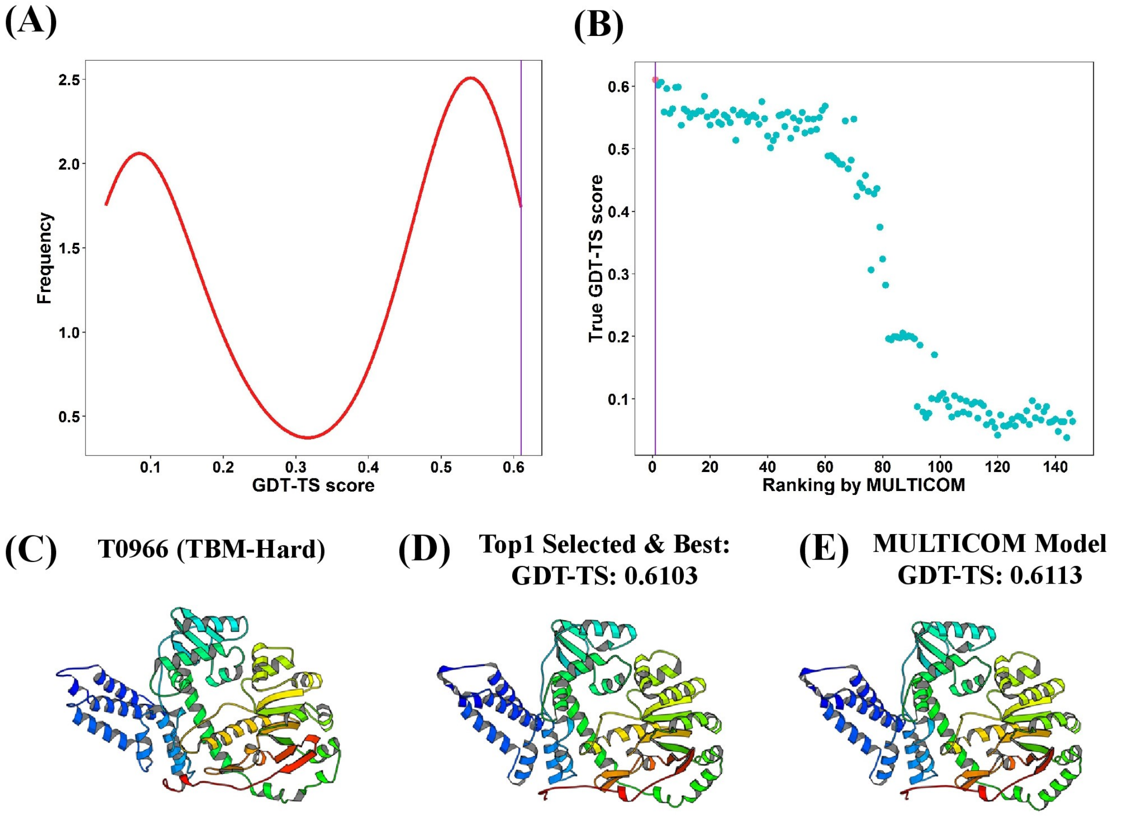


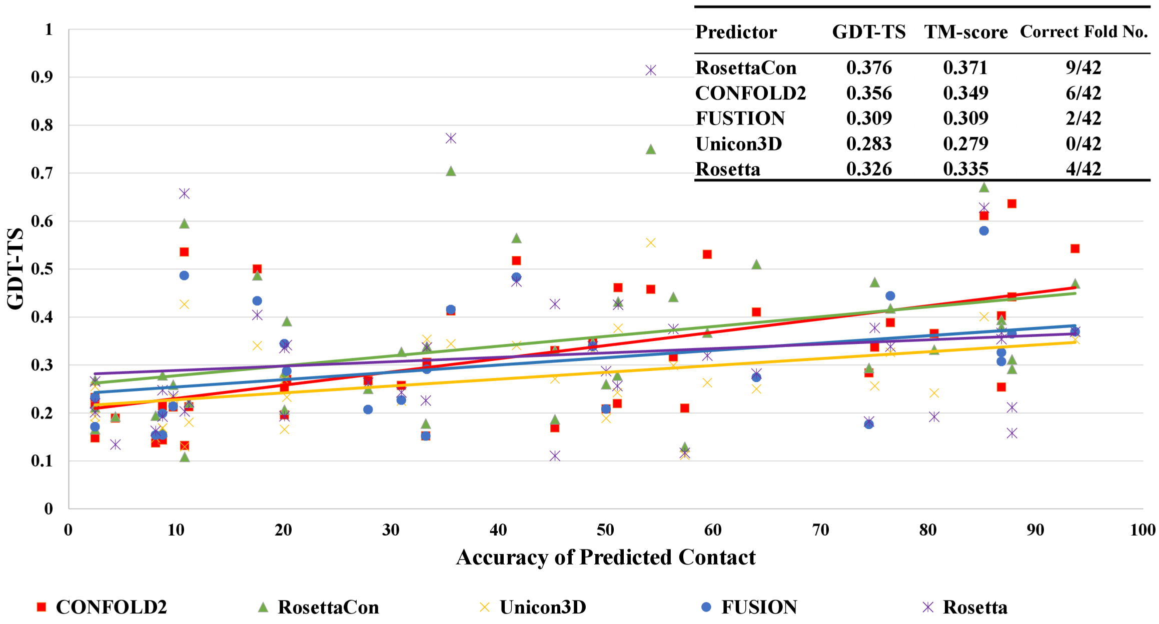
Figure 5.7: Tertiary structure prediction for T0966. (A) The distribution of GDT-TS scores of 146 server models. (B) The plot of the true GDT-TS scores of models against their predicted ranking by MULTICOM. The point highlighted in red is the top model selected by DeepRank. (C) The native structure of target T0966 (PDB code: 5w6l). (D) The top selected model. (E) The final first MULTICOM model (TS1).

5.4.3 Comparison of different contact-based ab initio modeling methods on FM targets

To evaluate how predicted contact distances improved template-free modeling, we collected the top 5 models predicted by five ab initio modeling methods (CONFOLD2, RosettaCon - Rosetta with contacts, UniCon3D with contacts, FUSION with contacts, and Rosetta without contacts) for all domains that MULTICOM considered them as ”hard”. **Figure 5.8** shows that the GDT-TS scores of the ab initio models generally increase as the accuracy of contact prediction becomes higher for each method. This upward trend is most significant for CONFOLD2 and the correlation between the contact accuracy and the GDT-TS score of CONFOLD2 models is 0.578. This is expected because CONFOLD2 is the only pure contact distance-driven modeling method in the group and contact distances play a direct and dominant role in its modeling, while they only play an indirect role in the other three modeling methods assisted by contact predictions.

The average GDT-TS score and TM-score were also calculated for each method on the free-modeling targets. The models generated by RosettaCon has the highest average GDT-TS score of 0.376 and CONFOLD2 has the second highest average score of 0.356, followed by Rosetta, FUSION, and UniCon3D. It is interesting to note that CONFOLD2 started to work better than RosettaCon when top L/5 contact predictions reached a high accuracy (e.g., 80%). When the accuracy of contact prediction was lower, RosettaCon worked somewhat better than CONFOLD2 probably because the extra structural fragment information and its advanced energy function made some difference. The comparison of RosettaCon and Rosetta shows a 15.3% increase of GDT-TS score by using contact distance restraints, demonstrating that predicted contacts can significantly improve the fragment-assembly modeling.

Figure 5.9 shows a successful ab initio modeling example (a domain of target T1000) for which no significant templates were identified. For the FM domain of T1000 (residues 282-523), the accuracy of top L/5 predicted contacts is 100%, top L 79% and top 2L 50%. CONFOLD2 successfully built a complicated α-helix+β-sheet+α-helix model for the domain with TM-score of 0.8 and GDT-TS of 0.64, while RosettaCon failed to generate a correct topology (i.e., TM-score = 0.33 < 0.5 threshold). This example shows that the pure contact distance driven method such as CONFOLD2 can build high-quality structural models of complicated topology for large domains if a sufficient number of accurate contact predictions are provided.



Predictor	GDT-TS	TM-score	Correct Fold No.
RosettaCon	0.376	0.371	9/42
CONFOLD2	0.356	0.349	6/42
FUSTION	0.309	0.309	2/42
Unicon3D	0.283	0.279	0/42
Rosetta	0.326	0.335	4/42

Figure 5.8: The modeling performance of contact-based ab initio modeling methods versus the predicted contact accuracy (L/5 contacts) in CASP13. Each point represents the modeling accuracy in terms of GDT-TS score versus the accuracy of predicted contacts for a method. The colors represent different modeling methods. Rosetta without contacts (purple) was included for comparison. The averaged GDT-TS score and TM-score of five methods on the all CASP13 targets are summarized in the top-right table.

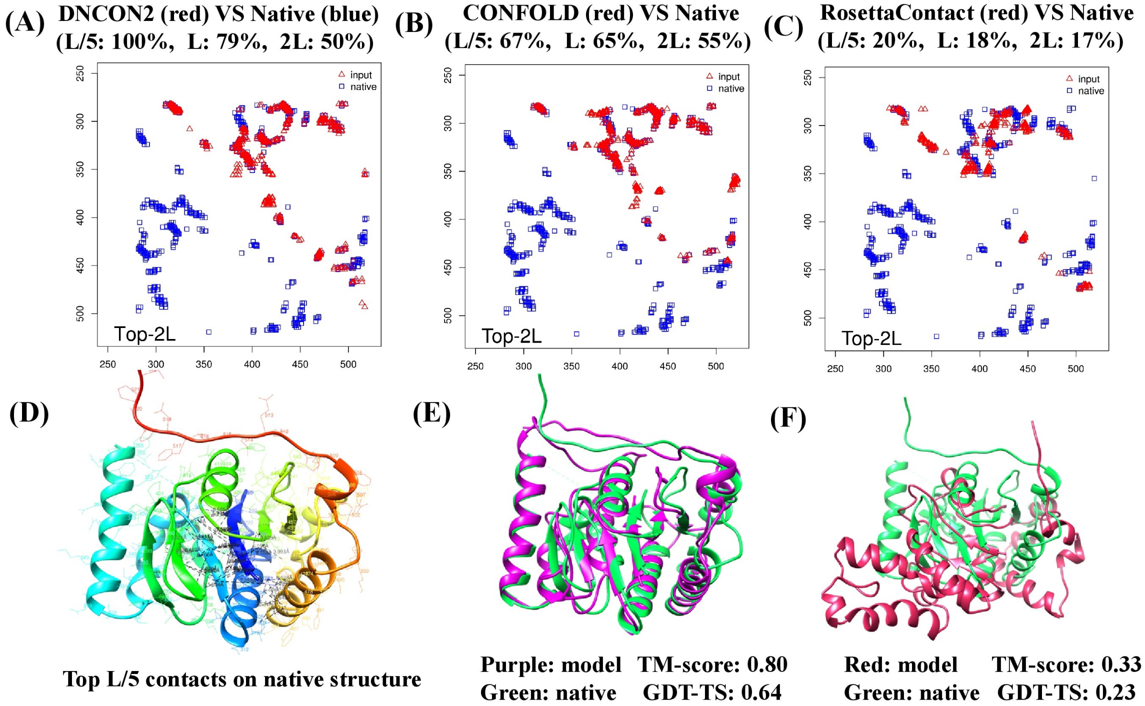


Figure 5.9: A successful ab initio modeling example (a domain of target T1000) for which no significant templates were identified. For the FM domain of T1000 (residues 282-523), the accuracy of top L/5 predicted contacts is 100%, top L 79% and top 2L 50%. CONFOLD2 successfully built a complicated α-helix+β-sheet+α-helix model for the domain with TM-score of 0.8 and GDT-TS of 0.64, while RosettaCon failed to generate a correct topology (i.e., TM-score = $0.33 < 0.5$ threshold). This example shows that the pure contact distance driven method such as CONFOLD2 can build high-quality structural models of complicated topology for large domains if a sufficient number of accurate contact predictions are provided.

5.4.4 Impact of domain parsing on structure prediction and model ranking

Protein domain identification is an important component in the MULTICOM predictors. When a target protein sequence was searched against a template library, the domain regions that were homologous to templates were marked as ”template-based” and modeled by the template-based modeling protocol. The unmarked regions were modeled by the contact distance-based ab initio modeling methods. The domain models were evaluated using the three QA methods and top models were assembled into full-length structures as final predictions. For the human predictor, the domain boundaries might be re-analyzed by taking the structural information of top ranked server models into account. We assessed the impact of domain parsing on the structure prediction of the CASP13 targets that were predicted as multi-domain proteins. The final predicted models of these multi-domain targets and the models without domain parsing were evaluated and compared according to the official domain definitions of CASP13. Among the 90 CASP13 targets, 31 targets were modeled as multi-domain by MULTICOM server predictors and 19 targets by MULTICOM human predictor. Supplemental Table S7 reports the scores of the models using or not using domain parsing. For the server predictors, the performance of structure prediction was substantially improved in terms of GDT-TS, TM-score and RMSD after the domain-based modeling was applied. For the human predictor, the quality of final predictions was also slightly improved when domain information was considered. And almost all the improvement is significant.

5.4.5 What went right?

In CASP13, a main progress was to apply contact distance prediction and deep learning to improve ab initio modeling. Predicted contacts were successfully utilized to guide ab initio structure modeling for several hard targets that could never be modeled correctly before. Supplemental Figure S12 shows the models and scores of nine hard targets that were folded into correct topology when the predicted contacts generated by DNCON2 were rather accurate. Remarkably, a pure contact distance-driven modeling method - CONFOLD2 can correctly predict complex folds of large domains if a sufficient amount of accurate contact distance predictions is provided. Furthermore, the inter-residue distance distribution predicted by DNCON2 (e.g., 6 Å, 7.5 Å, 8 Å, 8.5 Å and 10 Å) is valuable for structure prediction, demonstrated by the fact that it helped improve the accuracy of final top L/5 contact predictions from 57.11% to 61.97% on CASP13 targets (supplemental Figure S13). Another main progress is that MULTICOM performed better in ranking the models in CASP13 than in CASP12 due to the application of deep learning and contact prediction. MULTICOM successfully selected models that are identical or close to the best models for 28 targets (see the distribution of loss of model selection for all the targets and two good examples in supplemental Figure S14).

Moreover, we successfully tested a new heuristic method to apply domain-based contact predictions to validate multi-domain template-based models. One such example is T0996, a challenging template-based modeling target due to its very large size and very weak homology with existing templates (**Figure 5.10**). It was recognized by CASP13 as hard template-based target because only several weak partial templates (e.g., PDB code: 5UW2, chain A) could be detected. MULTICOM server predictors successfully divided T0996 into 7 domains and the predicted domain boundaries were largely accurate compared to the official domain definition. Each domain region was modeled through MULTICOM domain-based modeling pipeline. After the domain models were assembled, the full-length structural model was evaluated by the predicted contacts using ConEva [167]. The contacts in the model matched well with the contacts predicted by DNCON2 domain by domain, confirming that both domain parsing and structure modeling was largely correct (**Figure 5.10**). This contact-based validation approach was applied to all CASP13 targets during CASP13, providing a complementary validation for structure modeling.

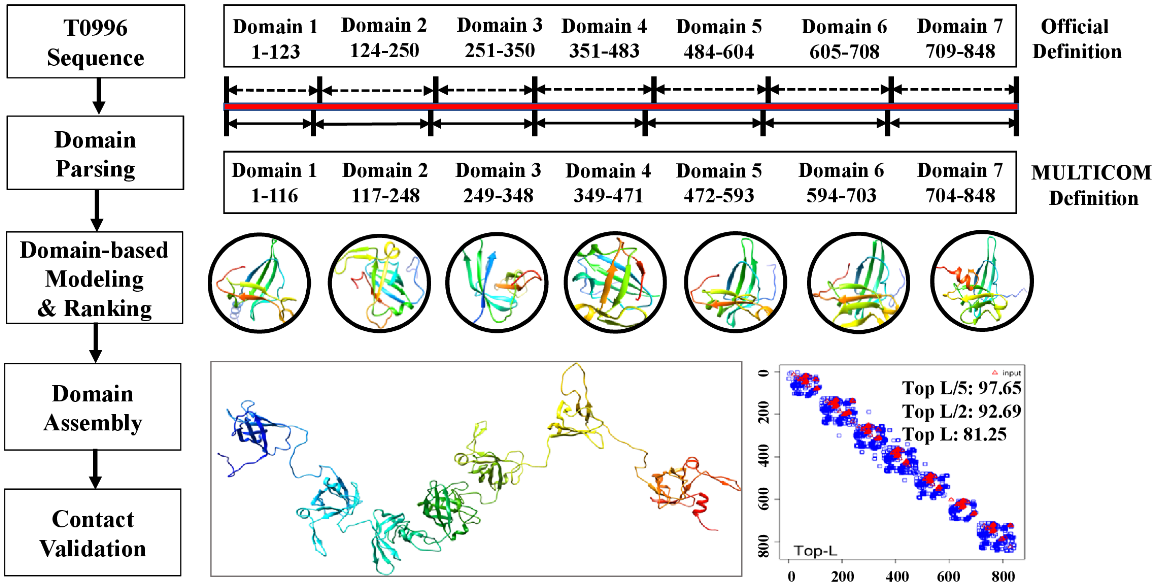


Figure 5.10: The successful modeling of a large multi-domain target T0996 and the contact-based validation. The contacts (red) predicted by DNCON2 match with the contacts (blue) in the template-based models domain by domain.

5.4.6 What went wrong?

Despite the significant progress of MULTICOM in CASP13, it has its several limitations. The first limitation is in contact distance prediction. DNCON2 sometime failed to generate a sufficient amount of accurate contact predictions to predict correct folds. The problem is particularly severe when the number of effective homologous sequences for a target is small (see supplemental Figure S15 for an example - T0998). One possible reason is that it did not use a metagenomics sequence database [168] that contains sequences not present in the non-redundant protein sequence database and the latest HHblits database [15] to collect homologous sequences. Another possible reason is the convolutional architecture used by DNCON2 is not deep enough in comparison with some other approaches [53, 142, 9]. The second limitation is that only the coarse distance restraints derived from binary contacts at 8 Å threshold were used with CONFOLD2 for ab initio modeling, without taking advantage of the more detailed distance distribution spanning multiple distance thresholds predicted by DNCON2, which limited its capability to build quality models [169].

The third limitation is that the deep learning-based quality assessment failed on some targets. As shown in supplemental Figure S14 (B), DeepRank method performed poorly with loss > 0.1 on 14 ”all groups” targets. The failed rankings are summarized in supplemental Table S8 and Figure S16-S29. The results show that its performance was worse on the free-modeling targets or hard-template targets than on other targets. A possible reason is that a large portion of low-quality models in the pool and less accurate features of measuring model quality (e.g., contact predictions) for the hard targets hinders the performance of the deep learning ranking.

5.5 Conclusion and Future Work

Our CASP13 results demonstrate that residue-residue contact prediction, more generally distance prediction, is the key direction to advance protein structure prediction, particularly ab initio prediction, and deep learning is the key technology to solve it. Not only do accurate contact distance prediction and deep learning enhance ab initio structure folding, but also model ranking for both template-based and free modeling. In the future, we will develop more advanced deep learning methods to directly predict real-value distances between residues and/or classify them into much finer intervals than DNCON2 currently does. The more detailed distance predictions will be used to more accurately fold proteins by the distance geometry [104, 10], simulated annealing and advanced gradient descent optimization [170, 171] as well as to rank protein models.

5.6 The MULTICOM protein structure prediction server empowered by deep learning and contact distance prediction

Prediction of the three-dimensional (3D) structure of a protein from its sequence is important for studying its biological function. With the advancement in deep learning contact distance prediction and residue-residue co-evolutionary analysis, significant progress have been made in both template-based and template-free protein structure prediction in the last several years. Here, we provide a practical guide for our latest MULTICOM protein structure prediction system built on top of the latest advances rigorously tested in the 2018 CASP13 experiment. Its specific functionalities include: (1) prediction of 1D structural features (secondary structure, solvent accessibility, disordered regions) and 2D inter-residue contacts; (2) domain boundary prediction; (3) template-based (or homology) 3D structure modeling; (4), contact distance-driven ab initio 3D structure modeling; and (5) large-scale protein quality assessment enhanced by deep learning and predicted contacts. The MULTICOM web server (http://sysbio.rnet.missouri.edu/multicom_cluster/) presents all the 1D, 2D and 3D prediction results and quality assessment to users via user-friendly web interfaces and emails. The source code of the MULTICOM package is also available at https://github.com/multicom-toolbox/multicom.

The MULTICOM server was blindly tested in 2018 CASP13 experiment and was ranked among top 10 servers. Compared with the existing servers such as I-TASSER [172] and ROSETTA [6], MULTICOM generate a more comprehensive set of predictions ranging from 1D features (secondary structures, solvent accessibility, disorder regions, and domain boundaries), 2D inter-residue contact features, 3D structures and templates, to the state-of-the-art quality assessment. These predictions such as 2D contact maps and 3D models are visualized in a user-friendly format. The cross-validation between 2D predicted contact maps and 3D models is unique. The ab initio modeling driven by contact distance prediction is also different from the fragment assembly approach used in I-TASSER and ROSETTA servers. Therefore, the MULTICOM server provides a unique, versatile tool for the community to predict protein structures.

5.6.1 Materials

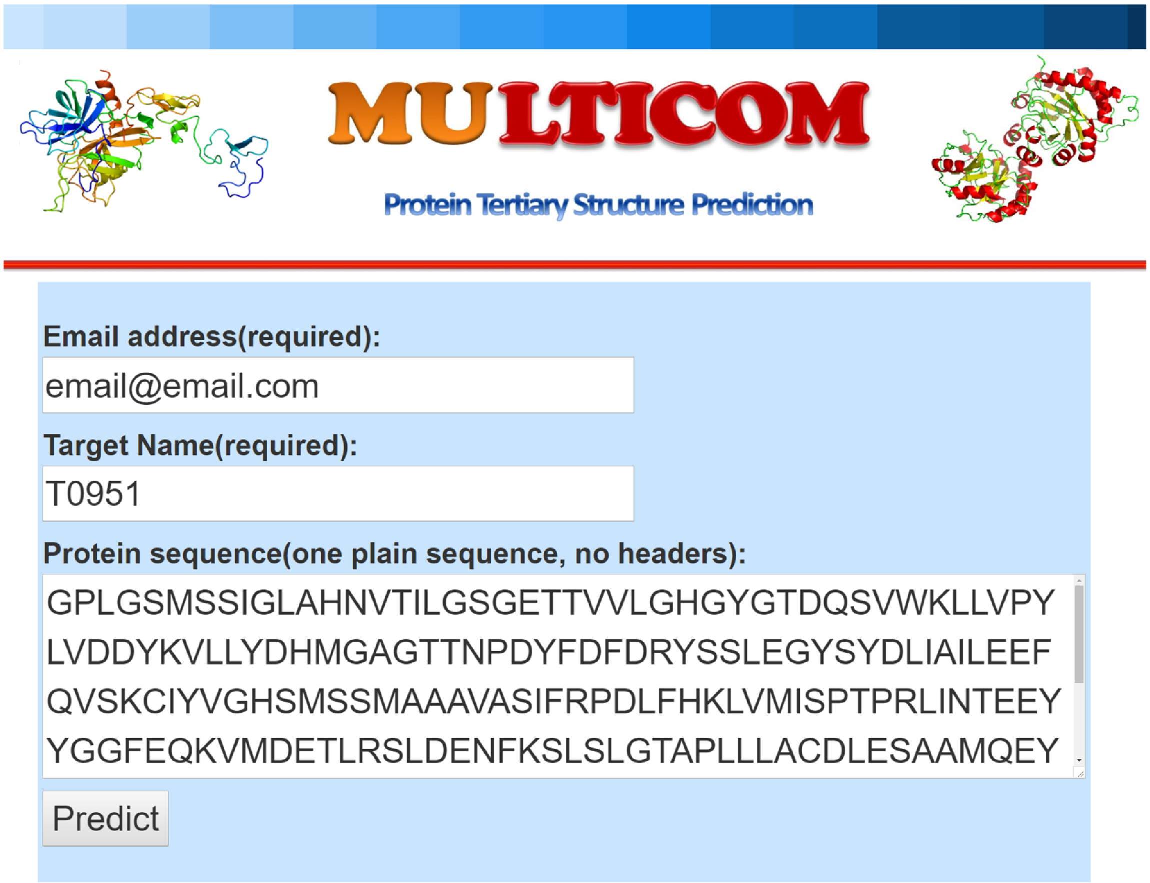
Input

Three types of information are required by the MULTICOM web server for protein structure prediction: (1) target name; (2) user's email address; and (3) one single-lettered protein sequence. The target name identifies the job being submitted. The prediction results will be sent to the user's email address once the task is finished. The protein sequence should be composed of 20 standard amino acids. **Figure 5.11** shows an input example (CASP13 target "T0951"). All data in the input fields, including the email address, target name and protein sequence should be verified by users before clicking on the 'predict' button.

Output

After the job is completed, the user receives two types of results through email: (1) top 5 predicted protein structures with detailed atomic coordinates; and (2) a unique web link for detailed results with visualization.

(1) The structure file attached in the email is in the standard Protein Data Bank (PDB) textual file format, containing the atomic coordinates (i.e., x, y, z) of each atom in the protein (`http://predictioncenter.org/casp13/index.cgi?page=format`). The PDB file can be visualized using any viewer tools such as Chimera [173], PyMOL [174], Rasmol [175], and Jmol (Jmol: an open-source Java viewer for chemical structures in 3D. `http://www.jmol.org/`).



Email address(required): email@email.com

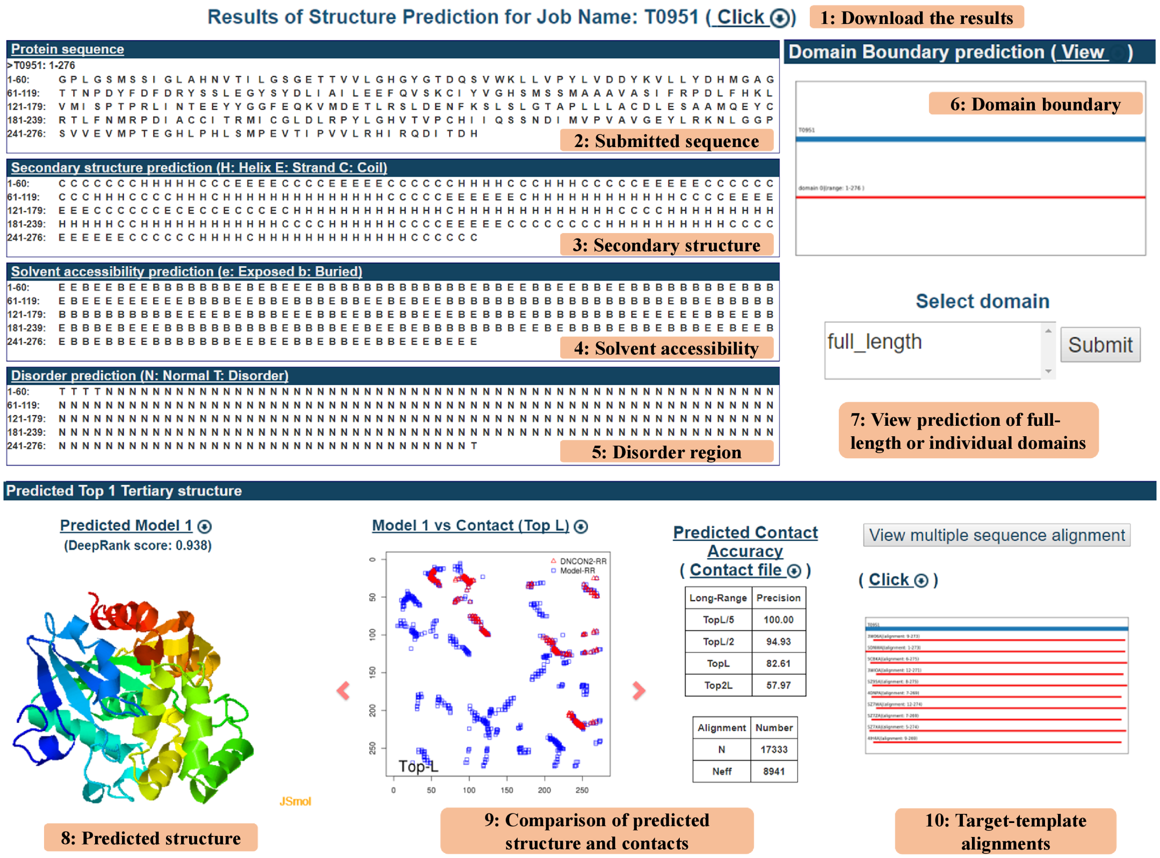
Target Name(required): T0951

Protein sequence(one plain sequence, no headers):

```
GPLGSMSSIGLAHNVTILGSGETTVVLGHGYGTDQSVWKLLVPY
LVDDYKVLLYDHMGAGTTNPDYFDFDRYSSLEGYSYDLIAILEEF
QVSKCIYVGHSMSSMAAAVASIFRPDLFHKLVMISPTPRLINTEEY
YGGFEQKVMDETLRSLDENFKSLSLGTAPLLLACDLESAAMQEY
```

Figure 5.11: The input web page of MULTICOM web server.

(2) The user will also receive one unique web link associated with the job identifier that the user provided. JavaScript enabled in the web browser is required to view the 3D structures in the web page. The recommended browsers are: Google Chrome, FireFox, Safari or Internet Explorer. Several predicted protein features are presented, including predicted secondary structure, solvent accessibility, disorder regions and predicted domain boundaries. The top 5 predicted structures, and their match with the predicted contact in terms of top L, top L/5, top L/2 and top 2L long-range contacts (see **Note 1**) are also visualized. **Figure 5.12** shows an example of the detailed results for Target "T0951". More details will be described in the Method section.



Results of Structure Prediction for Job Name: T0951 (Click) 1: Download the results

Protein sequence

```
>T0951: 1-276
1-60:     G P L G S M S S I G L A H N V T I L G S G E T T V V L G H G Y G T D Q S V W K L L V P Y L V D D Y K V L L Y D H M G A G
61-119:   T T N P D Y F D F D R Y S S L E G Y S Y D L I A I L E E F Q V S K C I Y V G H S M S S M A A A V A S I F R P D L F H K L
121-179:  V M I S P T P R L I N T E E Y Y G G F E Q K V M D E T L R S L D E N F K S L S L G T A P L L L A C D L E S A A M Q E Y C
181-239:  R T L F N M R P D I A C C I T R M I C G L D L R P Y L G H V T V P C H I I Q S S N D I M V P V A V G E Y L R K N L G G P
241-276:  S V V E V M P T E G H L P H L S M P E V T I P V V L R H I R Q D I T D H
```

2: Submitted sequence

Secondary structure prediction (H: Helix E: Strand C: Coil)

```
1-60:     C C C C C C C H H H H H C C C E E E E E C C C C E E E E E C C C C C C H H H H C C C H H H C C C C C E E E E E C C C C C C
61-119:   C C C H H H C C C C H H H C H H H H H H H H H H H H H H C C C C C E E E E E E C H H H H H H H H H H H H H C C C C E E E E
121-179:  E E E C C C C C C E C E C C E C C C E C H H H H H H H H H H H H H C H H H H H H H H H H H H H H C C C C H H H H H H H H H
181-239:  H H H H H C C H H H H H H H H H H H H H C C C C H H H H H H C C C C E E E E E C C C C C C C C H H H H H H H H H H C C C C
241-276:  E E E E E E C C C C C C H H H H C H H H H H H H H H H H H H H C C C C C C
```

3: Secondary structure

Solvent accessibility prediction (e: Exposed b: Buried)

```
1-60:     E E B E E B E E B B B B B B B E B E B E E E B B B B B B B B B B B B E B B E E B B E E B B E E B B B B B B B B E B B B
61-119:   E B E E E E E E E E E B B B B E E B B E B B B E B B E E B E B E E B B B B B B B B B B B B B B B B B E B B E B B B B B
121-179:  B B B B B B B B B B E E E E B E B B B E E E B B E E B B E B B E E E B E B B B E B B B B B B B B E E E E E E B B E E B B
181-239:  E B B B E B E E E B B B B B B B B B B B E B B B B E E B E E B E B B B B B B B B B E E B E B B B B E B B E B B B E E B E E B
241-276:  E B B E B B E E B B B B B B B E E B E E B B E B B E E B B E E E B E E E
```

4: Solvent accessibility

Disorder prediction (N: Normal T: Disorder)

```
1-60:     T T T T N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N
61-119:   N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N
121-179:  N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N
181-239:  N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N
241-276:  N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N T
```

5: Disorder region

Domain Boundary prediction (View)

6: Domain boundary

Select domain

full_length

Submit

7: View prediction of full-length or individual domains

Predicted Top 1 Tertiary structure

Predicted Model 1

(DeepRank score: 0.938)

8: Predicted structure

Model 1 vs Contact (Top L)

9: Comparison of predicted structure and contacts

Predicted Contact Accuracy

(Contact file)

Long-Range	Precision
TopL/5	100.00
TopL/2	94.93
TopL	82.61
Top2L	57.97

Alignment	Number
N	17333
Neff	8941

View multiple sequence alignment

(Click)

10: Target-template alignments

Figure 5.12: The MULTICOM web server's prediction for CASP13 target "T0951". The brown boxes denote the annotations of 10 different kinds of contents.

Availability

The MULTICOM web server is freely available at `http://sysbio.rnet.missouri.edu/multicom_cluster/`. The source code and tool packages are available at `https://github.com/multicom-toolbox/multicom`. Prediction time depends on several factors, including server load, length of the input sequence, and difficulty of the query (i.e., whether good templates can be found).

5.6.2 Methods

This section provides a step-by-step tutorial on how to use the MULTICOM server for protein structure prediction and how to interpret the predicted results.

Submit the sequence

1. Open a web browser such as Google Chrome and type the address `http://sysbio.rnet.missouri.edu/multicom_cluster/`. User will be taken to the homepage as shown in **Figure 5.11**.

2. In the section 'Email address', input the e-mail address that the results will be sent to.

3. In the section 'Target name', input the name for the protein sequence. A duplicate name can be accepted in case the user wants to reproduce the predictions. We recommend a target name with a short length.

4. In the 'Protein sequence' section, enter a protein sequence by copying the query sequence to the textbox. Non-standard amino acids (i.e., J, O, B, U) and any special characters (i.e., $, *) or white space characters will be removed from the sequence automatically. Both upper or lower-case letters of protein sequence are accepted and lower-case letters will be converted to upper-case automatically.

5. Press the 'Predict' button to submit a job. Once the job is received, the user will receive a confirmation email with the subject 'Job submission to MULTICOM'. The email includes the result link that the user can use to check the prediction results. The home page will also be directed to the waiting status page and the result will be shown once the job is completed. The user will also be notified through email when the job is completed. It may take hours or even longer for the results to be ready.

Acquire the predictions

Once the server completes the prediction, the results link will be sent to the corresponding e-mail address. The user can click the link and view/download the predicted results for the input sequence, as shown in **Figure 5.12**. The details of results are summarized as follows:

1. The entire predicted results can be downloaded as a file from the link shown in **Box 1** in **Figure 5.12**.

2. The predicted secondary structure, solvent accessibility and disordered regions for the input sequence are provided in **Box 2, 3, 4** and **5**. The secondary structure and solvent accessibility are predicted by SSpro/ACCpro [16], showing the putative 3-state secondary structure for each residue in the protein sequence, including alpha-helix (H), beta-strand (E) and coil (C). The disorder region is predicted by PreDisorder [17]. The disordered residues are marked as T, while the ordered residues are marked as N.

3. The predicted domain boundary in the protein sequence is visualized in **Box 6**. The domain boundary is parsed from the target-template sequence alignments. If the protein is identified as a multi-domain protein, the user can select a specific domain for detailed results that are shown in **Box 7** in **Figure 5.13**, otherwise,

the link for full-length results will be shown in **Box 7** in **Figure 5.12**. The predictions for each individual domain will be reported, including the template information, predicted contact maps and predicted domain structures, as shown in **Figure 5.13**.

4. In the 3D prediction section, the predicted tertiary structures by MULTICOM are visualized in the JSmol viewer, as shown in **Box 8**. The predicted structure can be viewed in 3D orientation by moving the mouse pointer to the JSmol screen and holding down the left-click mouse. More options are available by right-clicking the mouse including downloading the structure file or changing the visualization configuration. More detailed information for using JSmol can be found in `http://wiki.jmol.org/index.php/Main_Page`. The structural quality predicted by our quality assessment method, DeepRank [4], is provided along with the tertiary structure.

5. The predicted tertiary structure will be cross-validated by the predicted contacts using ConEva [167]. The match between predicted tertiary structures and predicted contacts made by deep learning is visualized in **Box 9**. The user can slide the window to view the comparison of top L/5, top L/2, top L, top 2L predicted long-range contacts (i.e., sequence distance ≥ 24) one by one (see **Note 1** for more details). In the contact map, the blue points are the residue contact derived from the predicted structure, and the red points show the contacts predicted from the sequence by deep learning. If the red points and blue points match very well (i.e., high precision), the quality of both tertiary structure predictions and contact predictions is expected to be good. Generally, a larger number of effective sequences in the sequence alignment is an indicator if the contacts are accurately predicted. The contact matching accuracy and the sequence alignment information is also provided for reference.

6. If the homologous templates are identified and used for structure modeling, the alignments between target protein and templates are reported in the **Box 10**. The image shows the coverage of the templates aligned with the target protein. The detailed alignments can be viewed by clicking the button 'View multiple sequence alignment'.

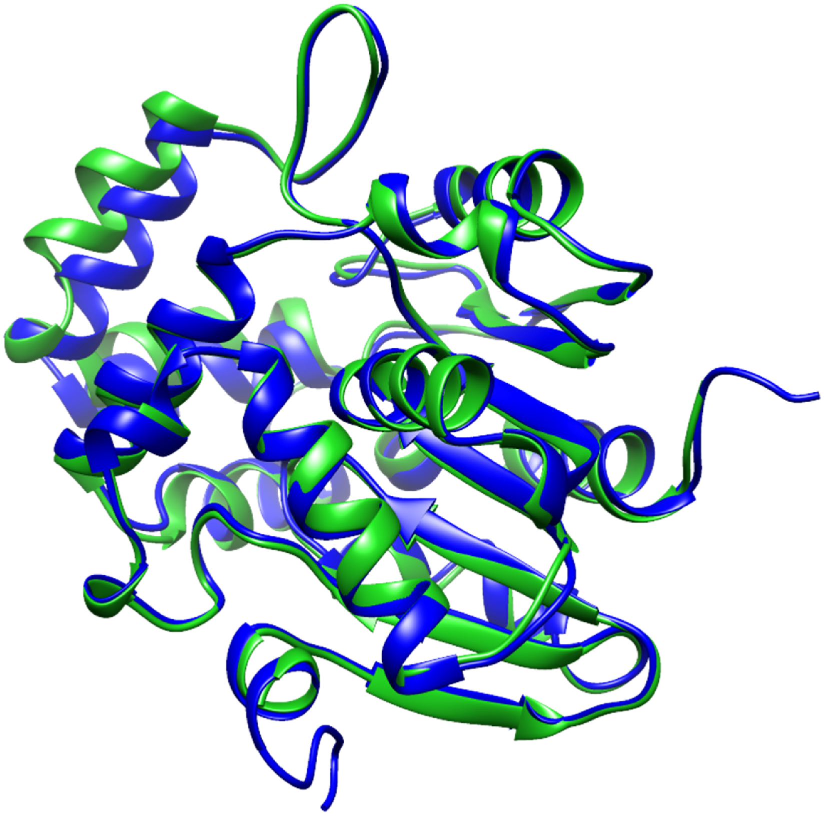


Figure 5.13: The native structure (shown in green color) and MULTICOM-predicted structure (shown in blue color) superimposed using Chimera for target T0951 (PDB code: 5z82)

5.6.3 Case Studies

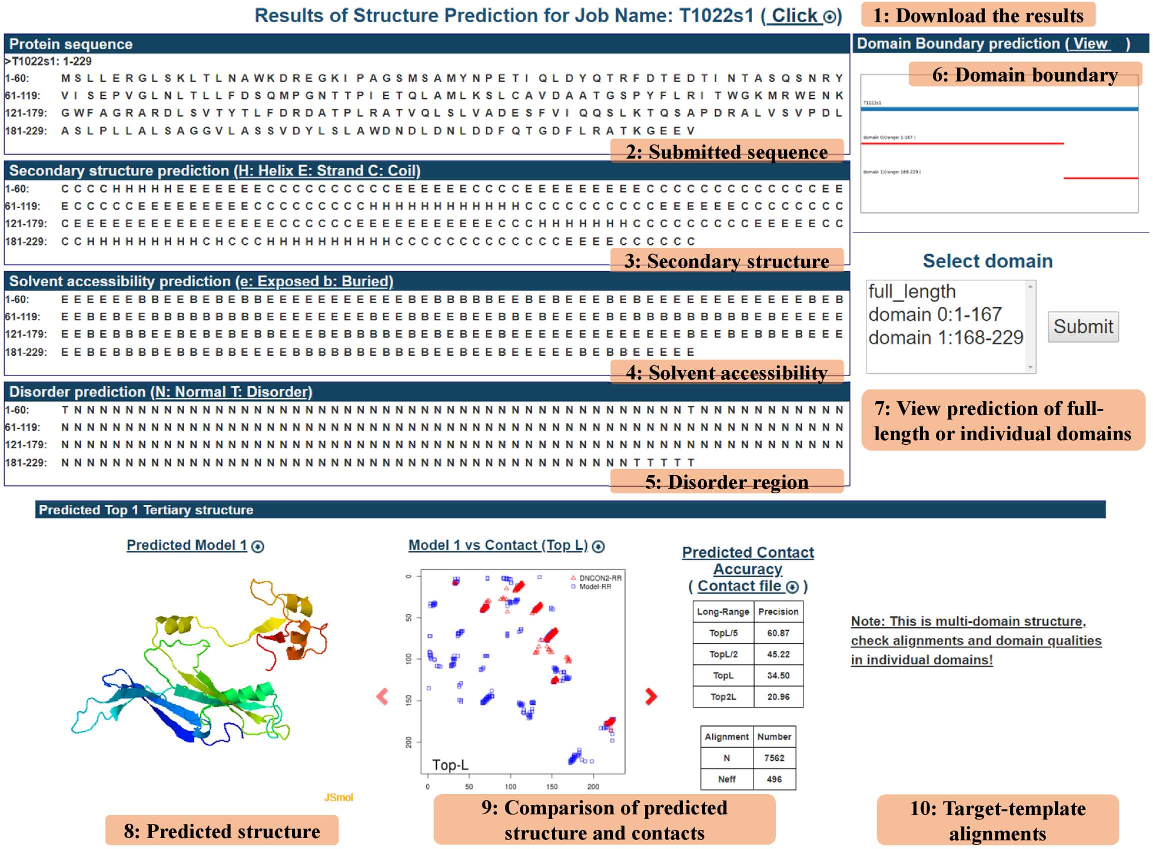
In this section, we will use two cases to illustrate the results that the MULTICOM server can provide. The two examples cover the four categories of protein structure modeling, including single-domain modeling (i.e., T0951), multi-domain modeling (i.e., T1022s1), template-based modeling (i.e., T0951, T1022s1: Domain 1), and template-free domain modeling (i.e., T1022s1: Domain 0).

Single-domain protein (T0951)

The first example is the CASP13 target T0951 (`http://predictioncenter.org/casp13/target.cgi?id=25&view=all`). According to the official domain definitions of CASP13 (`http://predictioncenter.org/casp13/domains_summary.cgi`), T0951 was classified as a single-domain template-based target (see **Note 1** in the Notes Section), and the PDB ID for this protein is 5z82. To predict the structure of T0951, its protein sequence consisting of 276 residues in a single line was copied and supplied as input to the MULTICOM web server as shown in **Figure 5.11**. After providing the target name (i.e., T0951) and email address (i.e., email@email.com), the job was submitted. MULTICOM server accepted the job and started to predict the 3D structure for the target. Once the task was completed, an email was sent to the email address and the results were visualized in the web page (**Figure 5.12**). Based on the results that MULTICOM server provided, the prediction of secondary structure and solvent accessibility is provided in **Box 3 & 4**, and the protein contains disordered regions at the N-terminal and C-terminal (see **Box 5**). MULTICOM identified multiple significant templates (see **Note 2**) for this protein (see **Box 10**) that covered the full-length target sequence, suggesting that the protein was a single-domain protein (or a single modeling unit covered by at least one complete template) (see **Box 6**). The predicted 3D structure was visualized in the **Box 8**. Additionally, the predicted structure was evaluated by the contacts predicted by the deep learning method using ConEva (see **Box 9**). Since the number of effective sequences for the target protein is very high (i.e., 8941), the prediction of contacts can be generally considered as accurate and convincing. If the contacts in the model matched well with the predicted contacts (i.e., the accuracy of long-range top L/5 contacts is 100.0%) (see **Note 3**), the quality of the predicted structure can be also considered as largely correct. MULTICOM also provides the results of the top 5 predicted structures. Compared with the native structure of the target (see **Note 4**), the TM-score and RMSD of the top 1 predicted structure is 0.9772 and 0.967, respectively, indicating that the prediction is accurate. The predicted structure and the native structure are superimposed and visualized in Chimera (**Figure 5.13**).

Multi-domain protein (T1022s1)

The second example is the CASP13 target T1022s1 (`http://predictioncenter.org/casp13/target.cgi?id=185&view=all`). According to the official domain definitions of CASP13 (`http://predictioncenter.org/casp13/domains_summary.cgi`), the target T1022s1 was classified as two-domain protein, where the first domain is a free-modeling (FM) domain (position: 1-157) and the second domain is a template-based modeling (TBM) domain (position: 158-224) (see **Note 1**). To predict the structure of the protein target T1022s1, its protein sequence of 229 residues in a single line was copied and supplied as input to the MULTICOM web server. After providing the target name (i.e., T1022s1) and email address (i.e., email@email.com), the job was submitted to MULTICOM server. Once the task was completed, the results link was sent to the user's email and the results were visualized in the web page as shown in **Figure 5.14**. Based on the results that MULTICOM server provided, the protein was predicted as two-domain protein (see **Box 6**), where the first domain was predicted from the position 1-167 and the second domain ranged from 168-229, which is largely correct. MULTICOM predicted the structures of the two domains individually. The detailed results for each domain can be viewed through the **Box 7**. For instance, the predicted structures of the first domain were visualized in the **Figure 5.15**. For this domain, MULTICOM treated it as a "hard" domain since no significant templates were identified. The structure was predicted using contact distance-based ab initio modeling methods (i.e., CONFOLD2). Similar to the full-length predictions, the predicted structure of the domain was evaluated by the predicted contacts using ConEva. For the second domain, as shown in **Figure 5.16**, MULTICOM predicted the structure of this domain using template-based modeling approaches because the significant template for this domain was found. The good match between the contacts derived from the predicted structure and the predicted contacts by deep learning also suggests that the prediction is reasonable because the good accuracy of predicted contacts was expected due to a large number of effective sequence (i.e., 1808). Finally, the structures of two domains were combined into a full-length structure which was visualized as **Box 8** in the **Figure 5.14**.



Results of Structure Prediction for Job Name: T1022s1 (Click)

Protein sequence

```
>T1022s1: 1-229
1-60:     M S L L E R G L S K L T L N A W K D R E G K I P A G S M S A M Y N P E T I Q L D Y Q T R F D T E D T I N T A S Q S N R Y
61-119:   V I S E P V G L N L T L L F D S Q M P G N T T P I E T Q L A M L K S L C A V D A A T G S P Y F L R I T W G K M R W E N K
121-179:  G W F A G R A R D L S V T Y T L F D R D A T P L R A T V Q L S L V A D E S F V I Q Q S L K T Q S A P D R A L V S V P D L
181-229:  A S L P L L A L S A G G V L A S S V D Y L S L A W D N D L D N L D D F Q T G D F L R A T K G E E V
```

Secondary structure prediction (H: Helix E: Strand C: Coil)

```
1-60:     C C C C H H H H H E E E E E E E C C C C C C C C C E E E E E E C C C C E E E E E E E E C C C C C C C C C C C C E E
61-119:   E C C C C C E E E E E E E E C C C C C C C C C H H H H H H H H H H H C C C C C C C C C C E E E E E E C C C C C C C C
121-179:  C E E E E E E E E E E E E E E E E C C C C C C E E E E E E E E E E E C C C H H H H H H C C C C C C C C C E E E E E C C
181-229:  C C H H H H H H H H C H C C C H H H H H H H H H C C C C C C C C C C C C E E E E C C C C C C
```

Solvent accessibility prediction (e: Exposed b: Buried)

```
1-60:     E E E E E E B B E E B E B E B B E E E E E E E E E E B E B B B B B E E B E B E E E B E B E E E E E E E E E E E B E B
61-119:   E E B E B E E B B B B B B B B E E E E E E E E E B E E B B E E B E E B B E B E E E E B E B B B B B B B B B E B E E E E E
121-179:  E E B E B B B E E B E B E B E B B E E E B E B B B B E B E B E B E E B E E E E E E E E E E E E B E E B E B B E B E E E
181-229:  E E B E B B B B E B B E B B E E E E B B B B B B E B B E B E E B E E B E E E E E B E B B E E E E E
```

Disorder prediction (N: Normal T: Disorder)

```
1-60:     T N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N T N N N N N N N N N N N
61-119:   N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N
121-179:  N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N
181-229:  N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N N T T T T T
```

Domain Boundary prediction (View)

Select domain

full_length
domain 0:1-167
domain 1:168-229

Predicted Top 1 Tertiary structure

Predicted Model 1

Model 1 vs Contact (Top L)

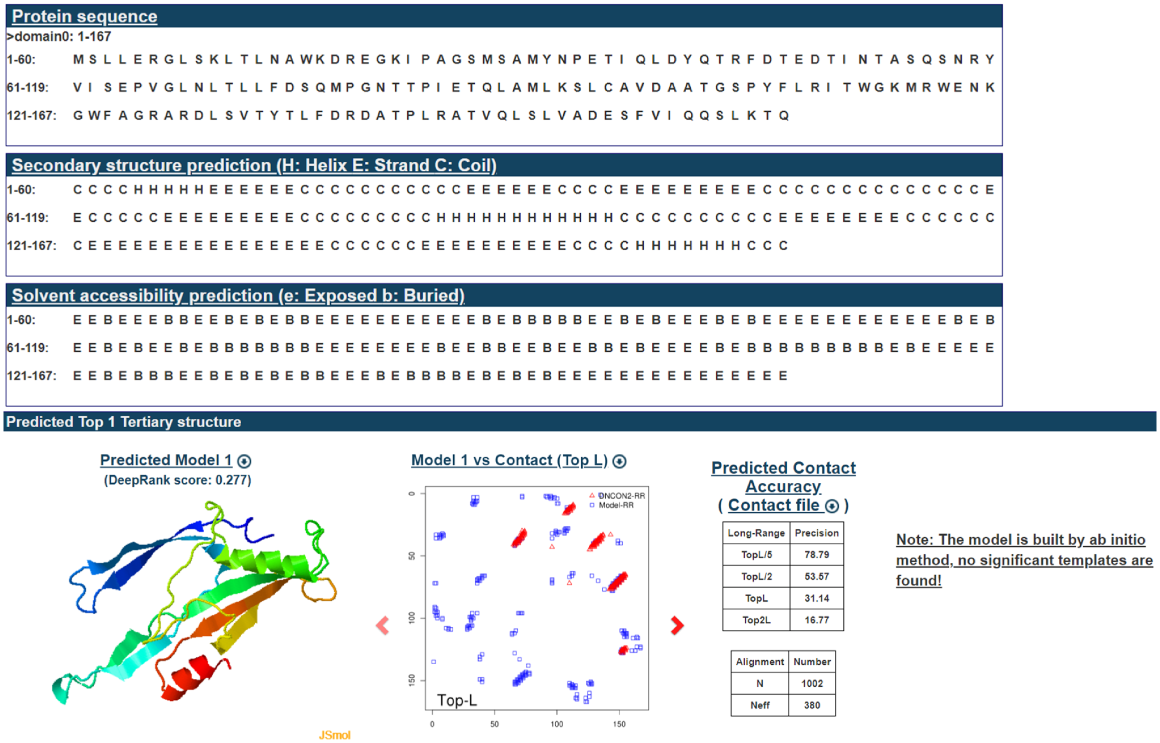
Predicted Contact Accuracy (Contact file)

Long-Range	Precision
TopL/5	60.87
TopL/2	45.22
TopL	34.50
Top2L	20.96

Alignment	Number
N	7562
Neff	496

Note: This is multi-domain structure, check alignments and domain qualities in individual domains!

Figure 5.14: The MULTICOM web server's prediction for CASP13 target T1022s1. The orange boxes annotate different prediction results. The target was predicted to have two domains.



Protein sequence

>domain0: 1-167

```
1-60:     M S L L E R G L S K L T L N A W K D R E G K I P A G S M S A M Y N P E T I Q L D Y Q T R F D T E D T I N T A S Q S N R Y
61-119:   V I S E P V G L N L T L L F D S Q M P G N T T P I E T Q L A M L K S L C A V D A A T G S P Y F L R I T W G K M R W E N K
121-167:  G W F A G R A R D L S V T Y T L F D R D A T P L R A T V Q L S L V A D E S F V I Q Q S L K T Q
```

Secondary structure prediction (H: Helix E: Strand C: Coil)

```
1-60:     C C C C H H H H H E E E E E E C C C C C C C C C C C E E E E E E C C C C E E E E E E E E E C C C C C C C C C C C C C C E
61-119:   E C C C C C E E E E E E E E E C C C C C C C C C H H H H H H H H H H H H C C C C C C C C C C E E E E E E E E C C C C C C
121-167:  C E E E E E E E E E E E E E E E E C C C C C C E E E E E E E E E E C C C C H H H H H H H C C C
```

Solvent accessibility prediction (e: Exposed b: Buried)

```
1-60:     E E B E E E B B E E B E B E B B E E E E E E E E E E E B E B B B B B E E B E B E E E B E B E E E E E E E E E E E B E B
61-119:   E E B E B E E B E B B B B B B B E E E E E E E E B E E B B E E B E E B B E B E E E E B E B B B B B B B B B E B E E E E E
121-167:  E E B E B B B E E B E B E B E B B E E E B E B B B B E B E B E B E E E E E E E E E E E E E E E
```

Predicted Top 1 Tertiary structure

Predicted Model 1
(DeepRank score: 0.277)

Model 1 vs Contact (Top L)

Predicted Contact Accuracy
(Contact file)

Long-Range	Precision
TopL/5	78.79
TopL/2	53.57
TopL	31.14
Top2L	16.77

Alignment	Number
N	1002
Neff	380

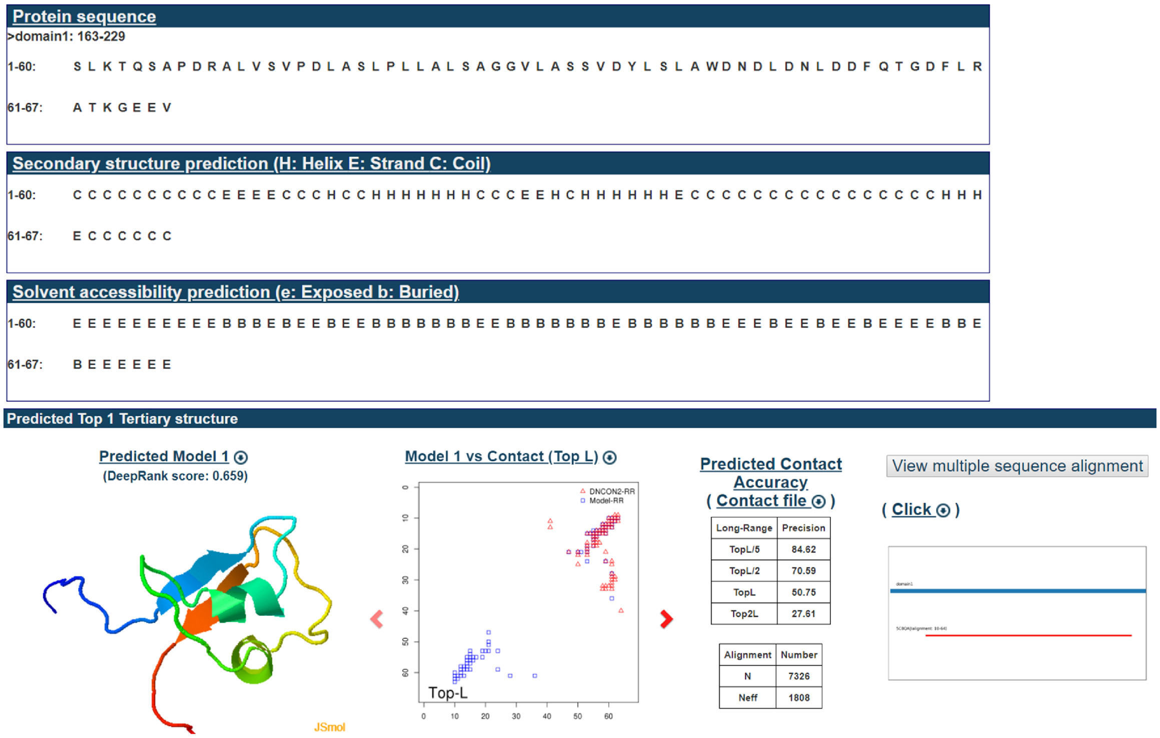
Note: The model is built by ab initio method, no significant templates are found!

Figure 5.15: The MULTICOM web server's prediction for the first domain of T1022s1.

5.6.4 Notes

A pair of residues in sequence is defined to be in contact when the distance between their C atoms (Cα in case of GLY) in the three-dimensional structure is less than 8.0 Å. The contacts with a separation of at least 24 residues along the sequence are defined as 'long-range' contacts. The top L, top L/5, top L/2, and top 2L contacts can be derived when the contact pairs are ranked by the predicted probabilities from the high to low (L is the length of the protein).

The significance of a template against the target sequence is defined by the e-value, which is generated by using an alignment tool like HHsearch [80] to search the query against the template library. Usually, a low e-value means that the template sequence has high similarity to the target sequence. The accuracy of contacts is defined as the percentage of correctly predicted contacts among the selected contacts. Specifically, the accuracy is calculated by the equation $\frac{TP}{TP+FP}$, where the true positives (TP) refers



Protein sequence

>domain1: 163-229

1-60: S L K T Q S A P D R A L V S V P D L A S L P L L A L S A G G V L A S S V D Y L S L A W D N D L D N L D D F Q T G D F L R

61-67: A T K G E E V

Secondary structure prediction (H: Helix E: Strand C: Coil)

1-60: C C C C C C C C C C E E E E C C C H C C H H H H H H C C C E E H C H H H H H E C C C C C C C C C C C C C C C C C H H H

61-67: E C C C C C C

Solvent accessibility prediction (e: Exposed b: Buried)

1-60: E E E E E E E E E E B B B E B E E B E E B B B B B B E E B B B B B B E B B B B B E E E B E E B E E B E E E E B B E

61-67: B E E E E E E

Predicted Top 1 Tertiary structure

Predicted Model 1 ⦿
(DeepRank score: 0.659)

Model 1 vs Contact (Top L) ⦿

Predicted Contact Accuracy
(Contact file ⦿)

Long-Range	Precision
TopL/5	84.62
TopL/2	70.59
TopL	50.75
Top2L	27.61

Alignment	Number
N	7326
Neff	1808

View multiple sequence alignment

(Click ⦿)

Figure 5.16: The MULTICOM web server's prediction for the second domain of T1022s1.

to the predicted contacts that are correct, and false positives (FP) are the incorrectly predicted contacts.

TM-score, RMSD (average root mean square distance between the corresponding Ca atoms), and GDT-TS score are commonly used metrics to compare and evaluate protein structure predictions [132]. The online version of the TM-score tool that can compare the structures of the same protein is available at `http://zhanglab.ccmb.med.umich.edu/TM-score/`. TM-score tool can also be downloaded for local use.

MULTICOM server usually takes 1∼2 days to finish a prediction. The execution time depends not only on the protein size, but also on the computational resources. Currently our server processes up to two sequences at the same time, and extra tasks will be waiting in the queue. The MULTICOM standalone package is also available for local installation, which is recommended if users want to predict structures for a large number of sequences.

Chapter 6

SAXSDom: Modeling multidomain protein structures using small-angle X-ray scattering data

6.1 Abstract

Many proteins are composed of several domains that pack together into a complex tertiary structure. Some multidomain proteins can be challenging for protein structure modeling, particularly those for which templates can be found for the domains but not for the entire sequence. In such cases, homology modeling can generate high quality models of the domains but not for the assembled protein. Small-angle X-ray scattering (SAXS) reports on the solution structural properties of proteins and has the potential for guiding homology modeling of multidomain proteins. In this work, we describe a novel multidomain protein assembly modeling method, SAXSDom, that integrates experimental knowledge from SAXS with probabilistic Input-Output Hidden Markov model (IOHMM). Four SAXS-based scoring functions were developed and tested, and the method was evaluated on multidomain proteins from two public datasets. Incorporation of SAXS information improved the accuracy of domain

assembly for 40 out of 46 CASP multidomain protein targets and 45 out of 73 multidomain protein targets from the AIDA dataset. The results demonstrate that SAXS data can provide useful information to improve the accuracy of domain-domain assembly. The source code and tool package are available at `http://github.com/multicom-toolbox/SAXSDom`. The supplemental material can be found at: `https://www.biorxiv.org/content/10.1101/559617v1.supplementary-material`

6.2 Introduction

Most proteins contain multiple domains. Vogel et al. define a protein domain as an "independent, evolutionary unit that can form a single-domain protein or be part of one or more different multidomain proteins" [176]. Protein domains range in length from about 40 to 500 amino acids, with 100 residues being the most frequent domain length [177, 178]. Obviously, the three-dimensional arrangement of domains within the folded protein - domain architecture - is central to the function of multidomain proteins.

Multidomain proteins present unique challenges to protein structure modeling. The most difficult case occurs when templates can be found only for the domains but not for the entire sequence. In this case, most computational methods adopt a "divide and conquer" strategy in which the sequence is parsed into domains, and the three-dimensional structures of the domains are predicted with either comparative (homology) structure modeling [179, 180] or de novo structure prediction [18, 181] on individual domains. The predicted structures of domains are subsequently assembled into a full-length structural model using a variety of approaches, such as treating the problem as special case of protein-protein docking, [182, 183, 184] using protein folding algorithms to predict the conformation of the linkers between rigid domains, [5, 185] and the use of ab initio folding potentials. [186] Despite these advances, the modeling of multidomain protein structures remains an ongoing area of research.

The use of experimental restraints has the potential to improve the accuracy of predicting multidomain protein structures. Cross-linking/mass spectrometry and small-angle X-ray (SAXS) scattering are two notable examples of experimental methods that provide distance information that can be combined with structure modeling into so-called ”hybrid” methods. [187, 188, 189] In particular, the explosion of biological SAXS over the last 5-10 years [190, 191, 192, 193] suggests that it may be especially impactful in hybrid methods. SAXS provides solution structural information in the form of the radius of gyration (Rg), the maximum particle dimension, and the electron pair distance distribution function (P(r)). Furthermore, SAXS provides information about the molecular mass in solution, oligomeric state, and quaternary structure. [194] Several groups have integrated SAXS data into their protein structure prediction pipeline. [195, 196, 197, 198] Also, in the recent Critical Assessment of Protein Structure Prediction (CASP) competition, SAXS information was incorporated into the data-assisted category that aimed to assess the potential of integrating SAXS data with protein structure prediction methods for protein folding. [188] Most CASP12 approaches utilized SAXS as additional driving restraints involving (1) the goodness-of-fit between the experimental SAXS curve and those computed from models; (2) comparison of the experimental P(r) to the P(r) histogram calculated from the model; and (3) Rg as a restraint on the size of the structure. Although SAXS-based hybrid modeling holds great promise, more research is needed to determine the best ways to fully leverage the experimental information from SAXS in protein structure modeling.

In this work, we investigated the use of restraints from SAXS multidomain assembly. We developed a novel framework to systematically integrate the probabilistic approach for protein conformational sampling with SAXS-assisted structure folding. Our method applies probabilistic Input-Output Hidden Markov model and Monte Carlo sampling to simulate the domain-domain orientation with SAXS related energies enforced, so that it can generate near-native structures that have low free energy and good agreement with the SAXS curve. In addition, we examined the correlation between the SAXS scoring functions and structural qualities (i.e., RMSD) on the CASP proteins, which shows the effectiveness of SAXS data in the structural analysis. Our method shows a significant improvement in domain assembly and structure folding after incorporating SAXS information as additional energies to the physics-based force field, which demonstrates the promise of using SAXS data in computational protein structure modeling.

6.3 Methods

6.3.1 Benchmark sets

To assess how well each SAXS-based pseudo-energy function correlates with structural quality (i.e., RMSD), [132] we collected predicted structural models generated for protein targets that were tested in the 8^{th}, 9^{th}, 10^{th}, and 11^{th} Critical Assessments of Structure Prediction (CASP) experiments. [136] The proteins whose experimental structures were available were selected for preliminary analysis. The dataset contains 112,050 models corresponding to 428 single-domain and multidomain proteins; the detailed statistics are provided in Supporting Information Table S3.

In addition, we evaluated our method on the two types of datasets to validate the effectiveness of SAXS data in protein domain assembly. The first dataset contains multidomain proteins from CASP8-12 whose experimental structures are available. The domain definition (i.e., number of domains and the domain boundaries) of each protein was determined by CASP assessors. [199] Since our method requires continuous domains as input, the domains with chain breaks (defined as distance of adjacent CA-CA atoms larger than 4) were removed from the dataset. Finally, we collected 51 CASP multidomain proteins for the domain assembly analysis. The length of domain linkers among the 51 proteins ranges from 5 to 21. We randomly selected 5 targets to determine the weights for the SAXS terms of the target function. The remaining 46 targets were used to compare the performance of different SAXS scoring functions for domain assembly. The structures of individual domains for all 51 CASP targets were directly derived from their native protein structures and were further used for domain assembly.

The second dataset is a collection of two-domain proteins curated in the Ab initio Domain Assembly (AIDA) server. [186] The number of domains in each protein was determined by DomainParser. [158] Unlike using the native domain structures for assembly in the CASP dataset, we first used our MULTICOM tertiary structure system [21] to predict the structures of individual domains of proteins from their homology templates. The domains whose predicted structures have TM-score > 0.9 against their native structures were selected for domain assembly. Finally, MULTICOM successfully predicted high-quality models for domains of 73 proteins in the AIDA dataset. The length of domain linkers in 73 proteins ranges from 5 to 15. The predicted structures were used for domain assembly analysis.

6.3.2 Domain-Domain orientation driven by united-residue model and probabilistic sampling

Given individual domain structures for a protein sequence, our method first converts the polypeptide chains of domains into united-residue representation as described in the UNRES model. [18, 200] In the UNRES model, the backbone of the polypeptide chain is approximated by a sequence of α-carbon atoms linked by virtual bonds, and the conformation of the protein chain is determined by virtual bond lengths ($b_{C\alpha_i}$), virtual bond angles (θ_i), virtual bond dihedral angles (τ_i) among adjacent α-carbon atoms (**Figure 6.1**). In addition, the united side chains are attached to the α-carbon atoms where two side-chain angles (δ_i and γ_i) and a virtual-bond length (b_{SC_i}) determine the location of side chain. The six variables parameterize the geometry of α-carbon ($C\alpha_i$) and side-chain (SC_i) at the i^{th} residue of a polypeptide chain in conformation space. We used Input-Output Hidden Markov Model (IOHMM) that was trained in our previous work [18] to sample the virtual-bond lengths and virtual-bond torsion angles given the predicted secondary structure in the linker regions. Each cycle of Monte Carlo sampling generates one acceptance move for domain-domain orientation using simulated annealing. The structures of the individual domains are unchanged during sampling (i.e., treated as rigid bodies). Thus, the conformation of the linker regions can be conditionally resampled given the known prior structural information of the domains based on the probabilistic model, which can predict more accurate local structural preferences of linkers than random sampling and potentially reduce the number of local movements in conformational space to achieve convergence.

Our method implements the domain assembly based on the following steps. Given the full-length sequence of a protein, we first predict the sequence's 8-class secondary structure using SSpro. [16] Then we sample the united-residue conformation for the entire polypeptide chain using IOHMM model for structure initialization. After the conformation is initialized, the torsion angles and virtual-bond lengths of α-carbon and its side chain atoms at each position of residues in the full-length polypeptide chain are updated according to their geometry in the pre-determined domain structures. The regions whose structure information are not provided in the provided domain structures are considered as linkers that anchors between domains together. The conformation of the linker regions is are then sampled using the IOHMM model and orients the domain structures using simulated annealing algorithm to generated structure models with lowest structural energy, as depicted in the **Figure 6.1**. Therefore, our method can be applied to assemble any number of domains for multidomain proteins.

6.3.3 Integrating physics-based force field with SAXS restraints for domain-domain assembly

Our method adopts the united-residue physics-based force field that was defined in our previous work to represent the energy of a united-residue peptide chain. [18] The physics energy includes the mean free energy of hydrophobic (hydrophilic) interactions between side chains ($E_{sc_i sc_j}$), excluded-volume potential of side-chain and peptide group interaction ($E_{sc_i p_j}$), and the backbone peptide group interaction to represent the average electrostatic interaction ($E_{p_i p_j}$) for any pair of residues in the i^{th} and j^{th} positions in the polypeptide chain, as represented in Equation 6.1:

$$E_{physics} = w_{sc} * \Sigma_j \Sigma_{i<j} E_{sc_i sc_j} + w_{sc \cdot p} * \Sigma_j \Sigma_{i \neq j} E_{sc_i p_j} + w_{el} * \Sigma_j \Sigma_{i<j} E_{p_i p_j}. \qquad (6.1)$$

Unlike our earlier approach that generated chain conformation based on stepwise sampling of foldon units, our current method only samples the conformation of the linker regions and keeps the structures of the domains fixed. Therefore, the physics-based force field of intra-domain interactions is stable during conformation sampling, and the energy of chain conformation is only affected by the interactions of all inter-domain residues (i.e., interaction interface) and all linker residues, where the physics energy can be further represented as in Equation 6.1:

$$E_{physics} = E_{physics}^{(intra-domain)} + E_{physics}^{(inter-domain)} + E_{physics}^{(linker)} \qquad (6.2)$$

It is worth noting that the energy of hydrophobic (hydrophilic) interactions between side chains of linker residues plays an important role in the protein folding and domain-domain movement. [201] Studies showed that the average residue hydrophobicity (hydrophilicity) is largely influenced by the size of linkers, where longer linkers are more hydrophilic and exposed so that they induced larger domain motions in the conformation space. Inversely, smaller linkers showed more hydrophobic character, which may significantly restrain the domain-domain movement. [202]

We introduced additional energy terms corresponding to the SAXS restraints for the total energy calculation, defined as:

$$E_{saxs} = E_{saxs \cdot IntFit} + E_{saxs \cdot \chi} + E_{saxs \cdot Pr} + E_{saxs \cdot R_g} \tag{6.3}$$

The first term in the SAXS energy, $E_{saxs \cdot IntFit}$, represents the normalized fitness between the experimental SAXS intensity and computed intensity from the models, which is defined as:

$$E_{saxs \cdot IntFit} = w_{saxs \cdot IntFit} * \frac{\Sigma_{i=1}^{N} |I_{exp}(q_i) - I_{model}(q_i)|}{\Sigma_{i=1}^{N} |I_{exp}(q_i)|} \tag{6.4}$$

In Equation 6.4, $I_{exp}(q)$ is the experimental SAXS intensity and $I_{model}(q)$ is the theoretical SAXS intensity calculated from decoys . We employ the same strategy as FoXS [203, 204] to calculate $I_{model}(q)$ and to determine the best fit between $I_{model}(q)$ and $I_{model}(q)$ by minimizing the χ function:

$$\chi = \sqrt{\frac{1}{N} \Sigma_{i=1}^{N} (\frac{I_{exp}(q_i) - cI_{model}(q_i)}{\sigma(q_i)})^2} \tag{6.5}$$

In Equation 6.5, $\sigma(q)$ is the experimental error of the measured SAXS profile, N is the number of points in the profile, and c is the scale factor determined from linear least-squares analysis to derive the minimum value of χ. The second term in the SAXS energy function, includes χ as an additional score term to account for the degree of SAXS profile matching and is defined as follows:

$$E_{saxs \cdot \chi} = w_{saxs \cdot \chi} * \chi \tag{6.6}$$

The third term in the SAXS energy function, $E_{saxs \cdot Pr}$, represents the Kullback-Leibler divergence between the pairwise atom-atom distance distribution function P(r) derived from the experimental SAXS profile and the pair distance distribution computed from the model, which is defined as:

$$E_{saxs \cdot Pr} = w_{saxs \cdot Pr} * \Sigma_{i=1}^{N} Pr_{model}(r_i) * log \frac{Pr_{model}(r_i)}{Pr_{exp}(r_i)} \tag{6.7}$$

The experimental P(r) is calculated from the experimental SAXS intensity curve using an indirect Fourier transform along with an assumption of the maximum particle size (d_{max}). [205, 206] The pair distance distribution of the protein structure is directly calculated from its atomic coordinates.

The last term in the SAXS energy function, $E_{saxs \cdot Rg}$, is a penalty function based the agreement between experimental Rg and the Rg calculated from the protein model:

$$E_{saxs \cdot Rg} = w_{saxs \cdot Rg} * \frac{|RG_{exp} - RG_{model}|}{RG_{exp}} \tag{6.8}$$

The SAXS-related quantities (i.e., SAXS intensity, P(r) and Rg) described above were calculated using algorithms implemented in the Integrated Modeling Platform (IMP) package. [207]

We adopted the same weight configuration for the physics-based force field energy terms listed in Equation 6.1 as our previous method, [18] where w_{sc}=1.00000, $w_{sc \cdot p}$=2.73684, and w_{el}=0.06833. For the SAXS energy terms described in the Equation 6.3, we set w_{χ}=10, $w_{saxs \cdot fit}$=700, $w_{saxs \cdot Pr}$=700, and $w_{saxs \cdot R_g}$=700 after experimenting with several weights on the small training proteins.

In summary, the energy for a multidomain polypeptide chain in our method is:

$$E_{total} = E_{physics}^{intra-domain} + E_{physics}^{inter-domain} + E_{physics}^{linker} + E_{saxs} \tag{6.9}$$

In addition to the four SAXS-related scoring functions as defined in Equation

(6.4-6.8), we also experimented with ten other SAXS-based scoring functions based on the agreement between the experimental SAXS profiles and those computed from models (functions 5-14 of Table S1).

Since the physics-based energies are calculated from united-residue models, but the SAXS energy calculations require the full-atom representation with at least a Cα-trace, we reconstruct the Cα-trace and side chains from the united- residue protein representation using PULCHRA [208] to generate full-atom protein models for SAXS energy calculation. In order to speed up SAXS fitting and computation, the functions of FoXS, [203] PULCHRA [208] and IMP [207] have been incorporated into our system instead of calling them as external programs during sampling.

We used simulated annealing Monte Carlo to search for the lowest-energy assembled multidomain conformation. Since only the linker regions are resampled during domain-domain orientation, the sampling space is significantly reduced. The number of Monte Carlo cycles for each linker is set to the number of residues in linker times 100. Given an assembled protein decoy in each cycle, the total energy, including the physics- and SAXS-based energies, is calculated and compared to the energy of previous conformation. The domain movement is accepted or rejected according to the probability proportional to $\alpha = min(1, e^{(-\Delta E)/t})$, where the ΔE represents the energy change for each domain movement, and t is the temperature of simulated annealing.

6.4 Results Discussion

6.4.1 Evaluation of different SAXS profile matching score functions

We first tested several SAXS scoring functions to identify those that correlate best with the structural quality of a predicted model. Fourteen functions were considered,

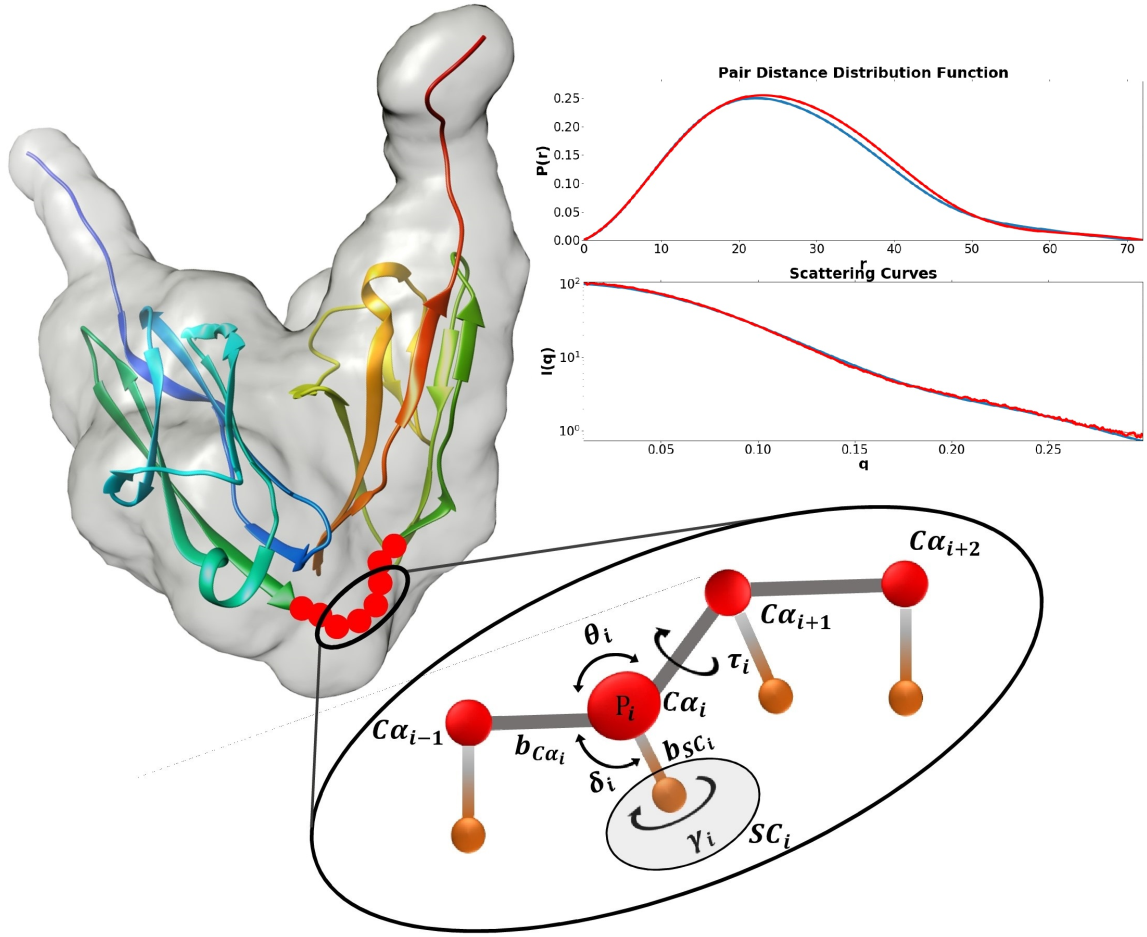


Figure 6.1: Parameterization of conformation in linker regions and overall shape match with SAXS data.

including the four described in detail above (Equations 6.4, 6.6, 6.7, 6.8) and ten more shown in Table S1. The test set consisted of the predicted server models of 428 targets from CASP8 to CASP11 (Table S3). Theoretical SAXS curves (I(q)) were calculated from both the experimental structures and the predicted models using FoXS, [203] and the resulting SAXS curves were used to calculate distance distribution functions (P(r)) using GNOM. [209] For each predicted model, we generated SAXS data from both the full-atom and $C\alpha$-atom structure. Model quality was expressed as the $C\alpha$ Root Mean Square Deviation (RMSD) between the model and its experimental structure.

The Pearson correlation coefficient (PCC) between the RMSD and each of the 14 SAXS scores of all the predicted models for each protein was calculated, and the averaged correlations over the 428 targets are listed in Table S1 (full-atom model) and Table S2 ($C\alpha$-atom model). Three SAXS scores stood out from the others. The P(r)-based function (score 2), Rg agreement function (score 3), and the normalized I(q) fitness function (score 5) showed the highest correlation with RMSD, with averaged PCCs of 0.6, 0.7, and 0.59, respectively when using the full-atom treatment (Table S1). The use of $C\alpha$-atom models led to a similar result, with scores 2, 3, and 5 outperforming the others (Table S2). This result is potentially useful, since $C\alpha$-trace modeling is typically faster than all-atom modeling. The averaged PCCs for the three best functions are shown in **Figure 6.2**. Since the χ function is a common metric for comparison of scattering curves for SAXS, we include it for comparison in **Figure 6.2**. Note that the χ score (score 1 in Table S1) achieved relatively low correlations of 0.47 and 0.38 for full-atom and $C\alpha$-atom models, respectively. Based on these results, we included the three top performing score functions (Equations 6.4, 6.7, 6.8) as SAXS energies in the SAXSdom domain assembly calculations described below.

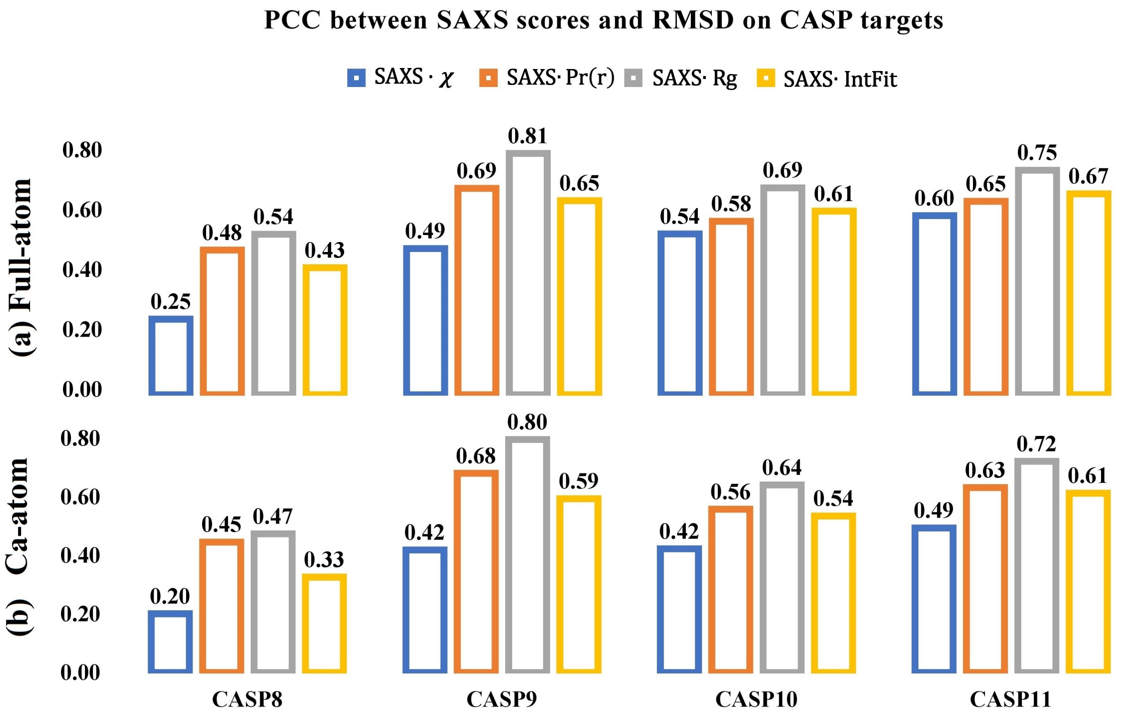


Figure 6.2: Average Pearson correlation coefficient between the structural quality (RMSD) and the SAXS score functions derived from (a) full-atom and (b) $C\alpha$ atom models of protein structure. Analysis was done based on the predicted models from CASP8-11.

6.4.2 Performance of SAXSDom in assembling 46 CASP multidomain proteins

In order to validate the improvement of domain assembly obtained by incorporating SAXS information, we first developed a baseline approach, SAXSDom-abinitio, which used only the united-residue physics based force field (Equation 1) and did not incorporate any SAXS information. We then tested five SAXS-based approaches that adopted four different SAXS energy terms either alone or in combination. The results using the SAXS functions individually are labeled as SAXSDom($E_{saxs \cdot IntFit}$), SAXSDom($E_{saxs \cdot Pr}$), and SAXSDom($E_{saxs \cdot Rg}$), and SAXSDom($E_{saxs \cdot \chi}$). Note these metrics correspond to the top performing functions identified in the previous section, plus the historical SAXS χ statistic. Results obtained when using all four SAXS functions in combination are denoted SAXSDom(E_{saxs}). All SAXSDom methods were employed to assemble domains for 46 CASP multidomain proteins, and each method generated 50 full-length decoys for each protein. For each protein, the initial coordinates of each domain were directly derived from the experimental structure, and the secondary structure of the full-length protein sequence was predicted by SCRATCH. [210] The ”experimental” SAXS intensity profile was calculated by FoXS from the experimental structure. After 50 decoys were generated, we assessed model quality with Qprob [113] to rank the assembled models. (Qprob estimates the prediction error using several physicochemical, structural and energy feature scores, and then uses the combination of probability density distribution of the errors for the global quality assessment.) Each domain assembly method was evaluated based on the averaged TM-score and RMSD of the Qprob-ranked best model, best in top five models, and best in all 50 models for the 46 proteins. The results for the six methods are reported in the **Table 6.1** and **Figure 6.3**.

Incorporation of SAXS information clearly improved the accuracy of domain assembly. For example, whether one considers either the best model, best in top five models, or the best in 50 models, the averaged TM-score and RMSD of the assembled models are consistently better when SAXS information is included compared to using only the physics-based force field (**Table 6.1**). The P-value for the difference between the SAXS-based method and ab initio modeling according to TM-score and RMSD are reported in Table S4. For instance, the method SAXSDom(E_{saxs}), which combines all four SAXS energy terms during conformation sampling, outperforms the method SAXSDom-abinitio by 9.59% (ie., $\frac{0.80-0.73}{0.73}$), 11.84%, 11.25% of TM-score and 38.52%, 46.21%, 46.73% of RMSD for top one, best of five, and best of 50 models respectively, as shown in **Table 6.1**. **Figure 6.3** shows the performance of five SAXSDom methods with different SAXS energies and SAXSDom-abinitio method evaluated on the best of 50 assembled models based on the RMSD, TM-score, and SAXS χ score. According to the evaluation, as shown in **Figure 6.3(A)**, the method SAXSDom(E_{saxs}) outperforms the SAXSDom-abinitio in 40 out of 46 proteins in terms of RMSD and TM-score. We also evaluated the distribution of SAXS χ scores for all generated models. As expected, the SAXS χ scores of assembled models using SAXS information were lower than that of models built by ab initio sampling. As shown in the plot, the distribution of SAXSDom(E_{saxs}) consistently shifted to lower SAXS χ score compared with SAXSDom-abinitio. **Figure 6.3 (B), (C), (D) and (E)** show the performance of domain assembly using four individual SAXS energy terms and their comparison with performance of ab initio sampling.

Altogether, these results show that incorporating SAXS information as additional energies for conformational sampling can improve the accuracy of the domain assembly.

6.4.3 Performance of SAXSDom in AIDA multidomain proteins using predicted domain structures

We also assessed the performance of SAXSDom using 73 multidomain proteins which were originally curated for evaluating the ab initio domain assembly approach AIDA.

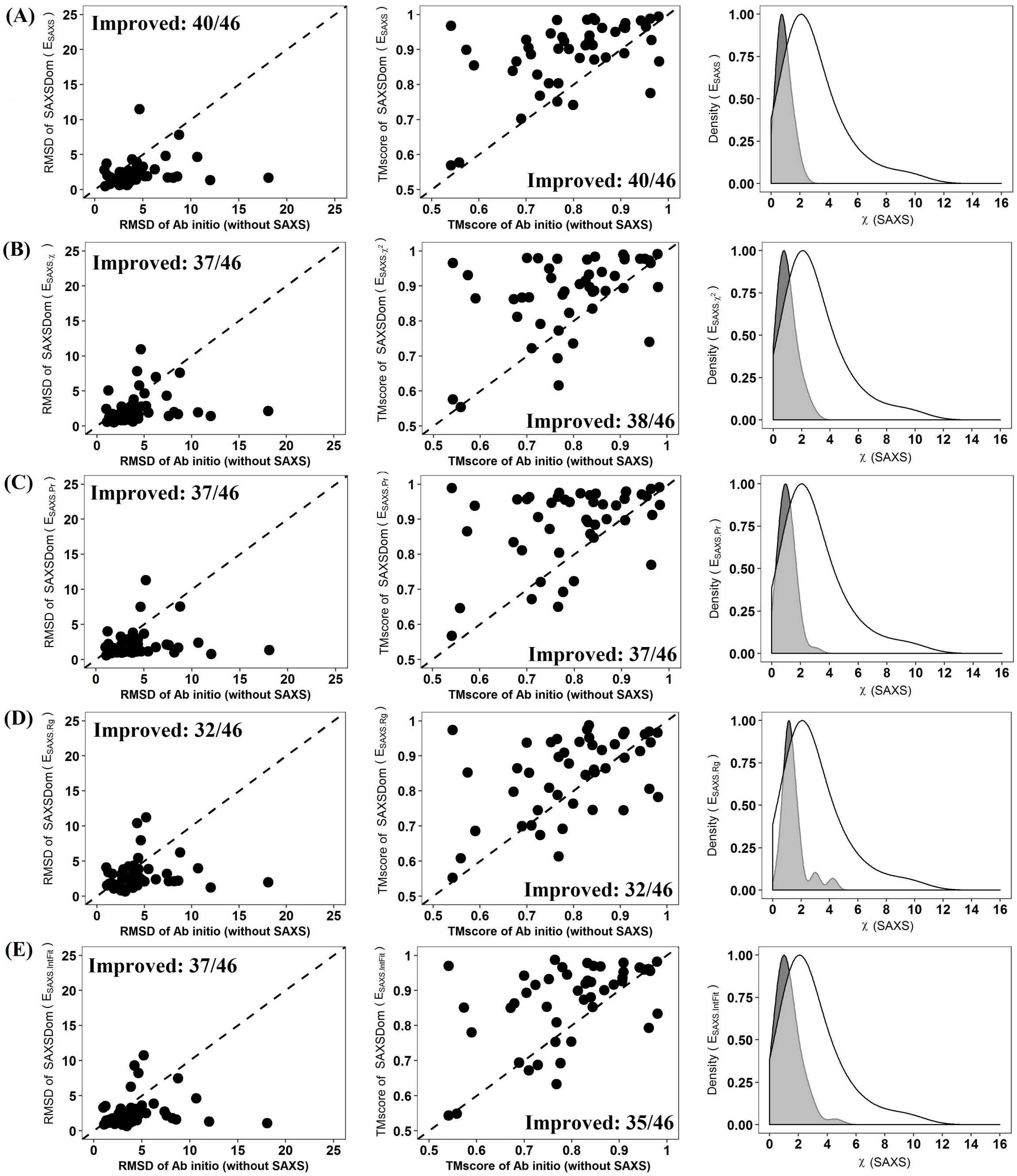


Figure 6.3: Comparison of five SAXSDom approaches with the SAXSDom-abinitio method (does not use SAXS) on the best 50 assembled models. (A) SAXSDom (E_{saxs}) versus SAXSDom-abinitio (Left plot: TM_scores of SAXSDom (E_{saxs}), models versus TM_scores of SAXSDom-abinitio models; Middle plot: RMSD of the models of the two methods; Right plot: Distribution of χ score of all assembled models for 46 proteins by two methods (mark the 2 curves in the plot). (B) SAXSDom ($E_{saxs\cdot\chi}$) versus SAXSDom-abinitio. (C) SAXSDom ($E_{saxs\cdot Pr}$) versus SAXSDom-abinitio. (D) SAXSDom ($E_{saxs\cdot Rg}$) versus SAXSDom-abinitio. (E) SAXSDom ($E_{saxs\cdot IntFit}$) versus SAXSDom-abinitio.

ScoreFunction	Best Model*		Best-of-Five*		Best-of-50*	
	TM-score	RMSD	TM-score	RMSD	TM-score	RMSD
SAXSDom-abinitio	0.73	8.41	0.76	6.47	0.80	4.43
SAXSDom ($E_{saxs \cdot \chi}$)	0.81	5.09	0.85	3.49	0.88	2.60
SAXSDom ($E_{saxs \cdot IntFit}$)	0.76	6.77	0.82	3.96	0.87	2.74
SAXSDom ($E_{saxs \cdot Pr}$)	0.80	5.27	0.85	3.46	0.89	2.29
SAXSDom ($E_{saxs \cdot Rg}$)	0.77	6.20	0.81	4.20	0.85	3.03
SAXSDom ($E_{saxs \cdot}$)	0.80	5.17	0.85	3.48	0.89	2.36

Table 6.1: Summary of the domain assembly performance using ab initio modeling (without SAXS) and ab initio modeling plus different SAXS-related score functions on the 46 multidomain proteins in CASP dataset.

Method	Best Model		Best of 5 models		Best of 50 models		P-value	
	TM-score	RMSD	TM-score	RMSD	TM-score	RMSD	TM-score	RMSD
AIDA	0.716	9.135	0.767	6.444	0.81	4.438	1.00E+00	0.9999
Modeller	0.62	16.207	0.622	15.349	0.621	14.953	2.20E-16	2.20E-16
SAXSDom-abinitio	0.705	9.005	0.724	6.917	0.742	5.811	5.60E-08	1.98E-08
SAXSDom	0.722	7.658	0.75	5.987	0.767	5.012		

Table 6.2: Summary of the domain assembly performance using four domain assembly methods on the 73 proteins in AIDA dataset.

[186] In our work, the domain structures for these 73 proteins were predicted by the MULTICOM tertiary structure prediction method and then further assembled using our protocol. SAXSDom then generated 50 assembled decoys using the reference SAXS intensities derived from the native structures of full-length proteins. Qprob was then used to re-rank the 50 models. The same protocol was applied to SAXSDom-abinitio to generate 50 decoys for the 73 proteins. The accuracy of top Qprob-ranked models (i.e., best model, best of five, best of 50 models) were subsequently evaluated according to TM-score and RMSD. We also compared our methods with another two state-of-art structure modeling approaches, Modeller [5] and AIDA. [186]. For each protein, Modeller and AIDA also generated 50 decoys which were ranked according to their default energies. The qualities of top ranked models generated by Modeller and AIDA were also evaluated and compared to our methods.

Table 6.2 reports the averaged TM-score and RMSD of top ranked models generated by the four methods tested. AIDA achieved relatively better performance in domain assembly compared to the other methods. The main difference between AIDA and our approach is that AIDA uses an all-atom representation of the protein structure, whereas SAXSDom uses a united-residue representation. The results also show that SAXSDom outperforms both SAXSDom-abinitio and Modeller in terms of all metrics with statistical significance shown by the one-sample paired t-test. **Figure 6.4** shows the performance of SAXSDom with SAXSDom-abinitio, AIDA and Modeller evaluated on the best of 50 assembled models based on the RMSD, TM-score, and SAXS χ scores. According to the evaluation, as shown in **Figure 6.4(A)**, the method SAXSDom outperforms the SAXSDom-abinitio in 50 out of 73 proteins in terms of RMSD and 45 out of 73 proteins in terms of TM-score. **Figure 6.4(B)** compares the performance of SAXSDom and AIDA. AIDA was able to assemble domains with slightly better qualities according to RMSD, while SAXSDom can generate assembled decoys that were better matched to the SAXS profile. **Figure 6.4(C)** shows that SAXSDom can generate significantly better models with lower SAXS χ scores compared to that of Modeller. The results of the method comparison evaluated on the top one and best five assembled models are also shown in Figure S3 and S4.

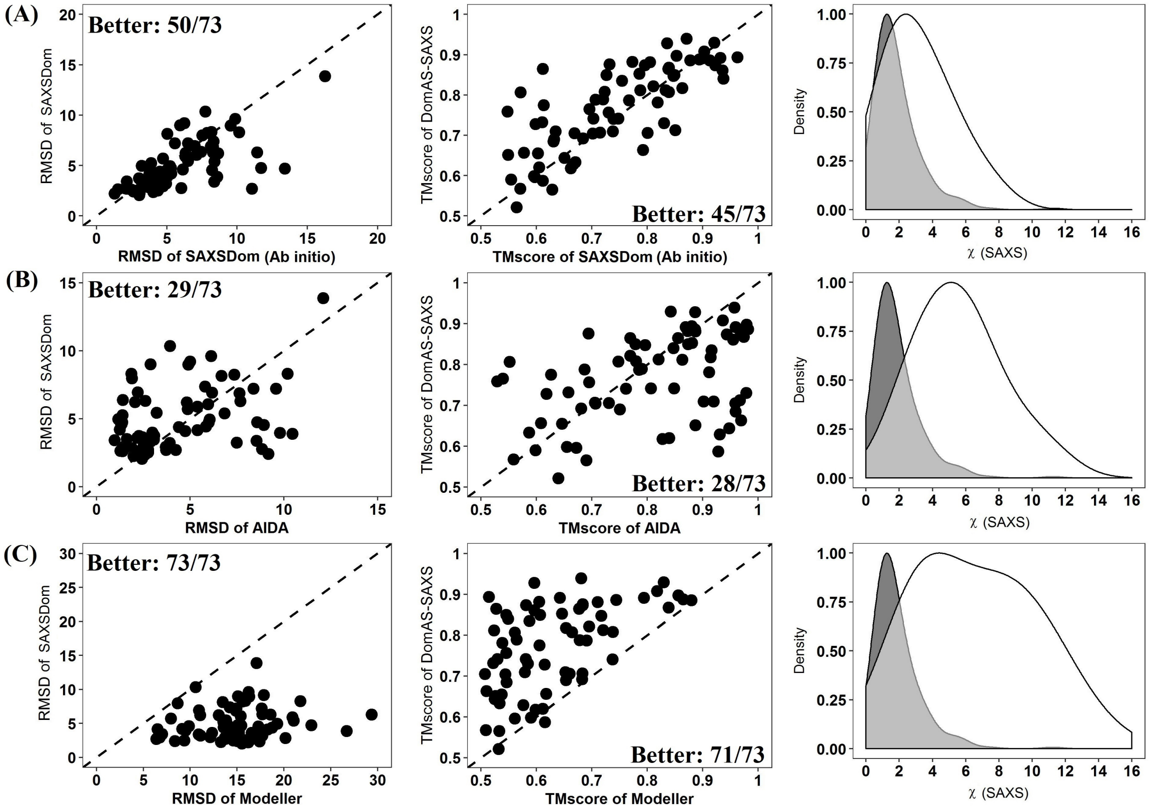


Figure 6.4: Comparison of SAXSDom with SAXSDom-abinitio, AIDA and Modeller on the best of 50 assembled models. (A) SAXSDom versus SAXSDom-abinitio (Left plot: TM_scores of SAXSDom models versus TM_scores of SAXSDom-abinitio models; Middle plot: RMSD of the models of the two methods; Right plot: Distribution of χ scores of all assembled models for 46 proteins by two methods). (B) SAXSDom versus AIDA. (C) SAXSDom versus Modeller.

In addition to the global statistical performance analysis provided so far, we present the results for four representative targets as three-dimensional structures (**Figure 6.5**). The crystal structure of signal recognition particle receptor from E .coli (PDB code 1FTS) consists of an α-helical domain (residues 1-82) connected to an $\alpha\beta\alpha$ domain (residues 92-295) by a of 9-residue linker (**Figure 6.5(A)**). SAXSDom successfully placed the domains into the correct orientation using SAXS information, although the linker conformation is not correct. The assembled structure agrees well with the envelope of the protein structure even though the variation of linker region is relatively large. The agreement of the SAXSDom model with the SAXS data is characterized by χ =2.8 (**Figure 6.6(A)**). **Figure 6.6(A)** shows that the SAXSDom model has better agreement with the SAXS data than the models from the other methods, both for P(r) and the scattering curve. The residue-by-residue distance errors between the experimental structure and the models shows that the accuracy of domain assembly was improved by incorporating SAXS energies in the SAXSDom compared to ab initio method SAXSDom-abinitio (**Figure 6.6(A)**). **Figure 6.6(B)** shows the predicted domain assembly for the ErmC' rRNA methyltransferase (PDB entry 1QAM). The structure consists of two domains, an N-terminal $\alpha\beta\alpha$ domain (residues 1-171) and a C-terminal α domain (residues 176-235). The predicted assembly model has RMSD= 3.0, TMscore=0.81 to the experimental structure, and χ score of 1.6 to the SAXS profile. The domain linker contains 4 residues and is folded into similar shape as that in the native structure. Domain assembly for a protein of unknown function (PDB code 3P02) also achieved good performance, with two β-domains combined into a native-like orientation (RMSD=3.4, TMscore=0.81 and χ score=1.7, **Figure 6.5(C)**). In this case, the structure has a rather short linker of only four residues, which restricts the conformational space needed to be sampled.

Finally, **Figure 6.5(D)** presents the predicted assembly for a myo-inositol monophosphatase (2BJI). The fold consists of a penta-layered $\alpha\beta\alpha\beta\alpha$ sandwich, and the linker connects the last strand of the first β-sheet to the first strand of the second β-sheet. SAXSDom successfully generated a native-like model with RMSD=2.7, TMscore=0.86 and χ score=0.70.

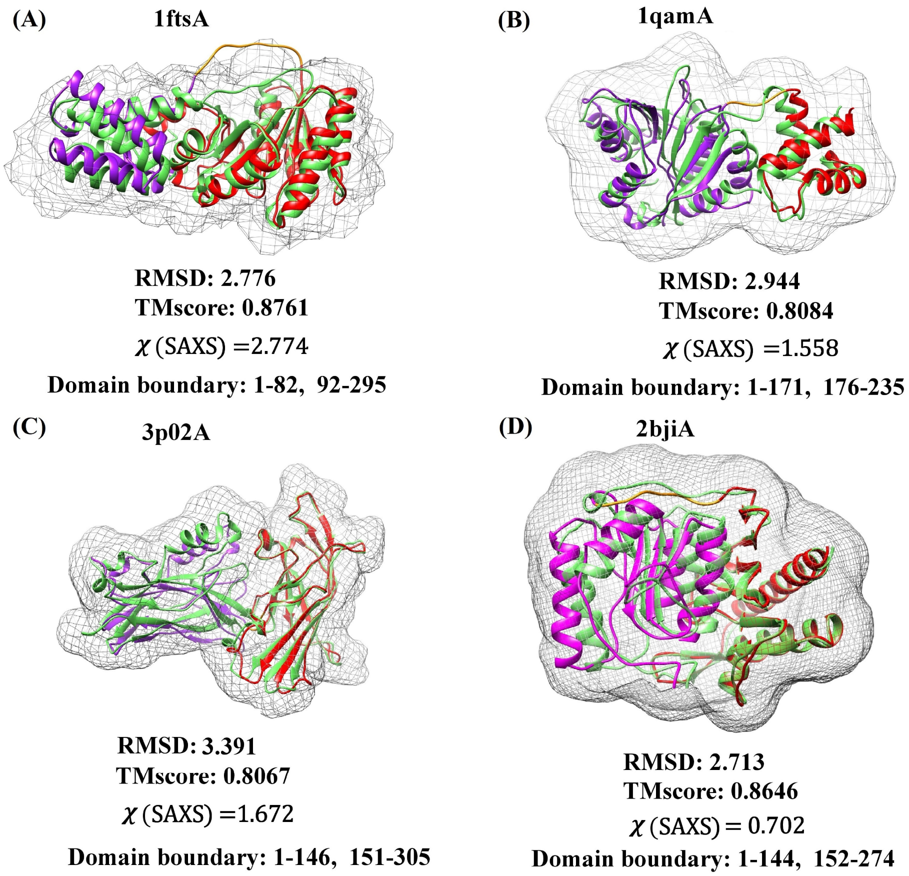


Figure 6.5: The predicted assembly models and shape envelops of five two-domain proteins. The predicted model (colored) and the native structure (green) is superimposed. The domain linker (yellow) and domains (purple, red) are highlighted in the predicted model. (A) The signal recognition particle receptor from E. coli (chain A of 1FTS), linker length = 9, RMSD=2.8, TMscore=0.88, χ score=2.8. (B) The rRNA methyltransferase ErmC' (chain A of 1QAM), linker length = 4, RMSD=2.9, TMscore=0.81, χ score=1.6. (C) Protein of unknown function from Bacteroides ovatus (chain A of 3P02), linker length = 4, RMSD=3.4, TMscore=0.81, χ score=1.7. (D) Myo-inositol monophosphatase (chain A of 2BJI), linker length = 7, RMSD=2.7, TMscore=0.86, χ score=0.70.

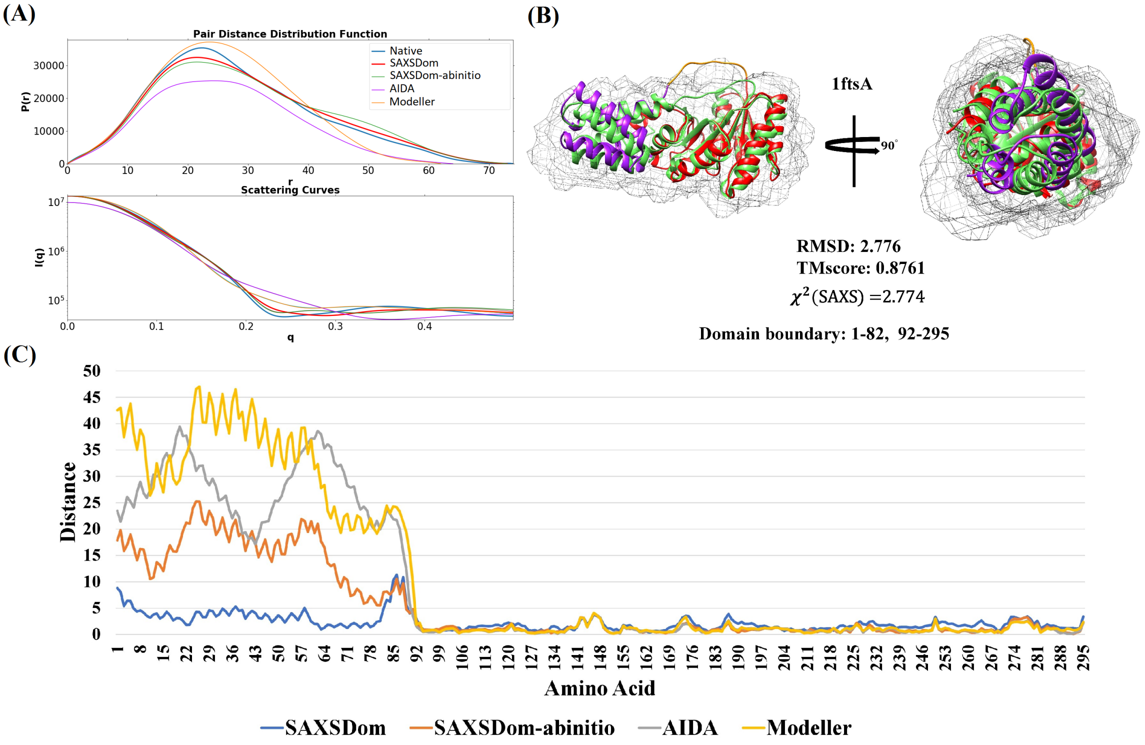


Figure 6.6: Comparison of predicted models for 1FTS by SAXSDom, SAXSDom-abinitio, AIDA and Modeller. (A) The SAXS profiles calculated from the models and the experimental structure. (B) The assembled full-length model with quality measurements. (C) Residue-by-residue distance error between the predicted models and the experimental structure.

6.5 Conclusion and Future work

In this work, we developed a data-assisted domain assembly method, SAXSDom, by integrating the probabilistic approach for backbone conformation sampling with SAXS-assisted restraints in domain assembly. We evaluated several SAXS-related score functions for structure modeling, including fitness of SAXS intensities, the divergence of pair-atom distance distribution, agreement of the radius of gyration, and the traditional chi score. Our results show that incorporating the restraints from SAXS data into de novo conformational sampling method can improve the protein domain assembly. SAXSDom can generate more accurate domain assembly for 40 cases among 46 CASP multidomain proteins in terms of RMSD and TMscore when compared to modeling without using SAXS information. On the AIDA dataset, SAXSDom also achieved better accuracy for 50 out of 73 multidomain proteins according to RMSD metric and 45 out of 73 targets in terms of TMscore. Despite the success of improving protein domain assembly using SAXS data, our method can still be improved in several ways: (1) adopting new physical energies derived from full-atom structures such as van der Waals hard sphere repulsion, residue environment, residue pair, radius of gyration as introduced in Rosetta [6]; (2) extending the continuous domain assembly with discontinuous domain assembly for those proteins with inserted domains; and (3) designing more advanced SAXS scoring functions to guide domain assembly.

Bibliography

[1] UniProt Consortium. Uniprot: the universal protein knowledgebase. *Nucleic acids research*, 46(5):2699, 2018.

[2] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[3] Zheng Wang, Jesse Eickholt, and Jianlin Cheng. Multicom: a multi-level combination approach to protein structure prediction and its assessments in casp8. *Bioinformatics*, 26(7):882–888, 2010.

[4] Jie Hou, Tianqi Wu, Renzhi Cao, and Jianlin Cheng. Protein tertiary structure modeling driven by deep learning and contact distance prediction in casp13. *bioRxiv*, page 552422, 2019.

[5] Narayanan Eswar, Ben Webb, Marc A MartiRenom, MS Madhusudhan, David Eramian, Minyi Shen, Ursula Pieper, and Andrej Sali. Comparative protein structure modeling using modeller. *Current protocols in bioinformatics*, 15(1):5.6. 1–5.6. 30, 2006.

[6] Carol A Rohl, Charlie EM Strauss, Kira MS Misura, and David Baker. *Protein structure prediction using Rosetta*, volume 383, pages 66–93. Elsevier, 2004.

[7] Faruck Morcos, Andrea Pagnani, Bryan Lunt, Arianna Bertolino, Debora S Marks, Chris Sander, Riccardo Zecchina, Jos N Onuchic, Terence Hwa, and Martin Weigt. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, 108(49):E1293–E1301, 2011.

[8] Badri Adhikari, Jie Hou, and Jianlin Cheng. Dncon2: Improved protein contact prediction using two-level deep convolutional neural networks. *Bioinformatics*, 34(9):1466–1472, 2017.

[9] Sheng Wang, Siqi Sun, Zhen Li, Renyu Zhang, and Jinbo Xu. Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLOS Computational Biology*, 13(1):e1005324, 2017.

[10] Badri Adhikari and Jianlin Cheng. Confold2: improved contact-driven ab initio protein structure modeling. *BMC bioinformatics*, 19(1):22, 2018.

[11] Luciano A Abriata, Giorgio E Tam, Bohdan Monastyrskyy, Andriy Kryshtafovych, and Matteo Dal Peraro. Assessment of hard target modeling in casp12 reveals an emerging role of alignmentbased contact prediction methods. *Proteins: Structure, Function, and Bioinformatics*, 86:97–112, 2018.

[12] Renzhi Cao, Debswapna Bhattacharya, Badri Adhikari, Jilong Li, and Jianlin Cheng. Massive integration of diverse protein quality assessment methods to improve template based modeling in casp11. *Proteins: Structure, Function, and Bioinformatics*, 84:247–259, 2016.

[13] Renzhi Cao, Debswapna Bhattacharya, Badri Adhikari, Jilong Li, and Jianlin Cheng. Large-scale model quality assessment for improving protein tertiary structure prediction. *Bioinformatics*, 31(12):i116–i123, 2015.

[14] Debswapna Bhattacharya, Jackson Nowotny, Renzhi Cao, and Jianlin Cheng. 3drefine: an interactive web server for efficient protein structure refinement. *Nucleic acids research*, 44(W1):W406–W409, 2016.

[15] Michael Remmert, Andreas Biegert, Andreas Hauser, and Johannes Sding. Hhblits: lightning-fast iterative protein sequence searching by hmm-hmm alignment. *Nature methods*, 9(2):173, 2012.

[16] Christophe N Magnan and Pierre Baldi. Sspro/accpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity. *Bioinformatics*, 30(18):2592–2597, 2014.

[17] Xin Deng, Jesse Eickholt, and Jianlin Cheng. Predisorder: ab initio sequence-based prediction of protein disordered regions. *BMC bioinformatics*, 10(1):436, 2009.

[18] Debswapna Bhattacharya, Renzhi Cao, and Jianlin Cheng. Unicon3d: de novo protein structure prediction using united-residue conformational search via stepwise, probabilistic sampling. *Bioinformatics*, 32(18):2791–2799, 2016.

[19] Debswapna Bhattacharya and Jianlin Cheng. De novo protein conformational sampling using a probabilistic graphical model. *Scientific reports*, 5:16332, 2015.

[20] Jianlin Cheng. A multi-template combination algorithm for protein comparative modeling. *BMC structural biology*, 8(1):18, 2008.

[21] Jilong Li, Xin Deng, Jesse Eickholt, and Jianlin Cheng. Designing and benchmarking the multicom protein structure prediction system. *BMC structural biology*, 13(1):2, 2013.

[22] Jie Hou, Zhiye Guo, and Jianlin Cheng. Dnss2: improved ab initio protein secondary structure prediction using advanced deep learning architectures. *bioRxiv*, page 639021, 2019.

[23] Jie Hou, Badri Adhikari, and Jianlin Cheng. Deepsf: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*, 34(8):1295–1303, 2017.

[24] Jie Hou, Renzhi Cao, and Jianlin Cheng. Deep convolutional neural networks for predicting the quality of single protein structural models. *bioRxiv*, page 590620, 2019.

[25] Jie Hou, Badri Adhikari, John J Tanner, and Jianlin Cheng. Saxsdom: Modeling multi-domain protein structures using small-angle x-ray scattering data. *bioRxiv*, page 559617, 2019.

[26] Linus Pauling, Robert B Corey, and Herman R Branson. The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. *Proceedings of the National Academy of Sciences*, 37(4):205–211, 1951.

[27] Mirco Michel, David Menndez Hurtado, and Arne Elofsson. Pconsc4: fast, accurate, and hassle-free contact predictions. *Bioinformatics*, pages bty1036–bty1036, 2018.

[28] David T Jones, Michael Tress, Kevin Bryson, and Caroline Hadley. Successful recognition of protein folds using threading methods biased by sequence similarity and predicted secondary structure. *Proteins: Structure, Function, and Bioinformatics*, 37(S3):104–111, 1999.

[29] Jeffrey K Myers and Terrence G Oas. Preorganized secondary structure as an important determinant of fast protein folding. *Nature Structural Molecular Biology*, 8(6):552, 2001.

[30] Ambrish Roy, Alper Kucukural, and Yang Zhang. I-tasser: a unified platform for automated protein structure and function prediction. *Nature protocols*, 5(4):725, 2010.

[31] Renzhi Cao, Debswapna Bhattacharya, Jie Hou, and Jianlin Cheng. Deepqa: improving the estimation of single protein model quality with deep belief networks. *BMC bioinformatics*, 17(1):495, 2016.

[32] Karolis Uziela, Nanjiang Shu, Bjrn Wallner, and Arne Elofsson. Proq3: Improved model quality assessments using rosetta energy terms. *Scientific reports*, 6:33509, 2016.

[33] Andriy Kryshtafovych, Bohdan Monastyrskyy, Krzysztof Fidelis, John Moult, Torsten Schwede, and Anna Tramontano. Evaluation of the template-based modeling in casp12. *Proteins: Structure, Function, and Bioinformatics*, 86:321–334, 2018.

[34] Sergey Ovchinnikov, Hahnbeom Park, David E Kim, Frank DiMaio, and David Baker. Protein structure prediction using rosetta in casp12. *Proteins: Structure, Function, and Bioinformatics*, 86:113–121, 2018.

[35] Burkhard Rost. Protein secondary structure prediction continues to rise. *Journal of structural biology*, 134(2-3):204–218, 2001.

[36] Yuedong Yang, Jianzhao Gao, Jihua Wang, Rhys Heffernan, Jack Hanson, Kuldip Paliwal, and Yaoqi Zhou. Sixty-five years of the long march in protein secondary structure prediction: the final stretch? *Briefings in bioinformatics*, 19(3):482–494, 2016.

[37] Peter Y Chou and Gerald D Fasman. Prediction of protein conformation. *Biochemistry*, 13(2):222–245, 1974.

[38] Stephen F Altschul, Thomas L Madden, Alejandro A Schffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–3402, 1997.

[39] Ofer Dor and Yaoqi Zhou. Achieving 80% tenfold crossvalidated accuracy for secondary structure prediction by largescale training. *Proteins: Structure, Function, and Bioinformatics*, 66(4):838–845, 2007.

[40] David T Jones. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of molecular biology*, 292(2):195–202, 1999.

[41] Gianluca Pollastri and Aoife Mclysaght. Porter: a new, accurate server for protein secondary structure prediction. *Bioinformatics*, 21(8):1719–1720, 2004.

[42] Gianluca Pollastri, Darisz Przybylski, Burkhard Rost, and Pierre Baldi. Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins: Structure, Function, and Bioinformatics*, 47(2):228–235, 2002.

[43] Qiaozhen Meng, Zhenling Peng, Jianyi Yang, and Alfonso Valencia. Coabind: a novel algorithm for coenzyme a (coa)-and coa derivatives-binding residues prediction. *Bioinformatics*, 1:7, 2018.

[44] William R Atchley, Jieping Zhao, Andrew D Fernandes, and Tanja Drke. Solving the protein sequence metric problem. *Proceedings of the National Academy of Sciences*, 102(18):6395–6400, 2005.

[45] Matt Spencer, Jesse Eickholt, and Jianlin Cheng. A deep learning network approach to ab initio protein secondary structure prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 12(1):103–112, 2015.

[46] L Howard Holley and Martin Karplus. Protein secondary structure prediction with a neural network. *Proceedings of the National Academy of Sciences*, 86(1):152–156, 1989.

[47] Ning Qian and Terrence J Sejnowski. Predicting the secondary structure of globular proteins using neural network models. *Journal of molecular biology*, 202(4):865–884, 1988.

[48] J-F Gibrat, J Garnier, and Barry Robson. Further developments of protein secondary structure prediction using information theory: new parameters and consideration of residue pairs. *Journal of molecular biology*, 198(3):425–443, 1987.

[49] Scott C Schmidler, Jun S Liu, and Douglas L Brutlag. Bayesian segmentation of protein secondary structure. *Journal of computational biology*, 7(1-2):233–248, 2000.

[50] Paul Stolorz, Alan Lapedes, and Yuan Xia. Predicting protein secondary structure using neural net and statistical methods. *Journal of Molecular Biology*, 225(2):363–377, 1992.

[51] Chao Fang, Yi Shang, and Dong Xu. Mufoldss: New deep inceptioninsideinception networks for protein secondary structure prediction. *Proteins: Structure, Function, and Bioinformatics*, 86(5):592–598, 2018.

[52] Eshel Faraggi, Tuo Zhang, Yuedong Yang, Lukasz Kurgan, and Yaoqi 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*, 33(3):259–267, 2012.

[53] Rhys Heffernan, Yuedong Yang, Kuldip Paliwal, and Yaoqi 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*, page btx218, 2017.

[54] Sheng Wang, Jian Peng, Jianzhu Ma, and Jinbo Xu. Protein secondary structure prediction using deep convolutional neural fields. *Scientific reports*, 6, 2016.

[55] Rhys Heffernan, Kuldip Paliwal, James Lyons, Abdollah Dehzangi, Alok Sharma, Jihua Wang, Abdul Sattar, Yuedong Yang, and Yaoqi Zhou. Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning. *Scientific reports*, 5:11476, 2015.

[56] Mirko Torrisi, Manaz Kaleel, and Gianluca Pollastri. Porter 5: fast, state-of-the-art ab initio prediction of protein secondary structure in 3 and 8 classes. *bioRxiv*, page 289033, 2018.

[57] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[58] Ming Liang and Xiaolin Hu. Recurrent convolutional neural network for object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3367–3375.

[59] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[60] Joel Moniz and Christopher Pal. Convolutional residual memory networks. *arXiv preprint arXiv:1606.05262*, 2016.

[61] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. *arXiv preprint arXiv:1605.07648*, 2016.

[62] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

[63] Guoli Wang and Roland L Dunbrack Jr. Pisces: a protein sequence culling server. *Bioinformatics*, 19(12):1589–1591, 2003.

[64] Weizhong Li and Adam Godzik. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–1659, 2006.

[65] Adam Zemla, eslovas Venclovas, Krzysztof Fidelis, and Burkhard Rost. A modified definition of sov, a segmentbased measure for protein secondary structure prediction assessment. *Proteins: Structure, Function, and Bioinformatics*, 34(2):220–223, 1999.

[66] Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: pattern recognition of hydrogenbonded and geometrical features. *Biopolymers: Original Research on Biomolecules*, 22(12):2577–2637, 1983.

[67] UniProt Consortium. Uniprot: a hub for protein information. *Nucleic acids research*, 43(D1):D204–D212, 2014.

[68] Milot Mirdita, Lars von den Driesch, Clovis Galiez, Maria J Martin, Johannes Sding, and Martin Steinegger. Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic acids research*, 45(D1):D170–D176, 2016.

[69] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[70] Renxiang Yan, Dong Xu, Jianyi Yang, Sara Walker, and Yang Zhang. A comparative assessment and analysis of 20 representative sequence alignment methods for protein structure prediction. *Scientific reports*, 3:2619, 2013.

[71] Liam J McGuffin, Kevin Bryson, and David T Jones. The psipred protein structure prediction server. *Bioinformatics*, 16(4):404–405, 2000.

[72] Ken A Dill, S Banu Ozkan, M Scott Shell, and Thomas R Weikl. The protein folding problem. *Annu. Rev. Biophys.*, 37:289–316, 2008.

[73] Caroline Hadley and David T Jones. A systematic comparison of protein structure classifications: Scop, cath and fssp. *Structure*, 7(9):1099–1112, 1999.

[74] Alexey G Murzin, Steven E Brenner, Tim Hubbard, and Cyrus Chothia. Scop: a structural classification of proteins database for the investigation of sequences and structures. *Journal of molecular biology*, 247(4):536–540, 1995.

[75] Lesley H Greene, Tony E Lewis, Sarah Addou, Alison Cuff, Tim Dallman, Mark Dibley, Oliver Redfern, Frances Pearl, Rekha Nambudiry, and Adam Reid. The cath domain structure database: new protocols and classification levels give a more comprehensive resource for exploring evolution. *Nucleic acids research*, 35(suppl 1):D291–D297, 2007.

[76] Lisa Holm and Chris Sander. The fssp database of structurally aligned protein fold families. *Nucleic acids research*, 22(17):3600, 1994.

[77] Hua Cheng, R Dustin Schaeffer, Yuxing Liao, Lisa N Kinch, Jimin Pei, Shuoyong Shi, Bong-Hyun Kim, and Nick V Grishin. Ecod: an evolutionary classification of protein domains. *PLoS computational biology*, 10(12):e1003926, 2014.

[78] John-Marc Chandonia, Naomi K Fox, and Steven E Brenner. Scope: Manual curation and artifact removal in the structural classification of proteinsextended database. *Journal of Molecular Biology*, 2016.

[79] Taeho Jo, Jie Hou, Jesse Eickholt, and Jianlin Cheng. Improving protein fold recognition by deep learning networks. *Scientific reports*, 5:17573, 2015.

[80] Johannes Sding. Protein homology detection by hmmhmm comparison. *Bioinformatics*, 21(7):951–960, 2005.

[81] Renzhi Cao and Jianlin Cheng. Integrated protein function prediction by mining function associations, sequences, and proteinprotein and genegene interaction networks. *Methods*, 93:84–91, 2016.

[82] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990.

[83] Steven Henikoff and Jorja G Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, 1992.

[84] Jianzhu Ma, Sheng Wang, Zhiyong Wang, and Jinbo Xu. Mrfalign: protein homology detection through alignment of markov random fields. *PLoS Comput Biol*, 10(3):e1003500, 2014.

[85] I-Fang Chung, Chuen-Der Huang, Ya-Hsin Shen, and Chin-Teng Lin. Recognition of structure classification of protein folding by nn and svm hierarchical

learning architecture. *Artificial Neural Networks and Neural Information ProcessingICANN/ICONIP 2003*, pages 179–179, 2003.

[86] Theodoros Damoulas and Mark A Girolami. Probabilistic multi-class multi-kernel learning: on protein fold recognition and remote homology detection. *Bioinformatics*, 24(10):1264–1270, 2008.

[87] Qiwen Dong, Shuigeng Zhou, and Jihong Guan. A new taxonomy-based protein fold recognition approach based on autocross-covariance transformation. *Bioinformatics*, 25(20):2655–2662, 2009.

[88] Leyi Wei, Minghong Liao, Xing Gao, and Quan Zou. Enhanced protein fold prediction method through a novel feature extraction technique. *IEEE transactions on nanobioscience*, 14(6):649–659, 2015.

[89] Hong-Bin Shen and Kuo-Chen Chou. Ensemble classifier for protein fold pattern recognition. *Bioinformatics*, 22(14):1717–1722, 2006.

[90] Jiaqi Xia, Zhenling Peng, Dawei Qi, Hongbo Mu, and Jianyi Yang. An ensemble approach to protein fold classification by integration of template-based assignment and support vector machine classifier. *Bioinformatics*, 33(6):863–870, 2016.

[91] Jianlin Cheng and Pierre Baldi. A machine learning information retrieval approach to protein fold recognition. *Bioinformatics*, 22(12):1456–1463, 2006.

[92] Taeho Jo and Jianlin Cheng. Improving protein fold recognition by random forest. *BMC bioinformatics*, 15(11):S14, 2014.

[93] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[94] Jesse Eickholt and Jianlin Cheng. Predicting protein residueresidue contacts using deep networks and boosting. *Bioinformatics*, 28(23):3066–3072, 2012.

[95] Lisa N Kinch, Wenlin Li, R Dustin Schaeffer, Roland L Dunbrack, Bohdan Monastyrskyy, Andriy Kryshtafovych, and Nick V Grishin. Casp 11 target classification. *Proteins: Structure, Function, and Bioinformatics*, 2016.

[96] Lisa N Kinch, Shuoyong Shi, Hua Cheng, Qian Cong, Jimin Pei, Valerio Mariani, Torsten Schwede, and Nick V Grishin. Casp9 target classification. *Proteins: Structure, Function, and Bioinformatics*, 79(S10):21–36, 2011.

[97] Yang Zhang and Jeffrey Skolnick. Tm-align: a protein structure alignment algorithm based on the tm-score. *Nucleic acids research*, 33(7):2302–2309, 2005.

[98] Jinrui Xu and Yang Zhang. How significant is a protein structure similarity with tm-score= 0.5? *Bioinformatics*, 26(7):889–895, 2010.

[99] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

[100] Xuefeng Cui, Zhiwu Lu, Sheng Wang, Jim Jing-Yan Wang, and Xin Gao. Cmsearch: simultaneous exploration of protein sequence space and structure space improves not only protein homology detection but also protein structure prediction. *Bioinformatics*, 32(12):i332–i340, 2016.

[101] Sophie E Jackson and Alan R Fersht. Folding of chymotrypsin inhibitor 2. 1. evidence for a two-state transition. *Biochemistry*, 30(43):10428–10435, 1991.

[102] Sheng Wang, Shunyan Weng, Jianzhu Ma, and Qingming Tang. Deepcnf-d: predicting protein order/disorder regions by weighted deep convolutional neural fields. *International journal of molecular sciences*, 16(8):17315–17330, 2015.

[103] John Moult, Krzysztof Fidelis, Andriy Kryshtafovych, Torsten Schwede, and Anna Tramontano. Critical assessment of methods of protein structure prediction: Progress and new directions in round xi. *Proteins: Structure, Function, and Bioinformatics*, 84(S1):4–14, 2016.

[104] Badri Adhikari, Debswapna Bhattacharya, Renzhi Cao, and Jianlin Cheng. Confold: residue-residue contact-guided ab initio protein folding. *Proteins: Structure, Function, and Bioinformatics*, 83(8):1436–1449, 2015.

[105] Jianlin Cheng, Jesse Eickholt, Zheng Wang, and Xin Deng. Recursive protein modeling: a divide and conquer strategy for protein structure prediction and its case study in casp9. *Journal of bioinformatics and computational biology*, 10(03):1242003, 2012.

[106] John Moult, Krzysztof Fidelis, Andriy Kryshtafovych, Torsten Schwede, and Anna Tramontano. Critical assessment of methods of protein structure prediction (casp)round x. *Proteins: Structure, Function, and Bioinformatics*, 82(S2):1–6, 2014.

[107] Renzhi Cao, Zheng Wang, and Jianlin Cheng. Designing and evaluating the multicom protein local and global model quality prediction methods in the casp10 experiment. *BMC structural biology*, 14(1):13, 2014.

[108] Renzhi Cao, Zheng Wang, Yiheng Wang, and Jianlin Cheng. Smoq: a tool for predicting the absolute residue-specific quality of a single protein model with support vector machines. *BMC bioinformatics*, 15(1):120, 2014.

[109] Arjun Ray, Erik Lindahl, and Bjrn Wallner. Improved model quality assessment using proq2. *BMC bioinformatics*, 13(1):224, 2012.

[110] Woong-Hee Shin, Xuejiao Kang, Jian Zhang, and Daisuke Kihara. Prediction of local quality of protein structure models considering spatial neighbors in graphical models. *Scientific reports*, 7, 2017.

[111] Gregory E Sims and Sung-Hou Kim. A method for evaluating the structural quality of protein models by using higher-order pairs scoring. *Proceedings of the National Academy of Sciences of the United States of America*, 103(12):4428–4432, 2006.

[112] Renzhi Cao, Badri Adhikari, Debswapna Bhattacharya, Miao Sun, Jie Hou, and Jianlin Cheng. Qacon: single model quality assessment using protein structural and contact information with machine learning techniques. *Bioinformatics*, 33(4):586–588, 2017.

[113] Renzhi Cao and Jianlin Cheng. Protein single-model quality assessment by feature-based probability density functions. *Scientific reports*, 6:23990, 2016.

[114] Xiaoyang Jing, Qiwen Dong, Xuan Liu, and Bin Liu. Protein model quality assessment by learning-to-rank. In *Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on*, pages 91–96. IEEE.

[115] Liam J McGuffin. Prediction of global and local model quality in casp8 using the modfold server. *Proteins: Structure, Function, and Bioinformatics*, 77(S9):185–190, 2009.

[116] Kliment Olechnovi and eslovas Venclovas. Voromqa: Assessment of protein structure quality using interatomic contact areas. *Proteins: Structure, Function, and Bioinformatics*, 85(6):1131–1145, 2017.

[117] Minyi Shen and Andrej Sali. Statistical potential for assessment and prediction of protein structures. *Protein science*, 15(11):2507–2524, 2006.

[118] Zheng Wang, Allison N Tegge, and Jianlin Cheng. Evaluating the absolute quality of a single protein model using structural features and support vector machines. *Proteins: Structure, Function, and Bioinformatics*, 75(3):638–647, 2009.

[119] Adam Zemla, eslovas Venclovas, John Moult, and Krzysztof Fidelis. Processing and analysis of casp3 protein structure predictions. *Proteins: Structure, Function, and Bioinformatics*, 37(S3):22–29, 1999.

[120] Liam J McGuffin and Daniel B Roche. Rapid model quality assessment for protein structure predictions using the comparison of multiple models without structural alignments. *Bioinformatics*, 26(2):182–188, 2009.

[121] Marcin J Skwark and Arne Elofsson. Pconsd: ultra rapid, accurate model quality assessment for protein structure prediction. *Bioinformatics*, 29(14):1817–1818, 2013.

[122] Zheng Wang, Jesse Eickholt, and Jianlin Cheng. Apollo: a quality assessment service for single and multiple protein models. *Bioinformatics*, 27(12):1715–1716, 2011.

[123] Silvio CE Tosatto and Roberto Battistutta. Tap score: torsion angle propensity normalization applied to local protein structure evaluation. *BMC bioinformatics*, 8(1):155, 2007.

[124] Balachandran Manavalan and Jooyoung Lee. Svmqa: supportvector-machine-based protein single-model quality assessment. *Bioinformatics*, 33(16):2496–2503, 2017.

[125] Jian Zhang and Yang Zhang. A novel side-chain orientation dependent potential derived from random-walk reference state for protein fold selection and structure prediction. *PloS one*, 5(10):e15386, 2010.

[126] Yinghao Wu, Mingyang Lu, Mingzhi Chen, Jialin Li, and Jianpeng Ma. Opusca: A knowledgebased potential function requiring only c positions. *Protein science*, 16(7):1449–1463, 2007.

[127] Yuedong Yang and Yaoqi Zhou. Specific interactions for ab initio folding of protein terminal regions with secondary structures. *Proteins: Structure, Function, and Bioinformatics*, 72(2):793–803, 2008.

[128] Jianyi Yang, Yan Wang, and Yang Zhang. Resq: an approach to unified estimation of b-factor and residue-specific error in protein structure prediction. *Journal of molecular biology*, 428(4):693–701, 2016.

[129] Jesse Eickholt and Jianlin Cheng. A study and benchmark of dncon: a method for protein residue-residue contact prediction using deep networks. *BMC bioinformatics*, 14(14):S12, 2013.

[130] Hongyi Zhou and Jeffrey Skolnick. Goap: a generalized orientation-dependent, all-atom statistical potential for protein structure prediction. *Biophysical journal*, 101(8):2043–2052, 2011.

[131] Adam Zemla. Lga: a method for finding 3d similarities in protein structures. *Nucleic acids research*, 31(13):3370–3374, 2003.

[132] Yang Zhang and Jeffrey Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710, 2004.

[133] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

[134] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[135] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147.

[136] Andriy Kryshtafovych, Bohdan Monastyrskyy, and Krzysztof Fidelis. Casp 11 statistics and the prediction center evaluation system. *Proteins: Structure, Function, and Bioinformatics*, 84:15–19, 2016.

[137] Lisa N Kinch, Wenlin Li, Bohdan Monastyrskyy, Andriy Kryshtafovych, and Nick V Grishin. Evaluation of free modeling targets in casp11 and roll. *Proteins: Structure, Function, and Bioinformatics*, 84:51–66, 2016.

[138] Joerg Schaarschmidt, Bohdan Monastyrskyy, Andriy Kryshtafovych, and Alexandre MJJ Bonvin. Assessment of contact predictions in casp12: Co-evolution and deep learning coming of age. *Proteins: Structure, Function, and Bioinformatics*, 86:51–66, 2018.

[139] Debora S Marks, Lucy J Colwell, Robert Sheridan, Thomas A Hopf, Andrea Pagnani, Riccardo Zecchina, and Chris Sander. Protein 3d structure computed from evolutionary sequence variation. *PloS one*, 6(12):e28766, 2011.

[140] Bohdan Monastyrskyy, Daniel D'Andrea, Krzysztof Fidelis, Anna Tramontano, and Andriy Kryshtafovych. New encouraging developments in contact prediction: Assessment of the casp 11 results. *Proteins: Structure, Function, and Bioinformatics*, 84:131–144, 2016.

[141] Jack Hanson, Kuldip Paliwal, Thomas Litfin, Yuedong Yang, Yaoqi Zhou, and Alfonso Valencia. Accurate prediction of protein contact maps by coupling resid-

ual two-dimensional bidirectional long short-term memory with convolutional neural networks. *Bioinformatics*, 2018.

[142] David T Jones and Shaun M Kandathil. High precision in protein contact prediction using fully convolutional neural networks and minimal sequence features. *Bioinformatics*, 1:8, 2018.

[143] Magnus Ekeberg, Tuomo Hartonen, and Erik Aurell. Fast pseudolikelihood maximization for direct-coupling analysis of protein structure from many homologous amino-acid sequences. *Journal of Computational Physics*, 276:341–356, 2014.

[144] David T Jones, Daniel WA Buchan, Domenico Cozzetto, and Massimiliano Pontil. Psicov: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics*, 28(2):184–190, 2011.

[145] Stefan Seemayer, Markus Gruber, and Johannes Sding. Ccmpredfast and precise prediction of protein residueresidue contacts from correlated mutations. *Bioinformatics*, 30(21):3128–3130, 2014.

[146] Andreas Biegert and Johannes Sding. Sequence context-specific profiles for homology searching. *Proceedings of the National Academy of Sciences*, 106(10):3770–3775, 2009.

[147] Ruslan Sadreyev and Nick Grishin. Compass: a tool for comparison of multiple protein alignments with assessment of statistical significance. *Journal of molecular biology*, 326(1):317–336, 2003.

[148] Dong Xu, Lukasz Jaroszewski, Zhanwen Li, and Adam Godzik. Ffas-3d: improving fold recognition by including optimized structural features and template re-ranking. *Bioinformatics*, 30(5):660–667, 2013.

[149] Robert D Finn, Jody Clements, and Sean R Eddy. Hmmer web server: interactive sequence similarity searching. *Nucleic acids research*, 39(suppl˙2):W29–W37, 2011.

[150] L Steven Johnson, Sean R Eddy, and Elon Portugaly. Hidden markov model speed heuristic and iterative hmm search procedure. *BMC bioinformatics*, 11(1):431, 2010.

[151] Richard Hughey and Anders Krogh. Sam: Sequence alignment and modeling software system. 1995.

[152] Martin Madera. Profile comparer: a program for scoring and aligning profile hidden markov models. *Bioinformatics*, 24(22):2630–2631, 2008.

[153] Morten Kllberg, Gohar Margaryan, Sheng Wang, Jianzhu Ma, and Jinbo Xu. *RaptorX server: a resource for template-based protein structure modeling*, pages 17–27. Springer, 2014.

[154] Sitao Wu and Yang Zhang. Muster: improving protein sequence profileprofile alignments by using multiple sources of structure information. *Proteins: Structure, Function, and Bioinformatics*, 72(2):547–556, 2008.

[155] Lszl Kajn, Thomas A Hopf, Mat Kala, Debora S Marks, and Burkhard Rost. Freecontact: fast and free software for protein contact prediction from residue co-evolution. *BMC bioinformatics*, 15(1):85, 2014.

[156] Andrew Leaver-Fay, Michael Tyka, Steven M Lewis, Oliver F Lange, James Thompson, Ron Jacak, Kristian W Kaufman, P Douglas Renfrew, Colin A Smith, and Will Sheffler. *ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules*, volume 487, pages 545–574. Elsevier, 2011.

[157] Michael J Bower, Fred E Cohen, and Roland L Dunbrack Jr. Prediction of protein side-chain rotamers from a backbone-dependent rotamer library: a new homology modeling tool1. *Journal of molecular biology*, 267(5):1268–1282, 1997.

[158] Ying Xu, Dong Xu, and Harold N Gabow. Protein domain decomposition using a graph-theoretic approach. *Bioinformatics*, 16(12):1091–1104, 2000.

[159] Nickolai Alexandrov and Ilya Shindyalov. Pdp: protein domain parser. *Bioinformatics*, 19(3):429–430, 2003.

[160] Axel T Brunger. Version 1.2 of the crystallography and nmr system. *Nature protocols*, 2(11):2728, 2007.

[161] Mikhail Karasikov, Guillaume Pags, and Sergei Grudinin. Smooth orientation-dependent scoring function for coarse-grained protein quality assessment. *Bioinformatics*, 2018.

[162] Mingyang Lu, Athanasios D Dousis, and Jianpeng Ma. Opus-psp: an orientation-dependent statistical all-atom potential derived from side-chain packing. *Journal of molecular biology*, 376(1):288–301, 2008.

[163] Dmitry Rykunov and Andrs Fiser. Effects of amino acid composition, finite size of proteins, and sparse statistics on distance-dependent statistical pair potentials. *Proteins: Structure, Function, and Bioinformatics*, 67(3):559–568, 2007.

[164] Kliment Olechnovic and Ceslovas Venclovas. Voronota: A fast and reliable tool for computing the vertices of the voronoi diagram of atomic balls. *Journal of computational chemistry*, 35(8):672–681, 2014.

[165] Jesper Lundstrm, Leszek Rychlewski, Janusz Bujnicki, and Arne Elofsson. Pcons: A neural-networkbased consensus predictor that improves fold recognition. *Protein Science*, 10(11):2354–2362, 2001.

[166] Valerio Mariani, Marco Biasini, Alessandro Barbato, and Torsten Schwede. lddt: a local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics*, 29(21):2722–2728, 2013.

[167] Badri Adhikari, Jackson Nowotny, Debswapna Bhattacharya, Jie Hou, and Jianlin Cheng. Coneva: a toolbox for comprehensive assessment of protein contacts. *BMC bioinformatics*, 17(1):517, 2016.

[168] Sergey Ovchinnikov, Hahnbeom Park, Neha Varghese, Po-Ssu Huang, Georgios A Pavlopoulos, David E Kim, Hetunandan Kamisetty, Nikos C Kyrpides, and David Baker. Protein structure determination using metagenome sequence data. *Science*, 355(6322):294–298, 2017.

[169] Badri Adhikari and Jianlin Cheng. Improved protein structure reconstruction using secondary structures, contacts at higher distance thresholds, and non-contacts. *BMC bioinformatics*, 18(1):380, 2017.

[170] Tuan Trieu and Jianlin Cheng. Large-scale reconstruction of 3d structures of human chromosomes from chromosomal contact data. *Nucleic acids research*, 42(7):e52–e52, 2014.

[171] Tuan Trieu and Jianlin Cheng. 3d genome structure modeling by lorentzian objective function. *Nucleic acids research*, 45(3):1049–1058, 2016.

[172] Yang Zhang. I-tasser server for protein 3d structure prediction. *BMC bioinformatics*, 9(1):40, 2008.

[173] Eric F Pettersen, Thomas D Goddard, Conrad C Huang, Gregory S Couch, Daniel M Greenblatt, Elaine C Meng, and Thomas E Ferrin. Ucsf chimeraa visualization system for exploratory research and analysis. *Journal of computational chemistry*, 25(13):1605–1612, 2004.

[174] LLC Schrodinger. The pymol molecular graphics system. *Version*, 1(5):0, 2010.

[175] Roger Sayle. Rasmol v2. 5. *Molecular Visualization Program*, 1992.

[176] C. Vogel, M. Bashton, N. D. Kerrison, C. Chothia, and S. A. Teichmann. Structure, function and evolution of multidomain proteins. *Curr Opin Struct Biol*, 14(2):208–16, 2004.

[177] D. A. Korasick and J. M. Jez. *Protein Domains: Structure, Function, and Methods*, pages 91–97. Academic Press, Waltham, 2016.

[178] S. J. Wheelan, A. Marchler-Bauer, and S. H. Bryant. Domain size distributions can predict domain boundaries. *Bioinformatics*, 16(7):613–8, 2000.

[179] Elmar Krieger, Sander B Nabuurs, and Gert Vriend. Homology modeling. *Methods of biochemical analysis*, 44:509–524, 2003.

[180] Jilong Li, Badri Adhikari, and Jianlin Cheng. An improved integration of template-based and template-free protein structure modeling methods and its assessment in casp11. *Protein and peptide letters*, 22(7):586–593, 2015.

[181] David E Kim, Dylan Chivian, and David Baker. Protein structure prediction and analysis using the robetta server. *Nucleic acids research*, 32(suppl_2):W526–W531, 2004.

[182] T. M. Cheng, T. L. Blundell, and J. Fernandez-Recio. Structural assembly of two-domain proteins by rigid-body docking. *BMC Bioinformatics*, 9:441, 2008.

[183] Y. Inbar, H. Benyamini, R. Nussinov, and H. J. Wolfson. Combinatorial docking approach for structure prediction of large proteins and multi-molecular assemblies. *Phys Biol*, 2(4):S156–65, 2005.

[184] S. Lise, A. Walker-Taylor, and D. T. Jones. Docking protein domains in contact space. *BMC Bioinformatics*, 7:310, 2006.

[185] Carol A Rohl, Charlie EM Strauss, Kira MS Misura, and David Baker. Protein structure prediction using rosetta. *Methods in enzymology*, 383:66–93, 2004.

[186] Dong Xu, Lukasz Jaroszewski, Zhanwen Li, and Adam Godzik. Aida: ab initio domain assembly for automated multi-domain protein structure prediction and domaindomain interaction prediction. *Bioinformatics*, 31(13):2098–2105, 2015.

[187] Adam Belsom, Michael Schneider, Oliver Brock, and Juri Rappsilber. Blind evaluation of hybrid protein structure analysis methods based on cross-linking. *Trends in biochemical sciences*, 41(7):564–567, 2016.

[188] John Moult, Krzysztof Fidelis, Andriy Kryshtafovych, Torsten Schwede, and Anna Tramontano. Critical assessment of methods of protein structure prediction (casp)round xii. *Proteins: Structure, Function, and Bioinformatics*, 86:7–15, 2018.

[189] Tadeusz L Ogorzalek, Greg L Hura, Adam Belsom, Kathryn H Burnett, Andriy Kryshtafovych, John A Tainer, Juri Rappsilber, Susan E Tsutakawa, and Krzysztof Fidelis. Small angle xray scattering and crosslinking for data assisted protein structure prediction in casp 12 with prospects for improved accuracy. *Proteins: Structure, Function, and Bioinformatics*, 86:202–214, 2018.

[190] K. N. Dyer, M. Hammel, R. P. Rambo, S. E. Tsutakawa, I. Rodic, S. Classen, J. A. Tainer, and G. L. Hura. High-throughput saxs for the characterization of biomolecules in solution: a practical approach. *Methods Mol. Biol.*, 1091:245–58, 2014.

[191] M. A. Graewert and D. I. Svergun. Impact and progress in small and wide angle x-ray scattering (saxs and waxs). *Curr Opin Struct Biol*, 23(5):748–54, 2013.

[192] G. L. Hura, A. L. Menon, M. Hammel, R. P. Rambo, 2nd Poole, F. L., S. E. Tsutakawa, Jr. Jenney, F. E., S. Classen, K. A. Frankel, R. C. Hopkins, S. J.

Yang, J. W. Scott, B. D. Dillard, M. W. Adams, and J. A. Tainer. Robust, high-throughput solution structural analyses by small angle x-ray scattering (saxs). *Nat. Methods*, 6(8):606–612, 2009.

[193] A. T. Tuukkanen, A. Spilotros, and D. I. Svergun. Progress in small-angle scattering from biological solutions at high-brilliance synchrotrons. *IUCrJ*, 4(Pt 5):518–528, 2017.

[194] D. A. Korasick and J. J. Tanner. Determination of protein oligomeric structure from small-angle x-ray scattering. *Protein Sci*, 27(4):814–824, 2018.

[195] Marcelo Augusto Dos Reis, Ricardo Aparicio, and Yang Zhang. Improving protein template recognition by using small-angle x-ray scattering profiles. *Biophysical journal*, 101(11):2770–2781, 2011.

[196] Brian Jimnez-Garca, Carles Pons, Dmitri I Svergun, Pau Bernad, and Juan Fernndez-Recio. pydocksaxs: proteinprotein complex structure by saxs and computational docking. *Nucleic acids research*, 43(W1):W356–W361, 2015.

[197] Keehyoung Joo, Seungryong Heo, InSuk Joung, Seung Hwan Hong, Sung Jong Lee, and Jooyoung Lee. Dataassisted protein structure modeling by global optimization in casp12. *Proteins: Structure, Function, and Bioinformatics*, 86:240–246, 2018.

[198] Tadeusz L Ogorzalek, Greg L Hura, Andriy Kryshtafovych, John A Tainer, Krzysztof Fidelis, and Susan E Tsutakawa. Small angle x-ray scattering for data-assisted structure prediction in casp12 with prospects to improve accuracy. *Biophysical Journal*, 114(3):576a–577a, 2018.

[199] Michael L Tress, Iakes Ezkurdia, and Jane S Richardson. Target domain definition and classification in casp8. *Proteins: Structure, Function, and Bioinformatics*, 77(S9):10–17, 2009.

[200] Adam Liwo, St Odziej, Matthew R Pincus, Ryszard J Wawak, Shelly Rackovsky, and Harold A Scheraga. A unitedresidue force field for offlattice proteinstructure simulations. i. functional forms and parameters of longrange sidechain interaction potentials from protein crystal data. *Journal of computational chemistry*, 18(7):849–873, 1997.

[201] Ken A Dill. Dominant forces in protein folding. *Biochemistry*, 29(31):7133–7155, 1990.

[202] Richard A George and Jaap Heringa. An analysis of protein domain linkers: their classification and role in protein folding. *Protein Engineering, Design and Selection*, 15(11):871–879, 2002.

[203] Dina Schneidman-Duhovny, Michal Hammel, and Andrej Sali. Foxs: a web server for rapid computation and fitting of saxs profiles. *Nucleic acids research*, 38(suppl_2):W540–W544, 2010.

[204] Dina Schneidman-Duhovny, Michal Hammel, John A Tainer, and Andrej Sali. Accurate saxs profile computation and its assessment by contrast variation experiments. *Biophysical journal*, 105(4):962–974, 2013.

[205] D Franke, MV Petoukhov, PV Konarev, A Panjkovich, A Tuukkanen, HDT Mertens, AG Kikhney, NR Hajizadeh, JM Franklin, and CM Jeffries. Atsas 2.8: a comprehensive data analysis suite for small-angle scattering from macromolecular solutions. *Journal of applied crystallography*, 50(4):1212–1225, 2017.

[206] Haiguang Liu and Peter H Zwart. Determining pair distance distribution function from saxs data using parametric functionals. *Journal of structural biology*, 180(1):226–234, 2012.

[207] Daniel Russel, Keren Lasker, Ben Webb, Javier Velzquez-Muriel, Elina Tjioe, Dina Schneidman-Duhovny, Bret Peterson, and Andrej Sali. Putting the pieces

together: integrative modeling platform software for structure determination of macromolecular assemblies. *PLoS biology*, 10(1):e1001244, 2012.

[208] Piotr Rotkiewicz and Jeffrey Skolnick. Fast procedure for reconstruction of fullatom protein models from reduced representations. *Journal of computational chemistry*, 29(9):1460–1465, 2008.

[209] DI Svergun. Determination of the regularization parameter in indirect-transform methods using perceptual criteria. *Journal of applied crystallography*, 25(4):495–503, 1992.

[210] Jianlin Cheng, Arlo Z Randall, Michael J Sweredoski, and Pierre Baldi. Scratch: a protein structure and structural feature prediction server. *Nucleic acids research*, 33(suppl˙2):W72–W76, 2005.

VITA

Jie Hou was born in Pingyang county of ZheJiang Province, China. He received his Bachelor's degree from Shanghai Maritime University at 2012, and M.A in Statistics from the University of Missouri-Columbia at 2014. He started his Ph.D. studies in the Department of Electrical Engineering and Computer Science at University of Missouri-Columbia in the fall of 2014. He plays a vital role for MULTICOM which ranks 3rd in protein tertiary modeling among all 98 participant groups and ranks No.1 among the protein model quality assessment methods in the 13th Critical Assessment of Techniques Protein Structure Prediction (CASP13) competition at 2018.

He is interested in developing and applying machine learning, data mining techniques to address biomedical problems. His research focuses on bioinformatics, machine learning and data mining. Specifically, his primary research is to develop data-driven computational methods (e.g., machine learning methods, deep learning techniques, and computational optimization methods) to predict protein tertiary structures from sequences and evaluate the quality of protein structural models. His secondary research is to develop data mining methods to analyze omics (e.g., RNA-seq transcriptomics and genomics) data to study genes and gene networks.