METHODS FOR PROTEIN STRUCTURE PREDICTION

A Thesis presented to
the Faculty of the Graduate School
at the University of Missouri

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Computer Science

by
ZHIQUAN HE
Dr. Dong Xu, Thesis Supervisor
May 2014

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

METHODS FOR PROTEIN STRUCTURE PREDICTION

Zhiquan He,

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

Dr. Dong Xu

Dr. Ioan Kosztin

Dr. Jianlin Cheng

Dr. Yi Shang

ACKNOWLEDGMENTS

First and foremost, I sincerely thank my advisor Dr. Dong Xu for his continuous support and never-ending patience for my PhD study during the years. His broad knowledge, in-depth of guidance and penetrating thoughts fundamentally changed my research methodology and the way of thinking about problems, which drove and will drive my life and career forward.

My great thanks must go to Dr. Yi Shang, Dr. Ioan Kosztin and Dr. Jianlin Cheng for their insightful discussions and collaborative efforts over the years, which greatly helped me solve problems and keep my research forward.

Also I thank the colleagues Jingfen Zhang, Yang Xu, Jianjiong Gao, Chao Zhang, Qiuming Yao, Trupti Joshi, Wenji Ma, Qingguo Wang, Xiaohu Shi, Han Wang for their generous help on my PhD study and personal life. I cherish the friendship with them.

Nothing can show how thankful I am to my families, especially my wife Jinxiang Sun. Without their unconditional love, understanding and unceasing efforts to support, I would not have completed my PhD study.

Over the years of study, I have been continuously supported by National Institutes of Health grants R21/R33-GM078601 and R01-GM100701. Some of the work was also supported by Shumaker Bioinformatics Scholarship, which covered the year of 2011.

Powerful computing servers and clusters are great and reliable resources in University of Missouri, without which my research will not be able to go any step further. I thank the University staff, especially the colleagues in Christopher S. Bond Life Sciences Center for their constant technical support.

TABLE OF CONTENTS

APPENDIX

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

ABSTRACT

With large amount of protein sequences generated by genome-sequencing projects, the lack of tertiary structures is a main obstacle to fully understanding the functions of these proteins. Traditionally, experimental determination of protein structures has utilized both X-ray crystallography and nuclear magnetic resonance (NMR), which are time consuming and costly. Computational structure prediction from amino acid sequence is a viable solution. Recent reviews showed that predicted models of different qualities can be used in various applications from drug design to helping predict protein functions.

Although several decades of efforts have been made to push protein structure prediction forward, it is still challenging nowadays. The major reason for this is the difficulty to capture the fundamental relationship between protein sequences and structures, especially when the sequence similarities among proteins are relatively low. The widely used method for protein structure prediction is comparative protein modeling, which heavily relies on fold recognition performance and alignment accuracy. Another step in protein structure prediction is the structural assessment for predicted protein structures, which obviously plays a critical role.

In this thesis, we discussed several methods for protein structure prediction to address the two important issues. The corresponding tools have been applied in our in-house protein structure prediction platform (MUFOLD). More specifically, we implemented a protein sequence alignment tool which is based on Conditional Random Field and improved its alignment quality by incorporating more complex scoring models. After deeper study of fold recognition and alignment problem, we

proposed a new protocol to improve the quality of sequence profiles, which intrinsically affects the performance of fold recognition and alignment accuracy.

Besides this, several machine learning methods have been proposed to combine knowledge scoring functions and consensus methods from different perspectives for structural quality assessment purpose. For example, graphical probability models such as Hidden Markov Model and Conditional Random Field have been used to combine sequence and structural features to predict the structural quality of predicted protein models. These tools have demonstrated good performance in discriminating protein models of different qualities.

Chapter 1

Introduction

1.1 Introduction to Protein Structure Prediction

Proteins are large biological molecules consisting of one or more chains of amino acids performing a vast array of functions within living organisms. Proteins differ from one another primarily in their sequence of amino acid, which make proteins folding into different and unique three dimensional structures. The functionalities of proteins are determined by their structures.

Massive amounts of protein sequence data are produced by modern large-scale DNA sequencing efforts such as the Human Genome Project. The lack of tertiary structures is a main obstacle to fully understand the functions of these proteins. Traditionally, experimental determination of protein structures has utilized both X-ray crystallography and nuclear magnetic resonance (NMR), which are time consuming and costly. As of January 21, 2014, there are in total 37,371,278 protein sequences available from http://www.ncbi.nlm.nih.gov/RefSeq/, however, only 98,285 structures have been resolved and deposited in RCSB Protein Data Bank database (PDB) [1], since January 1, 1972.

Computational structure prediction from amino acid sequence is a viable solution for this situation, which has been a hot research topic for more than two decades and still remains challenging in order to improve the accuracy and lower the computation [2, 3]. Recent reviews illustrated the applications of predicted protein models with different qualities [4, 5]. For example, high-resolution models with root mean square deviation (RMSD) of 1 to 1.5$\dot{A}$ are useful for almost any application, including drug design; and even if the model quality decreases to about 6$\dot{A}$ RMSD, the function of the protein could still be predicted thereby enabling prediction methods like mutagenesis to be designed based on the model.

Basically, there are two different types of methods for protein structure prediction, comparative protein modeling and Ab initio modeling. Comparative protein modeling uses solved PDB structures as templates and assumes that homologous proteins will share similar structures. This is effective as more and more protein structure have been solved and deposited into PDB database. Structure prediction methods such as Robetta [6], I-TASSER [7, 8], and MUFOLD [9] and Modeller [10] with alignments belong to this category. Ab initio- or de novo- protein modeling method goes a different way to build three-dimensional protein models "from scratch", which is based on physical principles and energies learned from previously solved structures. This method suffers from a computation bottleneck when the protein is relatively large, for example, longer than 100 amino acids. Rosetta [11] is a typical tool for ab initio modeling.

Currently, almost all protein structure prediction methods, comparative protein modeling and ab initio modeling, adopt a sampling-selection strategy. This is reasonable as it always increases the chance of producing better candidate models by sampling. With this strategy, the first step is to generate a large number of candidate models with a sampling procedure; and the second step is to apply a scoring method to identify the most native-like conformations. For this protocol to work, it is required that the sampling procedure is capable of producing at least some near-native conformations and the scoring method is able to identify native-like structures from the structural model pool. Accordingly, this makes protein structure quality assessment (QA) play an important role in protein structure prediction.

The basic components of comparative protein modeling method is shown Figure 1.1, which sequentially include

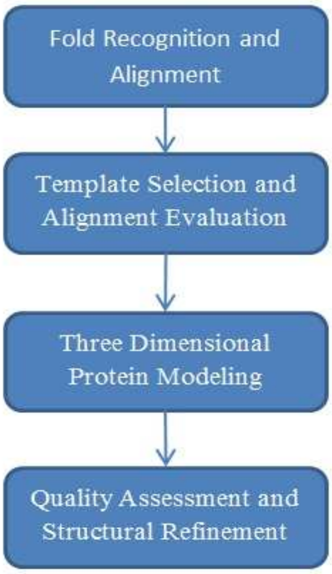
Fold Recognition and Alignment
Template Selection and Alignment Evaluation
Three Dimensional Protein Modeling
Quality Assessment and Structural Refinement

Figure 1.1: Flowchart of Comparative Protein Modeling

1. Fold recognition and alignment generation: search the query sequence against

PDB [1] database using alignment tool such as PSI-BLAST [12] or HHSearch [13]. Fold recognition is about the selection or ranking of protein templates from PDB. And the alignment is the residue mapping between the query and template protein.

2. Template selection and alignment evaluation: select the best templates and alignments using the confidence scores such as the E-Value given by PSI-BLAST or HHSearch.

3. Three-dimensional structure modeling: input the alignments to the model generation module such as Modeller [10] to build the three dimensional structures.

4. Model quality assessment (QA) and refinement. This step is to select the most near-native structures from the candidate pool and refine the selected protein models by solving their steric conflicts and making them more protein like.

1.2 Methods Developed for Protein Structure Prediction

Each of the steps plays its critical role for the overall prediction performance. In this dissertation, we present several methods for protein structure prediction that have been applied to the framework of MUFOLD [9] system.

Fold recognition and alignment between the query sequence and template structure is the first step in protein structure prediction, which is often involved together as alignment based template selection is more straightforward and accurate. Currently, the most accurate alignment and fold recognition tool is HHSearch [13] and conditional random field (CRF) based threading tool [14, 15]. We implemented our own CRF threading tool and improved it by incorporating a more complex model and tuned the tool to its best performance by optimizing its parameters and input features. This tool serves as an alignment platform for continuous improvement in future development.

Better alignment sensitivity and accuracy requires good quality of sequence profile as it is the one of the indispensable features used in current fold recognition and alignment methods. Currently, the default procedure to build sequence profile is to use PSI-BLAST to iteratively and incrementally search a sequence database, such as non-redundant sequence database (NR). However, there exists several problems of this method such as inclusion of non-homologous sequences into the hit list and domain shifting issue which means the query domains are aligned to non-conserved regions and extended to neighboring domains in template protein. In order to overcome these issues, we proposed and implemented a new procedure to improve the sequence profile quality based on Pfam [16], which is an domain annotated sequence database.

The step of quality assessment (QA) of predicted structures is the second step of sampling-selection strategy. A lot of work has been done in the area, particularly the development of knowledge based scoring functions such as OPUS-CA [17], DFIRE [18] and the efforts to combine these scoring functions with consensus methods to achieve better QA performance.

As most of the prediction methods adopt a sampling strategy to generate a number of near-native structure candidates, consensus method is the most effective to evaluate the structural quality of decoys. This is specially the case in Critical Assessment of Protein Structure Prediction (CASP) [19], in which all attending groups submit their most native like structures. However, for those "hard" cases for which we cannot find good templates or significant homologous, the sampled decoys are diversely distributed and naive consensus method often fails.

Unlike pure consensus method which uses only geometrical information from decoy structures, knowledge based scoring functions, such as OPUS-CA and DFIRE, consider the sequence and structure relationship and score the models based on the statistical information of structural attributes in known native structures. These scoring functions are widely used in protein structure prediction. However, knowledge-based scoring functions can only reflect some aspects of protein structures. For example, OPUS-CA uses the distance-dependent energies from the C-alpha atoms of a model.

To improve the protein structure quality assessment performance, it is a wise strategy to combine the advantages of knowledge based scoring functions and consensus methods, and avoid their shortcomings. For this purpose, we proposed and implemented several methods for this purpose. For example, in chapter 6, we developed a scheme to combine different knowledge based scoring functions and consensus GDT [20] based on pairwise comparison.

1.3 Thesis Organization

In Chapter 2, we presented a threading alignment method which is based on conditional random field (CRF) and functional gradient tree boosting and compared its alignment quality to the state-of-art alignment methods.

In Chapter 3, we explained the shortcomings of the current sequence profile generation method by PSI-BLAST and proposed a new protocol to improve the sequence profile quality based on PSI-BLAST and Pfam database.

Chapters 4, 5, 6, 7 and 8 talked about the methods we have developed for protein structure quality assessment (QA). Specifically, Chapter 4 gave the introduction of QA problem in protein structure prediction. Chapter 5 and 6 proposed two new methods to combine naive consensus GDT [20] method and knowledge based scoring functions for QA purpose. In Chapters 7 we discussed about a new Hidden Markov Model (HMM) to capture the sequence and structure compatibility, which can be used to as a scoring function for single structure quality assessment. And Chapter 8 presented a new QA method which used our in-house CRF framework to combine the protein sequence and structural features and consensus GDT information to predict the actual distance of the decoy to its native structure.

Finally, Chapter 9 summarized the work in the thesis and discussed some ideas or directions for future development.

Chapter 2

Protein Threading Alignment Based on Conditional Random Field and Functional Gradient Tree Boosting

2.1 Introduction

In template based protein modeling, an alignment between the query sequence and template structure is the starting point, which has great impact to the performance of this method. The alignment quality varies with the sequence similarity between the two proteins. When the sequence similarity is low, it is difficulty to construct accurate alignments. Researchers have shown that alignment quality drops rapidly when two protein share less than 25% sequence identity [21, 22]. Therefore, the design of scoring functions and optimal alignment algorithms have been extensively studied to improve the alignment quality for less similar proteins. The major difference among most of current alignment methods is the scoring function, (along with the features it used), which will be optimized by the dynamic programming procedure.

A number of alignment methods are based on the comparison between sequences or sequence profiles, such as BLAST or PSI-BLAST [12, 23, 24, 21]. These pure sequence profile based alignment methods have been very successful in both fold recognition, i.e. identifying correct template, and alignment accuracy, especially when the sequence similarity of proteins is not low. However, it has been demonstrated that incorporating structural information into the alignment model can bring further improvement. For example, PROSPECT [25, 26] and RAPTOR [27, 28, 29] use structure information such as secondary structure, solvent accessibility and contact capacity. SPARKS [30] uses additional residue depth information. FUGUE [31] and GenTHREADER [32] makes use of structural profile derived from structure alignments.

Most of the methods mentioned above use a scoring function that is a linear combination of several scoring terms. HHSearch [13], which is based on the comparison of profile Hidden Markov Models (HMM), is a typical alignment method to be compared with in alignment benchmark. In HHSearch, one alignment between two proteins is a co-emission path of the two corresponding profile HMM, and the raw score of the alignment is the probability of the co-emission path. Also, in HHSearch, secondary structure information is added to the model, which significantly improves the performance. However, Hidden Markov method has its own limitations from the perspective of learning power. Conditional Random Field (CRF) as a modern extension of HMM was proposed by John Lafferey in [33]. For the protein threading problem, CONTRAlign [34] implemented CRF model, where the internal feature function is linear combination of several terms. In [15, 35], a more complex CRF model for protein alignment was developed based on gradient tree boosting training method proposed by [36]. In this model, the feature function is a non-linear function represented by a regression tree.

We have implemented our own threading alignment tool which is similar to the work of [15]. We improved the performance by incorporating a more complex scoring model and optimizing the underlying parameters and features. This tool has achieved significant better alignment accuracy when compared to HHSearch [13] and servers as a platform for continuous improvement for future development.

2.2 Method

2.2.1 Tree CRF Model for Protein Threading Alignment

Conditional Random Field (CRF) is a probabilistic graphical model that has been widely used in sequence labeling problem. In protein threading scenario, for a given pair of proteins, their sequence and structure features are regarded as known observations and the alignment between them is regarded as the label sequence.

Let Q and T respectively denote the query sequence and template structure and the corresponding sequence and structure features. For both Q and T, the sequence information such as sequence profile by PSI-BLAST, predicted secondary structure by PSIPRED [37, 38] are available. The structure information is available only for T. Let the alphabet of $\Sigma = \{M, I_q, I_t\}$ be the set of all possible alignment state, where M means two positions from query and template are matched; I_q means an insertion of amino acid occurs in query protein and I_t means an insertion in template protein. An alignment of length L between Q and T is a sequence of alignment state denoted by $a = [a_1, a_2, .., a_L], a_i \in \Sigma$. CRF model for threading defines the conditional probability of a given Q and T as follows

$$P_\theta(a|Q,T) = \frac{\sum_{i=1}^{L} F(a_{i-1}, a_i, Q, T)}{Z(Q,T)} \tag{2.1}$$

where $Z(Q,T)$ is a normalization factor to make the right hand part be a probability and $F(a_{i-1}, a_i, Q, T)$ is the potential function that indicates the likelihood of the alignment state at alignment position i, given the input query sequence and template structure. There are in total 9 potential functions we need to train as there are total 3 different alignment states. In reality, the potential functions for the transition $I_q \rightarrow I_t$ or $I_t \rightarrow I_q$ are forbidden as these two alignment state transitions are rarely seen in actual alignments. The potential function $F(a_{i-1}, a_i, Q, T)$ in this model is a non-linear function which is represented by a weighted sum of a set of regression trees.

2.2.2 Training Tree CRF by Functional Gradient Boosting

To train this model, we need to calculate the functional gradient of the conditional probability with respect to $F(a_{i-1}, a_i, Q, T)$ [36, 15].

$$\frac{\partial lnP(a|Q,T)}{\partial F(u,v,Q,T)} = I(a_{i-1} = u, a_i = v) - P(a_{i-1} = u, a_i = v|Q,T) \tag{2.2}$$

where $I(,)$ is a indicator function with values of 0 or 1. It is not difficulty to understand the probabilistic meaning of the functional gradient of Eqn. 2.2, for which, an ideal model will make it zero. That means, given a training alignment, if the transition $u \to v$ is observed at alignment position i, the perfect model will have $P(a_{i-1} = u, a_i = v|Q,T) = 1$, otherwise $P(a_{i-1} = u, a_i = v|Q,T) = 0$. So, given an initial model $F(u,v,Q,T)$, in order to maximize $P(a|Q,T)$, we only need to update the current model in the direction of gradient by adding the a new model $\Delta F(u,v,Q,T)$ which fits the gradient by Eqn. 2.2. The model will be in the following form

$$\begin{aligned} F_m(u,v,Q,T) &= F_0(u,v,Q,T) + w_1 \Delta F_1(u,v,Q,T) +, ..., \\ &+ w_m \Delta F_m(u,v,Q,T) \end{aligned} \tag{2.3}$$

where $\Delta F_k(u,v,Q,T) = \frac{\partial ln P(a|Q,T)}{\partial F_{k-1}(u,v,Q,T)}$, which is represented by a regression tree.

Given a CRF model, which is a set of seven $F(u,v,Q,T)$ functions and a training alignment, we can calculate the probability $P(a_{i-1} = u, a_i = v|Q,T)$ using the forward and backward method. In the following, we use i as the position index in query protein and j in template protein. Let's define the forward variable $\alpha(v,i,j)$ to be the combined probability of all alignments up to the positions (i,j), ending in state v and the backward variable $\beta(v,i,j)$ be the combined probability of all alignments starting from the positions $(i+1,j+1)$, assuming that all alignments start from state v. As shown Figure. 2.1, each edge in the trellis matrix corresponds to an alignment state and each path from the left-top to right-bottom is an alignment. Let's define the parents of one edge e as $p(e)$ be the set of adjacent edges in the left-top corner and the children of e as $d(e)$ be the set of adjacent edges in the right-bottom corner. And we also use $ind(e) = (i,j)$ to denote the position of each e. So, we have the

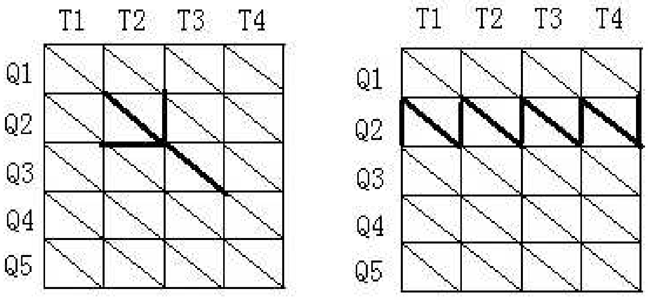

Figure 2.1: Trellis Matrix for CRF

following iterative form

$$\begin{aligned} \alpha(v,i,j) &= P(x_1...x_i, y_1...y_j, s_{ij}=v) \\ &= \sum_{u,u\in p(v)} e^{F(u,v,Q,T)} a(u, ind(u)) \end{aligned} \tag{2.4}$$

and

$$\begin{aligned} \beta(u,i,j) &= P(x_{i+1}...x_{Lq}, y_{j+1}...y_{Lt} | s_{ij}=u) \\ &= \sum_{v,v\in d(u)} e^{F(u,v,Q,T)} \beta(v, ind(v)) \end{aligned} \tag{2.5}$$

For example, if v is a match state

$$\begin{aligned} \alpha(M,i,j) &= e^{F(M,M,Q,T)}\alpha(M,i-1,j-1) \\ &+ e^{F(I_t,M,Q,T)}\alpha(I_t,i,j-1) \\ &+ e^{F(I_q,M,Q,T)}\alpha(I_q,i-1,j) \end{aligned} \tag{2.6}$$

and

$$\begin{aligned}\beta(M,i,j) &= e^{F(M,M,Q,T)}\beta(M,i+1,j+1) \\ &+ e^{F(M,I_t,Q,T)}\beta(I_t,i+1,j+1) \\ &+ e^{F(M,I_q,Q,T)}\beta(I_q,i+1,j+1) \end{aligned} \tag{2.7}$$

Then

$$P(a_{i-1}=u,a_i=v|Q,T) = \frac{a(u,ind(u))e^{F(u,v,Q,T)}\beta(v,ind(v))}{Z(Q,T)} \tag{2.8}$$

The normalizer $Z(Q,T)$ does not depend on the position

$$Z(Q,T) = \sum_{u\in C} a(u,ind(u))\beta(u,ind(u)) \tag{2.9}$$

where C is a set of edge such that every path in the trellis matrix going from the left-top corner to the right-bottom corner goes through the set, for example, the darken line in the right sub-figure in Figure. 2.1.

Given a set of training alignments, the gradient tree boosting based training procedure is shown as follows.

```
Function TreeBoosting(Data)
    $//Data = \{(a_j, Q_j, T_j)\}$, where $j$ indicates the $jth$ training example
    for state transitions $u \rightarrow v$
        initialize $F_0(u,v,) = 0$
    end for


    // training at most M iterations
    for m from 1 to M
        for state transitions $u \rightarrow v$
            $S(u, v) = GenerateGradient(u, v, Data)$
            $T_m(u, v) = FitRegressionTree(S(u, v))$
            $F_m(u, v) = F_{m-1}(u, v) + T_m(u, v)$
            end for
        end for
    return $F_m(u, v)$ as $f_m(u, v, Q, T)$
end function

Function GenerateGradient(u,v,Data)
    for all training alignment $a$
        for each edge in CRF trellis, calculate $\alpha$ and $\beta$
        for each state transition in CRF trellis, calculate gradient using Eqn. 2.2
            $\delta(u, v, Q, T) = I(a_{i-1} = u, a_i = v) - P(a_{i-1} = u, a_i = v|Q, T)$
            insert an example data $(\delta, Q_i, T_j)$ into $S(u, v)$
        end for
    end for
return $S(u, v)$
end function
```

2.2.3 Query-Template Alignment Algorithm

After the CRF model is trained, we can find the best alignment a by maximizing $P(a|Q,T)$, which is done by dynamic programming.

2.2.4 Sequence and Structure Features

Evolutionary and structural information are typically used for protein threading alignment. We generated position specific score matrix (PSSM) for template sequence and position specific frequency matrix (PSFM) for query sequence using PSI-BLAST (released in 2010) with five iterations against the NR database (release in 2010) and the E-Value is set to 0.001. $PSSM(i,a)$ is the mutation potential for amino acid a at template position i and $PSFM(j,a)$ is the occurring frequency of amino acid a at position j in query sequence. Secondary structure of query sequence is predicted by PSIPRED [37, 38]. Secondary structure of template is calculated by DSSP program [39]. We used the following features for all types of state transitions.

1. Sequence profile similarity: sequence profile similarity score between two aligned positions is calculated by $\sum_a PSSM(i,a) \times PSFM(j,a)$.

2. Environmental fitness score: this score measures the propensity of an amino acid type a to appear in a structure type, which is specified by the combinations of three types secondary structure (Helix, Beta sheet, and loop) and three types of solvent accessibility (Fully buried, intermediate and fully exposed) [25, 26]. The environment fitness score is given by $\sum_a PSFM(j,a) \times F(env_i, a)$.

3. Secondary structure (SS) Match Score: suppose the secondary structure type at template position i is SS_d, and a predicted secondary structure for query

sequence is SS_p with confidence value C given by PSIPRED, the matching score is the probability of SS_d predicted to be (SS_p, C), which is specified by a look-up table.

4. Solvent accessibility (SA) matching scores: similarly to secondary structure match score, the predicted SA SA_p is done by SSPro [40] and the true SA SA_d for template is computed using DSSP with cutoff 25% (above which means the exposed state and otherwise the buried state). If the SA state is matched, the score is 1 otherwise 0.

5. Secondary structure type SS_p and SS_d.

6. Solvent accessibility type SA_p and SA_d.

7. Hydrophobic count for query and template sequence with a window of 5 residues centered at each position [34].

2.2.5 Improving the CRF Model

From Eqn. 2.3, we see that the CRF model is a set of regression trees, added together. We improved the model by adding a constant offset to the function and searched for the best weight for each newly added tree at each iteration.

$$\begin{aligned} F_m(u,v,Q,T) &= F_0(u,v,Q,T) + w_1 \Delta F_1(u,v,Q,T) +, ..., \\ &+ w_m \Delta F_m(u,v,Q,T) + b_m \end{aligned} \tag{2.10}$$

where b_m is the offset for $F_m(u,v,Q,T)$.

2.3 Performance

From PDB database, we collected three non-overlaping datasets for training and testing the model. The training dataset contains 50 pairs of proteins, the validation dataset contains 200 pairs and the test dataset contains 447 pairs. The average size of these proteins is about 200 residues. For each pair of proteins, PSI-BLAST fails to generate reasonably good alignments. The true alignments used for training and validation are structure alignments generated by TMAlign [41]. The model performance is evaluated by the average alignment quality which is measured by the GDT score of the alignment using the query protein as reference.

2.3.1 Training Performance

Figure. 2.2 shows the training performance using the model defined by Eqn. 2.3. The blue curve shows the performance of the model at each iteration, and the red curve shoes the performance on the validation dataset. We can see that the model

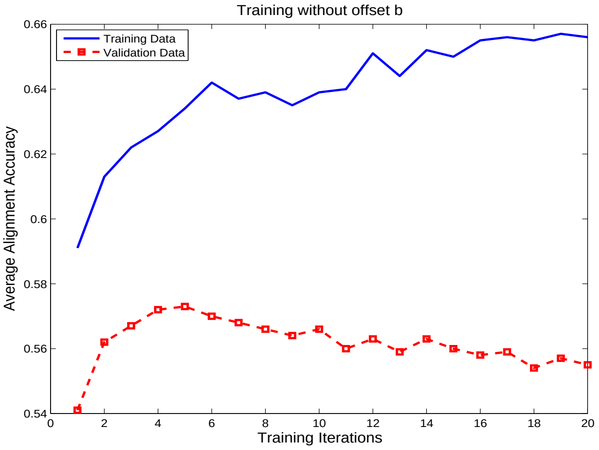

Figure 2.2: CRF Model without offset b

converges very fast. And also, we see that the performance on validation dataset keeps decreasing after it reach the maximum. The reason might be the over fitting problem, although some techniques have been taken to avoid it. Similarly, Figure. 2.3 shows the training performance using the improved model defined by Eqn. 8.4.

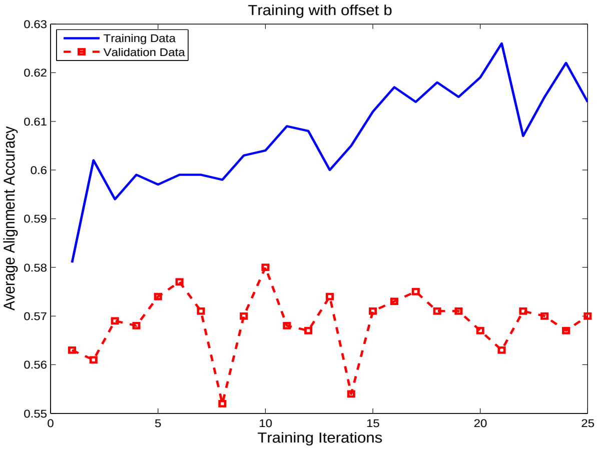

Figure 2.3: CRF Model with offset b

Figure. 2.4 compares the performance on the validation dataset of the two models, one has no the offset term defined in Eqn. 2.3 (blue curve) and the other one with the offset term defined in 8.4 (red curve).

We can see that adding an additional term the to potential function significantly increases the learning capability of the model.

2.3.2 Performance on Testing Dataset

The testing performance of CRF method is compared with HHSearch [13], which is the state-of-art method in terms of alignment accuracy. Figure. 2.5 compares the alignment generated by the structure alignment by CE [42], HHSearch and CRF specified by Eqn. 2.3, which does not contain the offset term. From the figure, we can

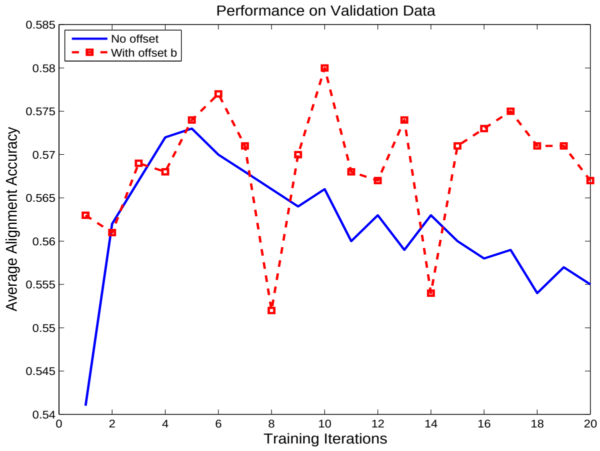

Figure 2.4: Comparison: with offset or not not

see that CRF is better than HHSearch in terms of alignment accuracy with average GDT improvement of $0.356 - 0.339 = 0.017$. Figure. 2.6 shows the comparison when the CRF is modeled by Eqn. 8.4, which has a constant term in the potential function. The performance of CRF method is slightly better, changing from 0.356 to 0.369. Now, the alignment GDT gain of CRF comparing to HHSearch is $0.369 - 0.339 = 0.03$. Figure. 2.7 shows the performance when CRF is modeled by Eqn. 2.3 and Eqn. 8.4 respectively.

Testing on CASP9 Dataset

In order to use CRF threading on CASP9 data set, we need to generate the alignment between the query and each template from PDB database. It is quite time consuming to use CRF to search the entire PDB database as reading features from text files requires heavy IO operations. We can speed up this by rewriting all the features of the database into binary files. And the final time efficiency should be comparable to HHSearch as both do the dynamic programming for one alignment for one pair

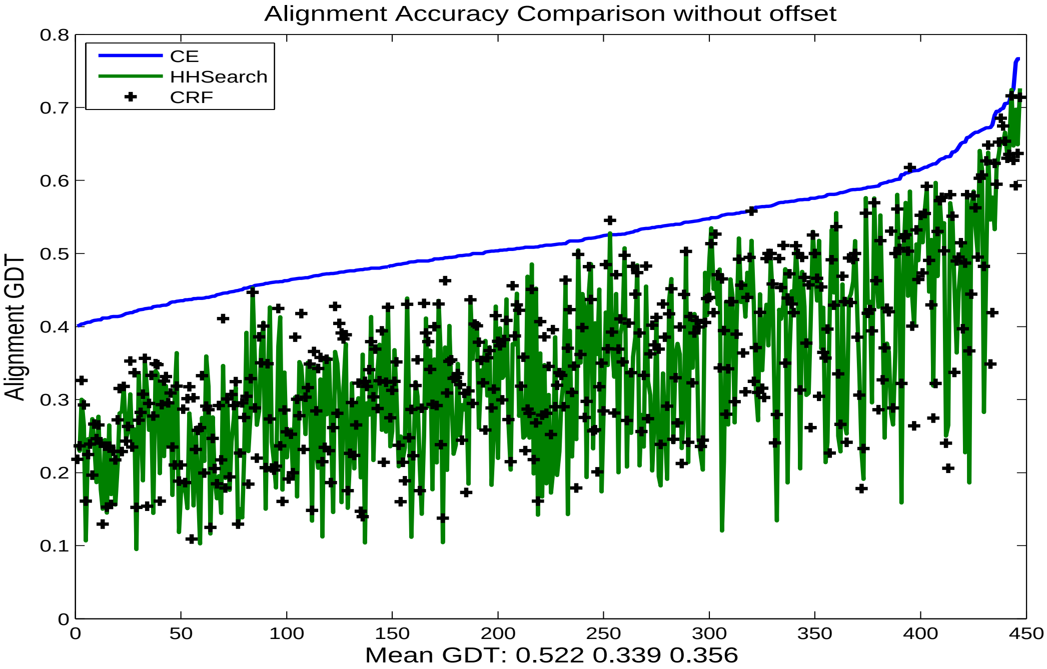

Figure 2.5: Test of CRF without offset term

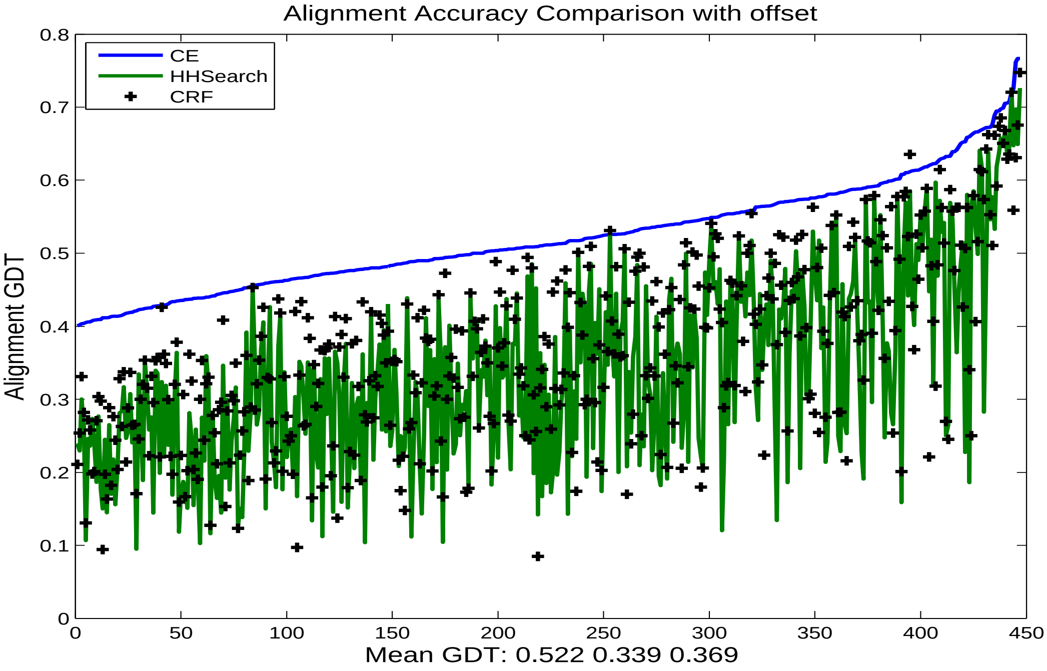

Figure 2.6: Test of CRF with offset term

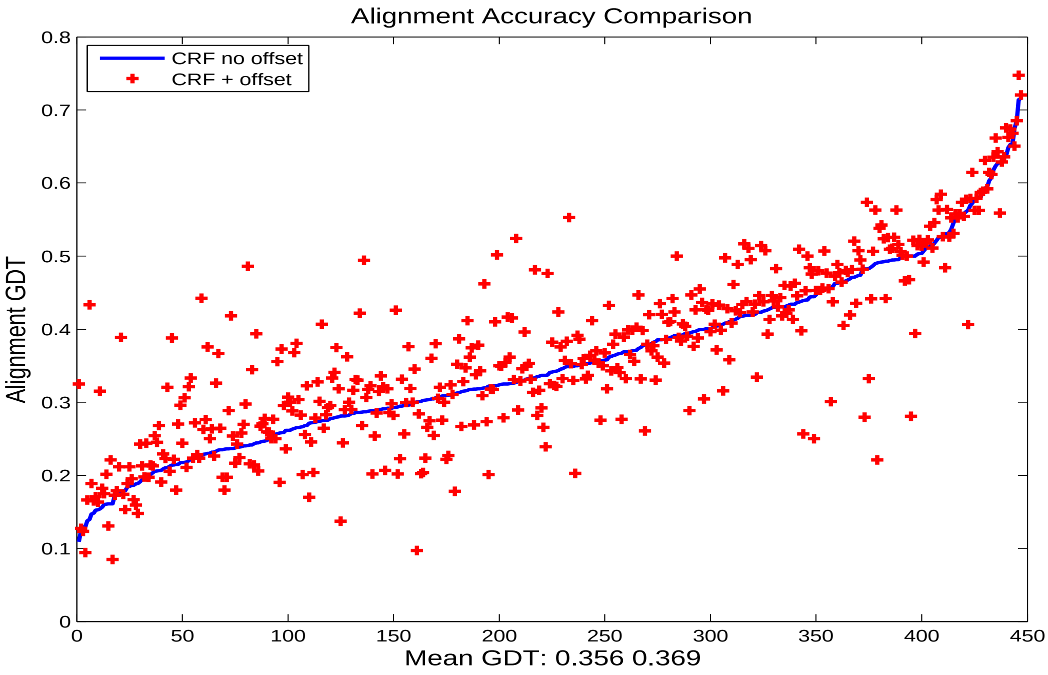

Figure 2.7: Comparison of two CRFs on testing data

of proteins. However, one problem is that we cannot directly use the probability given by Eqn. 2.1 as different pair of proteins have different alignment space and the probabilities associated with different alignment spaces are not comparable. We take the advantage of strong fold recognition capability of HHSearch by redoing the alignments between the query protein and the top templates reported by HHSearch.

Figure. 2.8 shows the average GDT of top-1 alignment for HHSearch, CRF Threading and PSI-BLAST on CASP 9 sequences. As we can see from the figure, HHSearch achieved the average GDT score of 0.492 and CRF threading with HHSearch achieved 0.518, both of which was much higher than that of PSI-BLAST. Figure. 2.9 shows the average GDT of top-5 alignments for HHSearch, CRF Threading with HHSearch and PSI-BLAST. CRF threading with HHSearch achieved almost 3 GDT points hight than that of HHSearch.

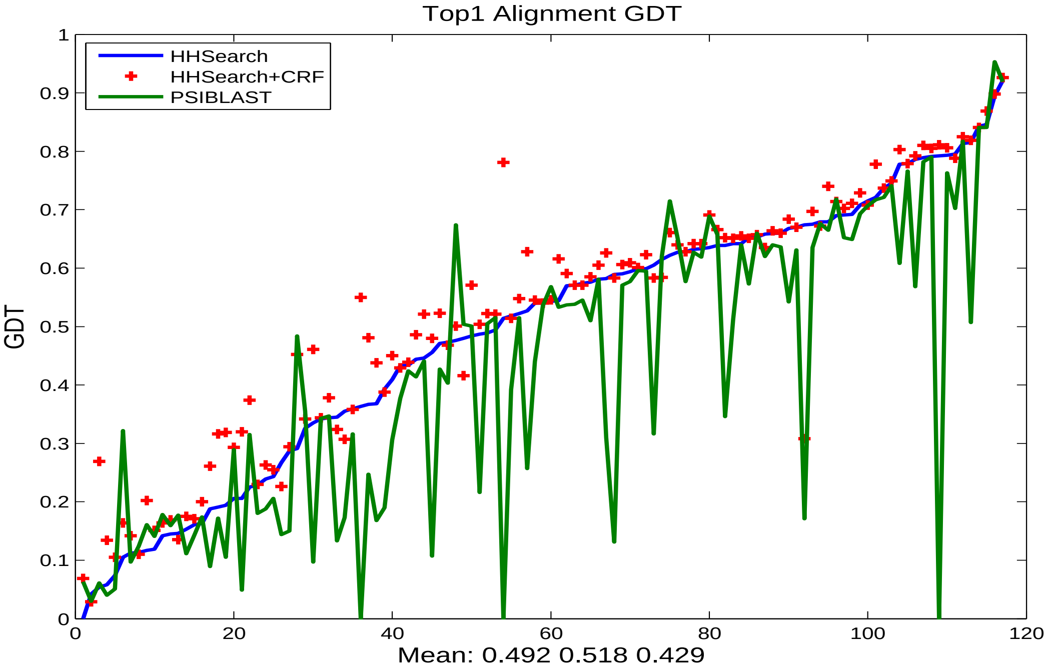

Figure 2.8: Top-1 Alignment Quality

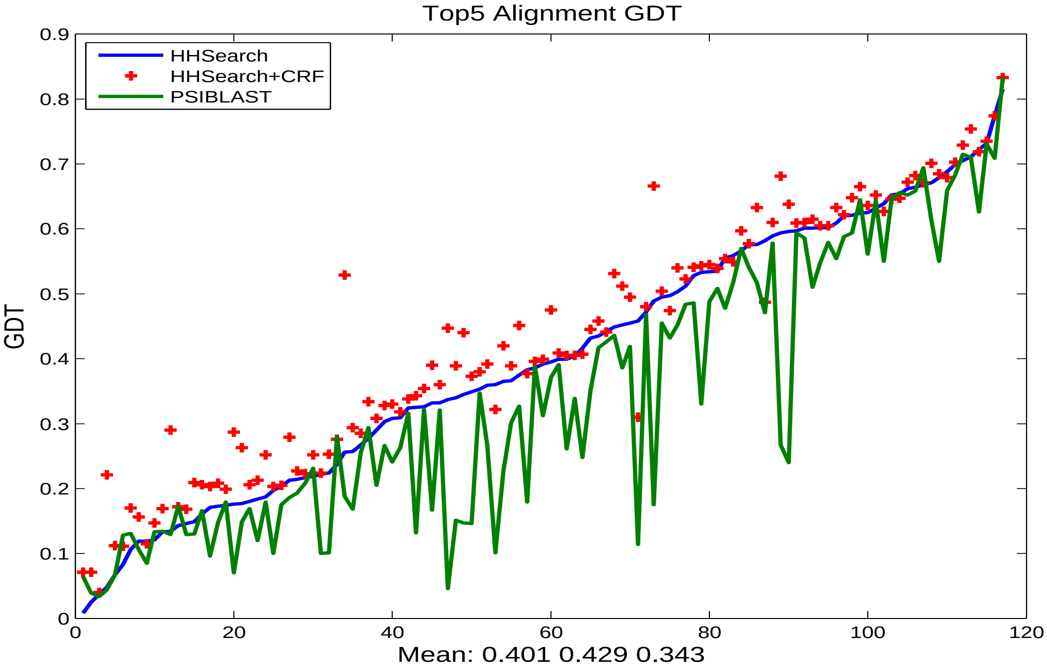

Figure 2.9: Top-5 Alignment Quality

Chapter 3

Improving Sequence Profile Using PFam Database

3.1 Introduction

Fold recognition and the alignment between query sequence and template protein from PDB [1] is the very first step for protein structure prediction. Better alignment and fold recognition performance requires good quality of sequence profiles. Sequence profile is a pattern describing a family of related protein sequences. A closely related concept is protein domain. Proteins are generally composed of one or more domains with different combinations. A protein domain is a conserved part of a given protein sequence and structure that can evolve, function, and exist independently of the rest of the protein chain. Proteins from the same family or supper family usually share some domains. Therefore, sequence profiles need preserve the domain information of the member sequences.

BLAST or PSI-BLAST [12] are widely used to build sequence profiles by searching the query sequence against a Non-redundant sequence database in iterative and incremental manner. PSI-BLAST is much more sensitive than BLAST in detecting distant homologous due to its iterative profile based search strategy. However, PSI-BLAST has made at least two types of errors. One is the relatively high false positive rate, which means non-homologous proteins are included and given high statistical significance [43]. Several work has been done to improve the performance of PSI-BLAST, for example [44] proposed a method to adjust the E-value score using the first round results, which is less corrupted. The second one is the alignment problem, even the subject protein is a homolog to the query protein. Query domains sometimes are aligned to non-conserved regions and extended to neighboring domains [45]. This problem gets more severe when the involving sequences have multiple domains. In [46], a protocol was proposed to clean PSI-BLAST-generated profile of errorneous extension caused by domain insertions for single domain sequences. In this method, the domain boundary at the insertion point was detected to build the alignment.

In this project, we did some analysis on how the sequence profile quality affects the performance of PSI-BLAST and proposed a protocol based on Pfam [16] database and PSI-BLAST to build better sequence profiles.

3.2 Profile Quality Analysis

The most straightforward way to test the quality of a sequence profile is to use it to search against the sequence database using PSI-BLAST. By checking the top hits returned by PSI-BLAST, we can know the performance of PSI-BLAST scoring system and the quality of the input sequence profile. We collected 26002 sequences from released PDB files, among which the mutual sequence identity was less than 70%, and searched each sequence against the sequence database using PSI-BLAST and found that for 813 sequences, the query sequence itself was ranked out of top 10.

selfRank	topAlnIden	selfAlnIden	avgAlnIdenAhead	maxAlnIdenAhead
22.22	0.26	0.99	0.23	0.32

Table 3.1: Average Performance of PSI-BLAST Picking out Query. "selfRank" is the rank of query given by PSI-BLAST; "topAlnIden" is the alignment identity of top hit; "selfAlnIden" is the alignment identity of self alignment; "avgAlnIdenAhead" is the average alignment identity of hits in front of query and "maxAlnIdenAhead" is the maximum alignment identity of front hits.

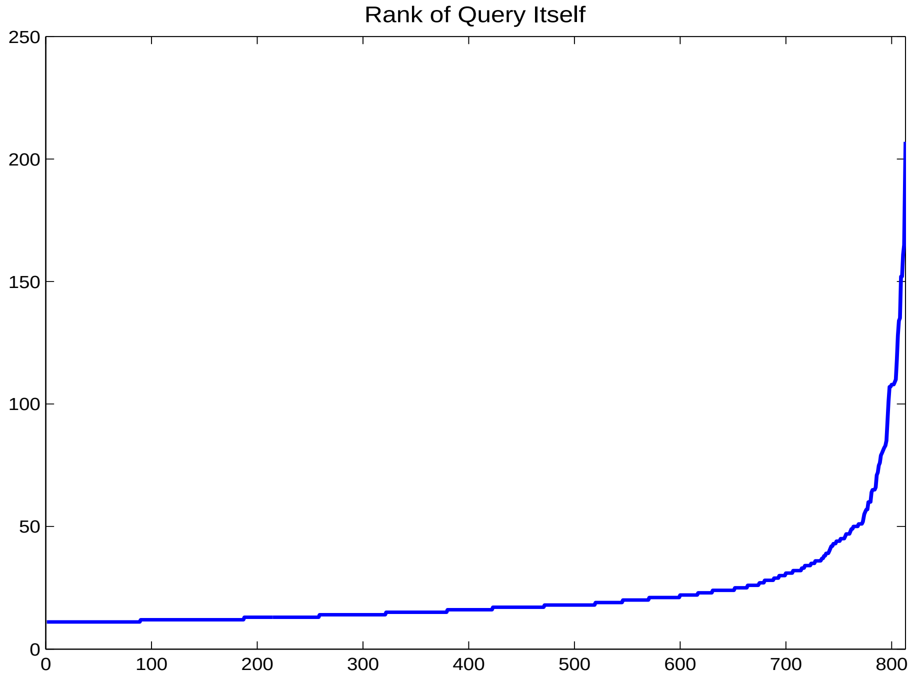

Figure 3.1: Self-Rank Performance of PSI-BLAST.

Table 3.1 showed the performance of PSI-BLAST finding the query itself. Figure. 3.1 showed the rank of query sequence for each test cases. Figure. 3.2 showed the

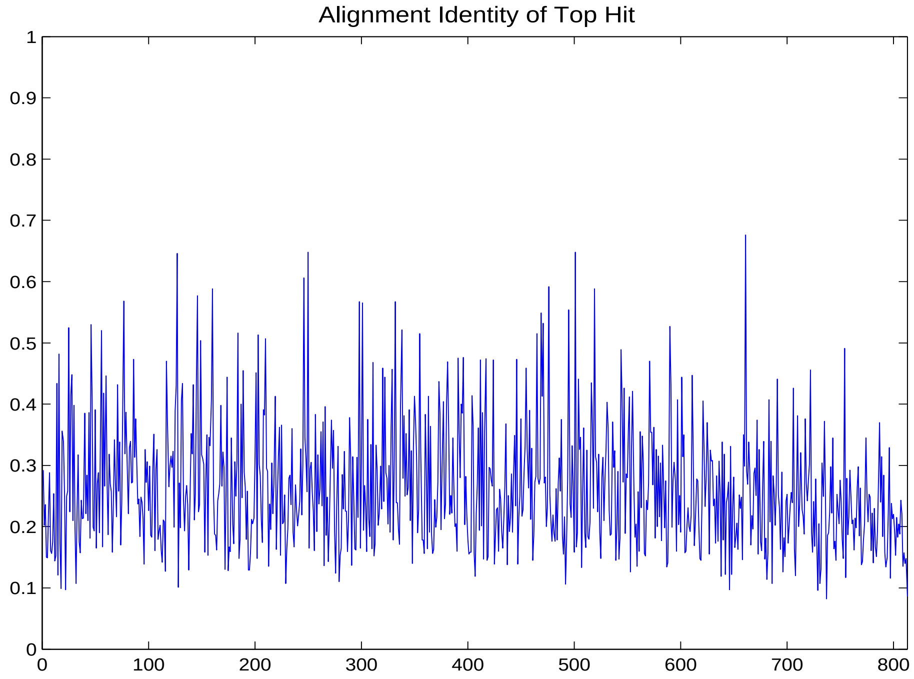

Figure 3.2: Alignment Identity of Top Hit.

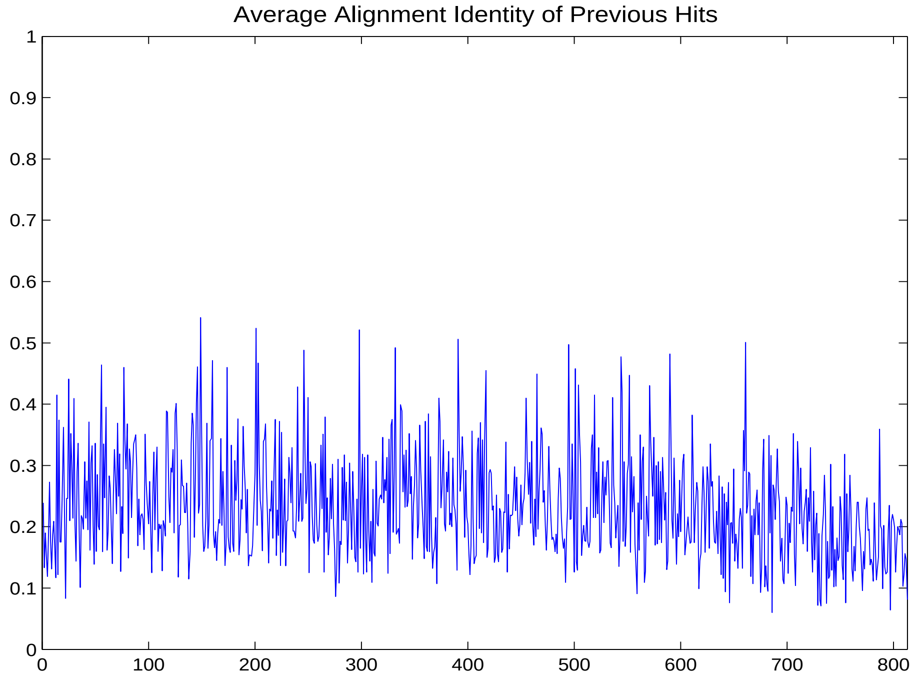

Figure 3.3: Average Alignment Identity of Front Hits.

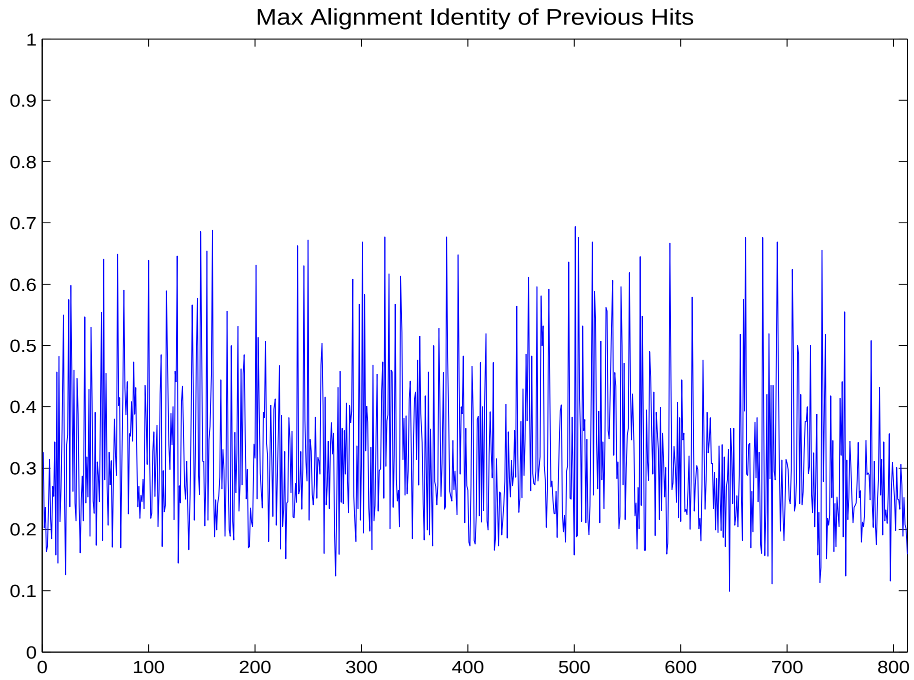

Figure 3.4: Maximum Alignment Identity of Front Hits.

alignment identity of top hit and Figure. 3.3, 3.4 showed the average and maximum alignment identity of those hits ahead of the query in details. From Table 3.1 and Figure. 3.1, we can see that the average self-rank is around 22, which means that PSI-BLAST ranked the query sequence itself at the 22nd place, when searching a sequence database containing the query sequence. And also, Table 3.1 showed that the average and maximum alignment identity of those hits ahead of the query was only 0.23 and 0.32 respectively, which means that those hits may not belong to the same fold of the query. From these observations, we see that the ranking of hits by E-Value had relatively high false positive rate.

For those 813 sequences, we checked how the alignments cover the domains specified by Pfam annotation. If a domain is covered more than 80% of its region by an alignment, we say the domain is hit by the alignment. In total 1708 alignments, Figure 3.5 showed the histogram of the domain coverage ratio which is defined as the ratio between the common domains hit by the alignment and the number of query domains hit by the alignment. From the figure, we can see that for several hundreds of alignments, the ratio is 0, which means in the alignment region, the domains are totally different. One of the reason is because the alignments do not align the two domains together.

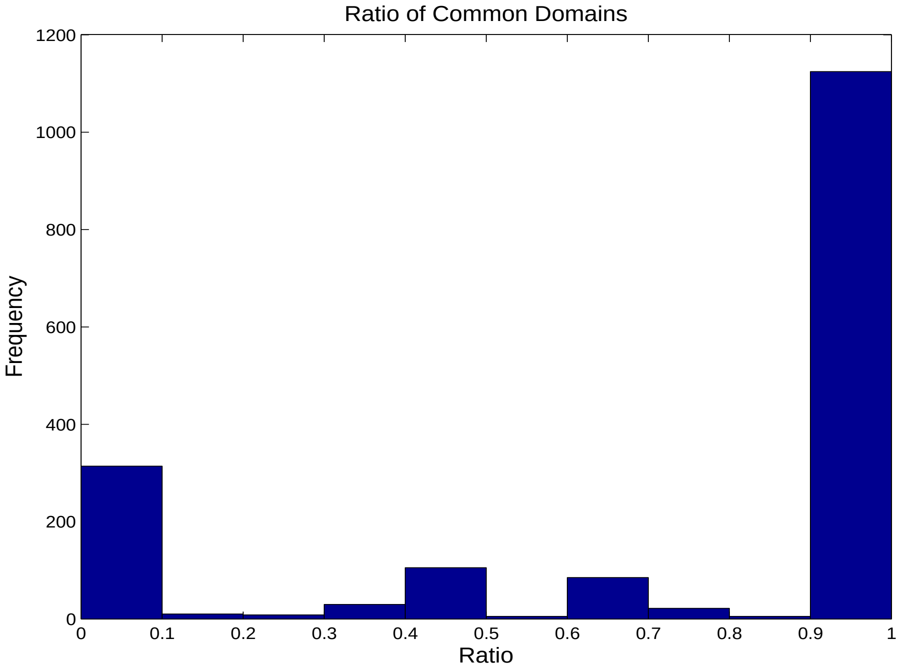

Figure 3.5: Common Domain Ratio in Alignment Region.

From the above observations, the sequence profile generated by PSI-BLAST can be easily corrupted due to the high false positive rate of non-homologous proteins and domain shift problem in the alignments.

3.3 Related Work

In this project, we tried to use a domain-annotated sequence database to build sequence profiles of better qualities. One of the related work is DELTA-BLAST which searches the query sequence against a database of pre-constructed position specific score matrix (PSSM) before searching a sequence database. This yields better and faster homology detection [47]. As shown in Figure 3.6, DELTA-BLAST first uses RPS-BLAST to search the query sequence against a conserved domain database (CDD) to obtain the initial PSSM. In CDD database, each conserved domain (CD) represented by a multiple sequence alignment (MSA) of homologous sequence segments is converted to a PSSM to facilitate efficient search [48]. Then a PSSM is constructed from the multiple alignments of the CDs. The new PSSM can be used

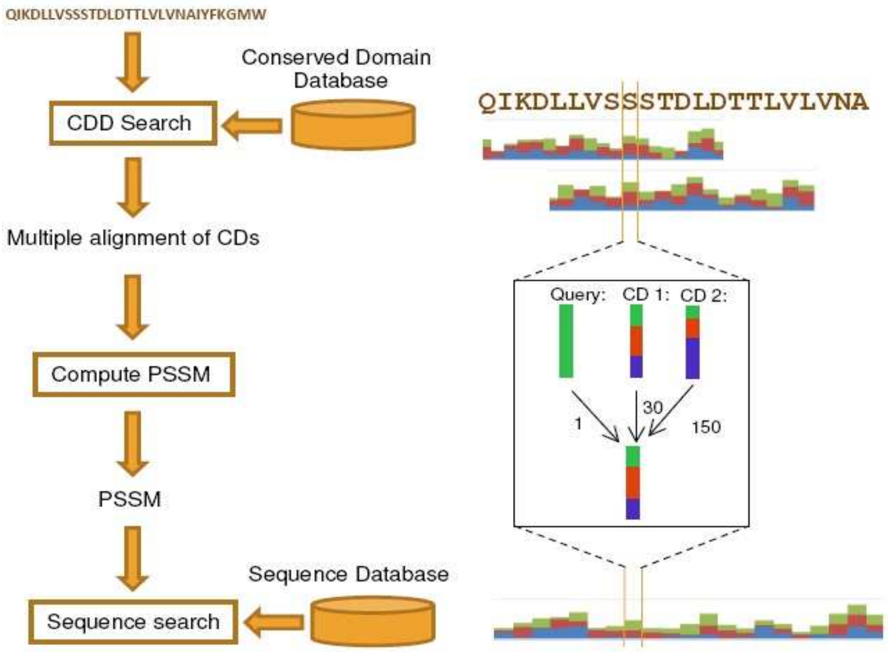

Figure 3.6: DELTA-BLAST Method

by PSI-BLAST to search the sequence database. The right part of Figure 3.6 shows how to combine the multiple alignment of CDs into a PSSM. The PSSM can also be iteratively updated in the way of PSI-BLAST.

DELTA-BLAST is faster and expected to be more sensitive than the original PSI-BLAST. However, the MSA DELTA-BLAST constructs is still a star MSA, which means all the sequences are aligned with the query sequence as a reference. On the other hand, the CDD database is used only once to build the initial PSSM to speed up the entire process. In this project, we introduced the Pfam database into the process of PSSM construction of PSI-BLAST and proposed a new protocol to improve the quality of sequence profiles.

3.4 Pfam Database

The Pfam database is a large collection of protein families, each represented by multiple sequence alignments and hidden Markov models (HMMs). Pfam database contains two main components, Pfam-A and Pfam-B. Pfam-A entries are high quality, manually curated sequence families while Pfam-B is automatically generated and has relatively lower quality.

3.4.1 Structure Distribution of Pfam Families

We scanned the PDB entries which have been assigned Pfam domains to analyze the structure distribution of each Pfam family with the following steps.

1. Get the sequence for each Pfam domain.

2. Remove domains with too few sequences.

3. Remove redundant sequences using CD-Hit [49] with the mutual sequence identity cutoff of 80%.

4. Remove domains with too few sequences again.

We studied the average GDT, standard deviation of GDT and the fold ratio, which is defined as the number of pairs that are of the same fold ($TMScore >= 0.5$) divided by the total number of pairs of structures within the domain annotation. Figure. 3.7 showed the structural distribution of each of 723 Pfam families. From Figure. 3.7,

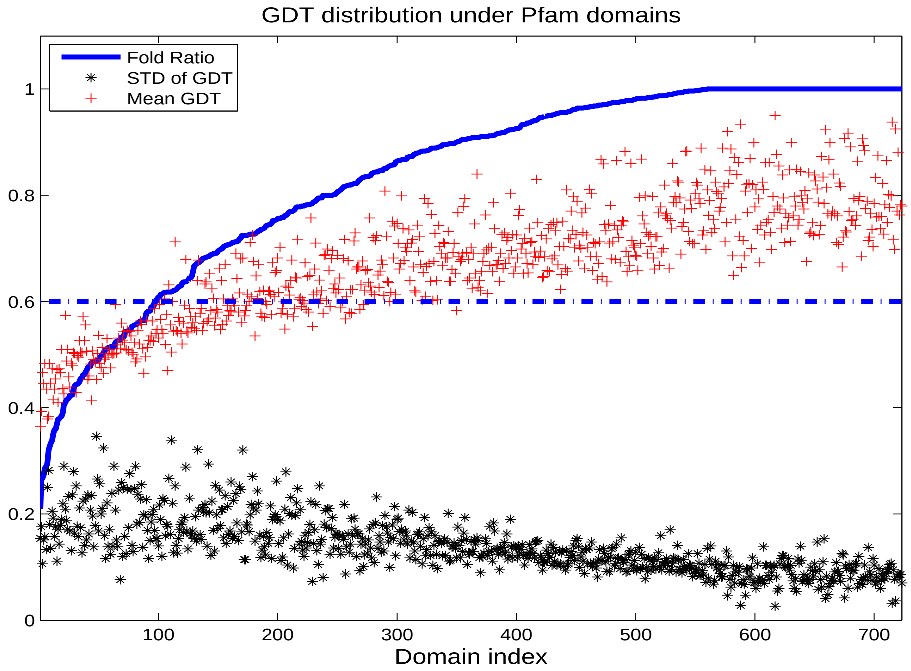

Figure 3.7: Structure Distribution of Pfam Domains. The blue curve is the fold ratio; average GDT of structures pairs from each domain are shown in red points and black stars are the standard deviation of GDT for structures from each domain.

we can see that the structures in most of the domains are closely distributed with relatively high average GDT and low standard deviation.

3.5 A New Protocol to Generate PSSM

The default procedure of PSI-BLAST to generate PSSM is shown in Figure. 3.8 described in the following steps.

1. Search the query sequence against a database such as non-redundant sequence database, such as NR database without PSSM or an empty PSSm, this is equivalent to simple BLAST.

2. Construct a multiple sequence alignment (MSA) and build a PSSM from the list of significant hits or alignments from step 1.

3. Search the query sequence against the database with the PSSM.

4. Construct a new MSA and update the PSSM from the alignments.

5. Go back to step 3 to search again until no new significant hits found.

From the analysis before, we know some of the errors or shortcomings of current protocol PSI-BLAST uses to build sequence profiles. In order to improve the sequence profile quality, we proposed a new protocol to take the advantages of accurate domain information from Pfam database and avoid some of the issues with PSI-BLAST. The Figure. 3.9 shows the protocol that is based on PSI-BLAST and Pfam database.

1. Search the query sequence against the Pfam sequence database.

2. Get the significant hits specified by PSI-BLAST E-Value with cutoff being 0.001

3. Group the alignments according to the domains specified by Pfam annotation.

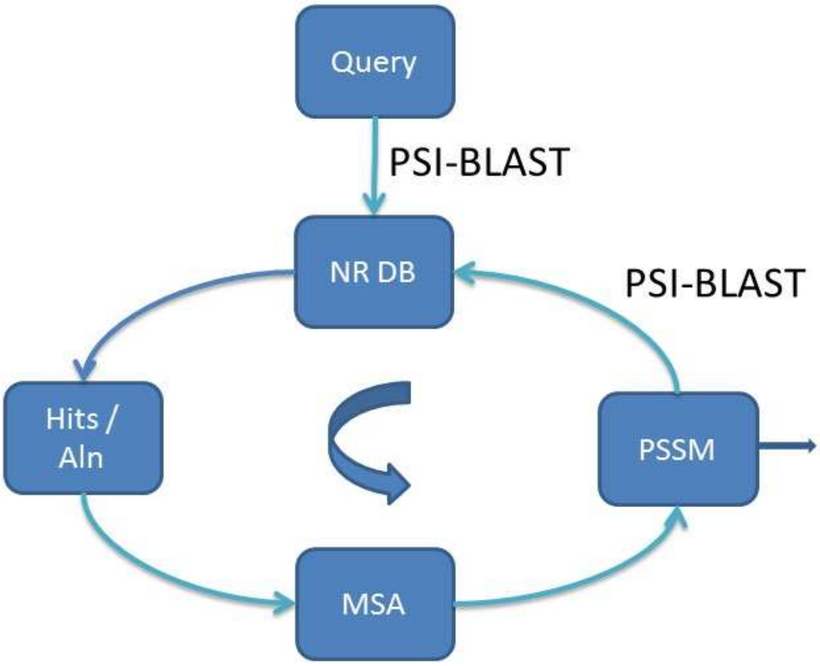

Figure 3.8: PSI-BLAST Default Procedure

4. Filter the domains. If the more than one domains covers the same region, the domain with less number of hits will be removed. And also, if a domain with less than three hits, the whole domain will be removed, as usually, one domain

5. Generate the MSA using the Hidden Markov Model (HMM) of the Pfam domain. This is assuming that the hit sequences are generated by the Hidden Markov Model of the domain sequence profile.

6. Generate the overall PSSM using the MSA from each domain

7. Generate Hidden Markov Model using the overall PSSM for HHSearch.

In the first step, we used PSI-BLAST to search the query sequence against Pfam domain sequence database, instead of non-redundant sequence database, which would

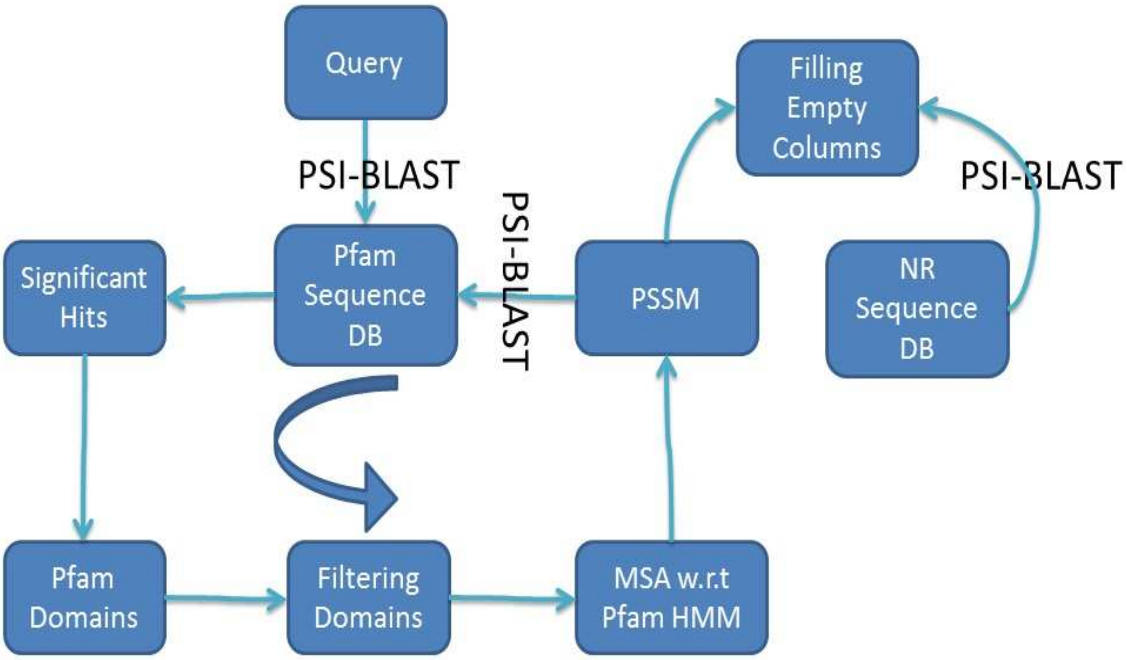

Figure 3.9: PSSM Construction Using PSI-BLAST and Pfam.

produce better sequence alignments as the Pfam domain sequences are of high quality and manually curated according to their domain assignment. In step 3, we grouped the significant hits according to the domains assigned by Pfam database, which greatly reduced the chance of including non-homologous sequences. Step 4 removed those domains that are not consistent with others within the hits returned by PSI-BLAST, which would help improve the alignment quality. In step 5, the HMM models associated with each Pfam domain were used to generate the multiple sequence alignment, which ensured the good quality of the alignments.

3.6 Benchmark Performance

To compare the performance of the new protocol and PSI-BLAST default procedure, we tested the sequence profiles from different protocols in real application by searching it against PDB database and checked the alignment quality and fold recognition performance. We input the sequence profiles from different protocols to HHSearch [13], which converted the sequence profiles into profile HMM models. The quality of different sequence profiles are then compared in terms of the alignment quality and fold recognition performance of HHSearch measured by corresponding GDT scores. The reason for comparing the quality of different sequence profiles this way is that HHSearch is the state-of-art alignment and fold recognition tool which can conveniently takes external sequence profiles as input. We used 99 of CASP 9 sequences and the database before CASP 9 as the benchmark dataset.

3.6.1 Performance of PSI-BLAST Using NR or Pfam Sequence Database

Figure. 3.10 and Figure. 3.11 compared the Top-1 alignment quality and fold recognition of PSI-BLAST between using NR database and Pfam sequence database, which contains the sequence of all the domains of PfamA database. From these two figures, we can see that the replacing NR database with Pfam sequence improves the fold recognition, especially for those "hard cases", for which the default PSI-BLAST fails to find the right template. But in terms of alignment accuracy, the alignments by the default procedure using NR database is significantly better than that of using Pfam sequence database.

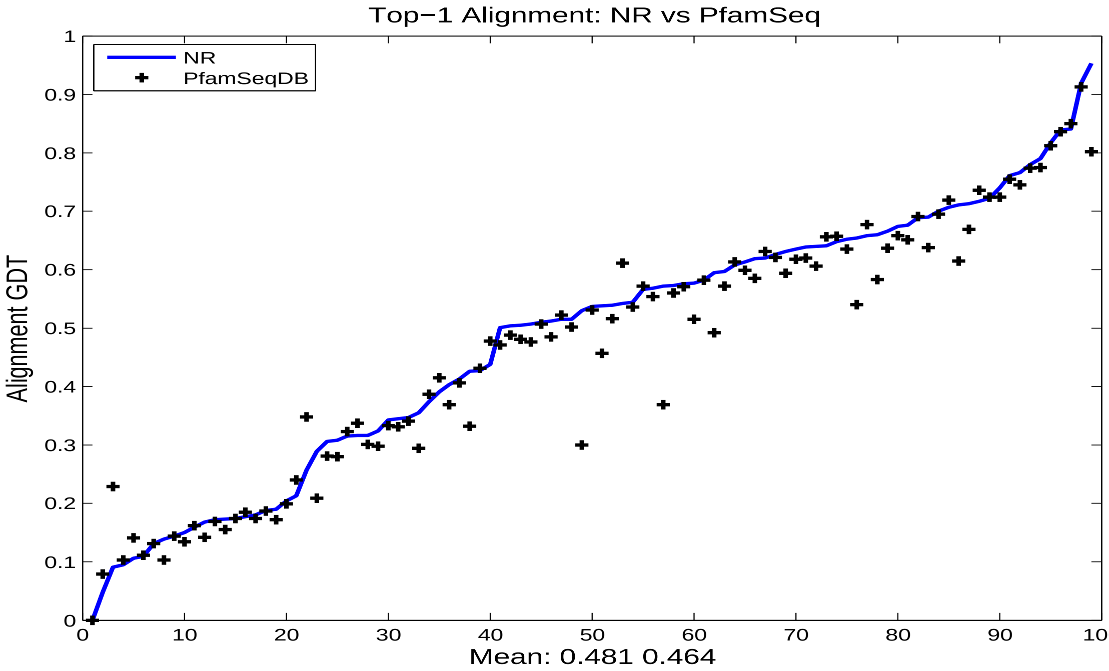

Figure 3.10: Top-1 Alignment Accuracy of PSI-BLAST: NR vs Pfam Sequence Database. Blue curve is the top-1 alignment accuracy of PSI-BLAST using NR database, and black dots are for that of using PfamA sequence database.

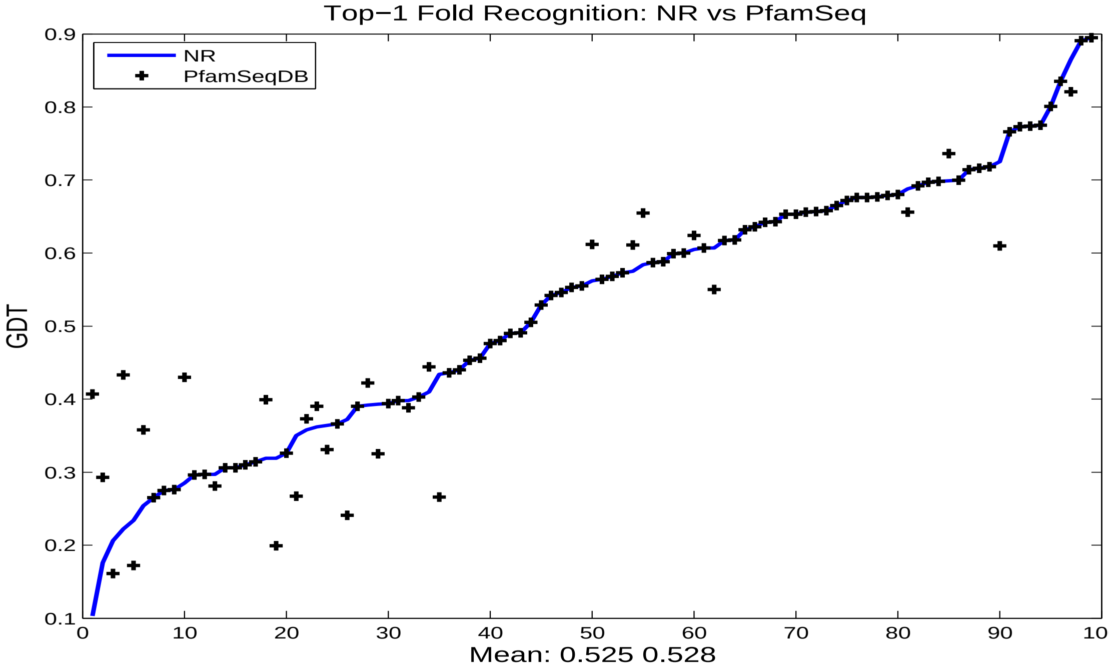

Figure 3.11: Top-1 Fold Recognition of PSI-BLAST: NR vs Pfam Sequence Database. Blue curve shows the fold recognition performance of PSI-BLAST using NR database and black dots are that of using PfamA sequence database.

3.6.2 Performance of DELTA-BLAST

Figure. 3.12 and 3.13 compared the top-1 alignment accuracy and fold recognition performance of PSI-BLAST with NR database and DELTA-BLAST with CDD database without iterative PSSM refinement and with iterative PSSM refinement. As we can see that DELTA-BLAST did not improves the top-1 alignment quality, when compared to the original PSI-BLAST. In terms of fold recognition, the overall performance was the same as that of PSI-BLAST. But from the Figure 3.13, we can see that for those cases on which PSI-BLAST fails, DELTA-BLAST improves observably.

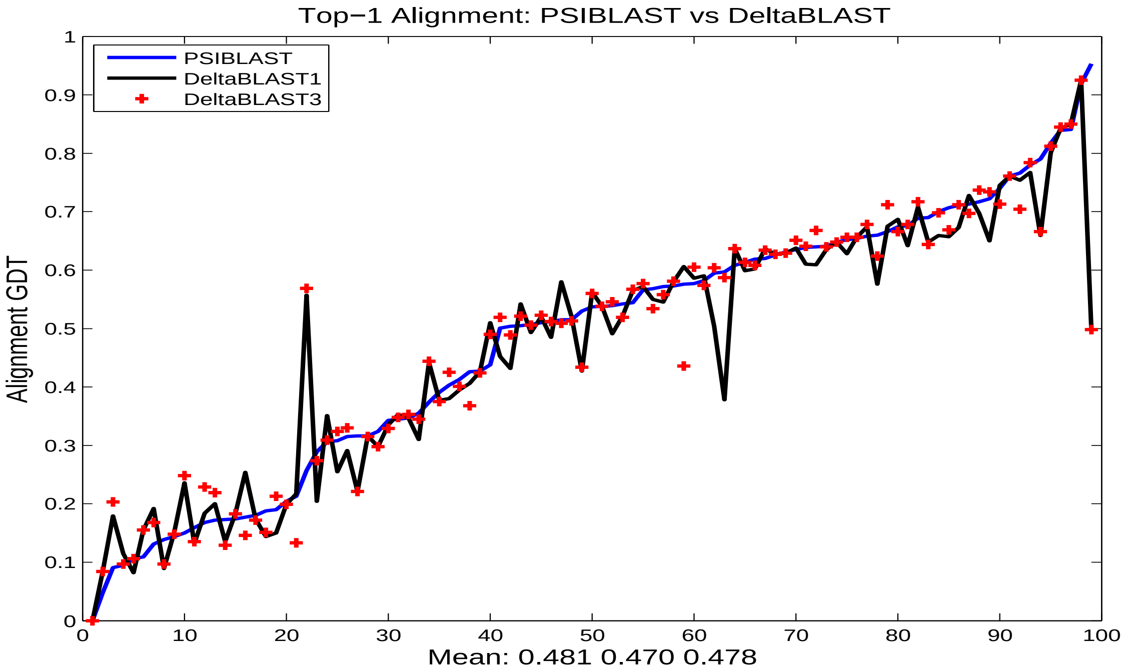

Figure 3.12: Top-1 Alignment Accuracy: PSI-BLAST vs DELTA-BLAST. Blue line shows the alignment accuracy of PSI-BLAST with NR database, black curve is the corresponding performance of DELTA-BLAST without iterative PSSM refinement and red points are that of DELTA-BLAST with iterative PSSM refinement.

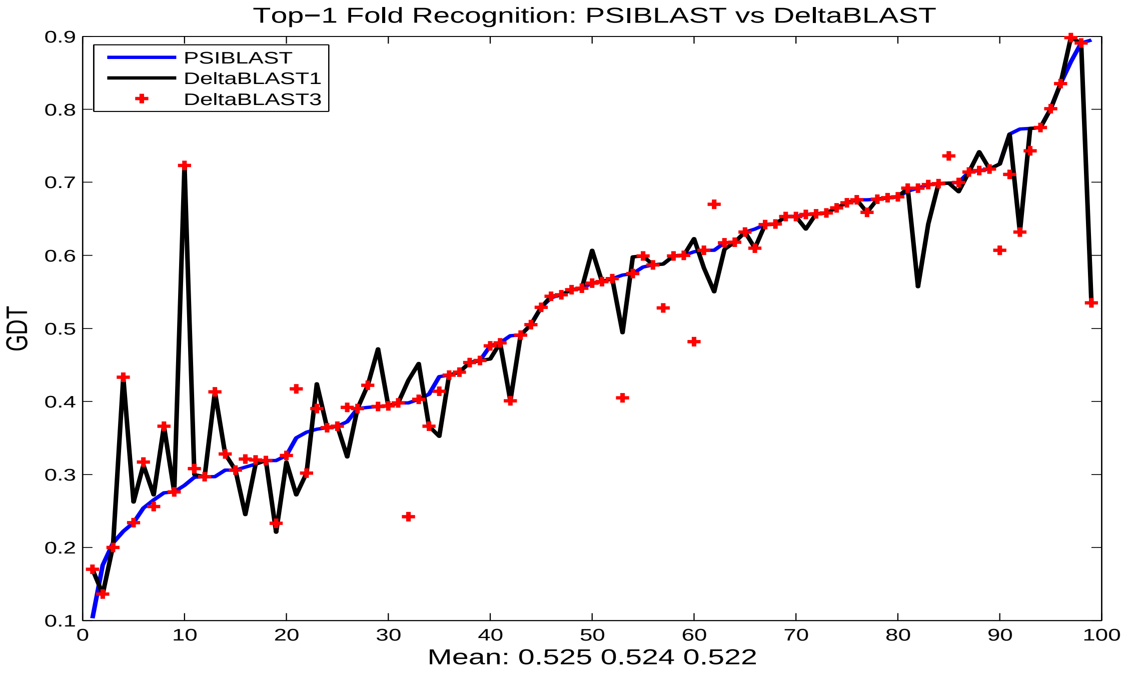

Figure 3.13: Top-1 Fold Recognition: PSI-BLAST vs DELTA-BLAST. Blue line shows the top-1 fold recognition performance of PSI-BLAST with NR database, black curve is the corresponding performance of DELTA-BLAST without iterative PSSM refinement and red points are that of DELTA-BLAST with iterative PSSM refinement.

3.6.3 Performance of the New Protocol

Figure. 3.14 and Figure. 3.15 compared the top-1 alignment accuracy and fold recognition performance of HHSearch default method and the protocol shown in Figure 3.9. As we can see that the protocol improves the top-1 fold recognition over the original HHSearch, for quite a number of cases, the improvement is significant, the average of which is $0.559 - 0.547 = 0.12$ GDT point. In terms of top-1 alignment quality, the performance is slightly worse than that of original HHSearch. But for those cases in the leftmost region of Figure 3.15 , HHSearch fails to find the right fold as the top-1 template, this method achieves quite good improvements.

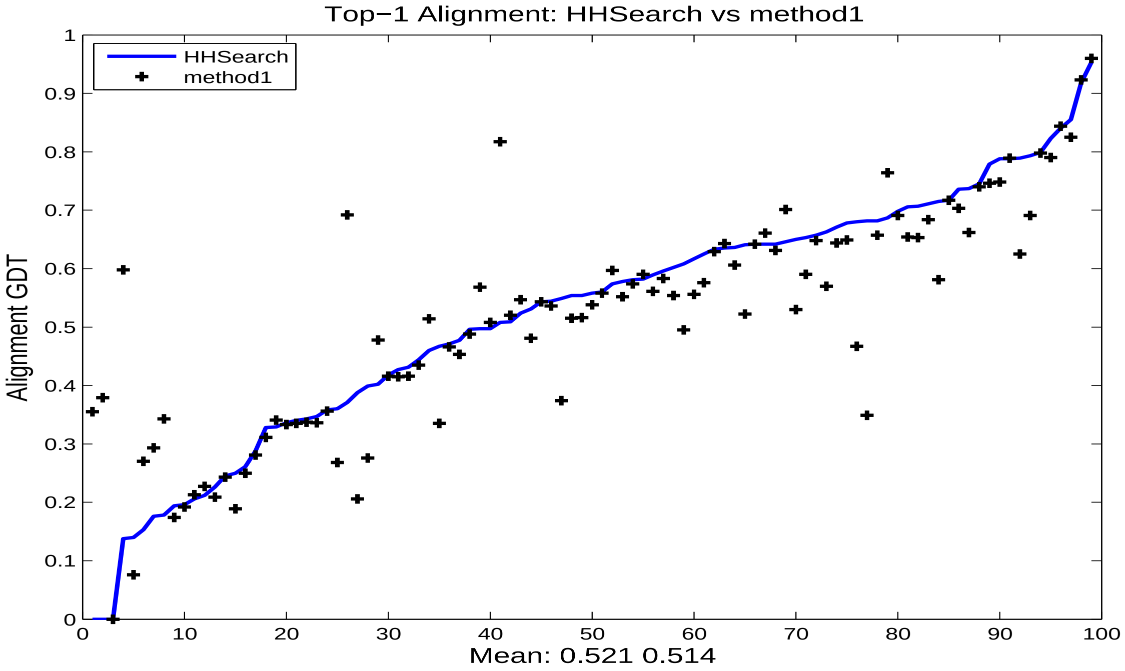

Figure 3.14: Top-1 Alignment Accuracy: Protocol1 vs HHSearch Default. Blue line shows the base line performance of HHSearch default procedure in terms of top-1 alignment accuracy. Black dots are the corresponding performance of the new protocol.

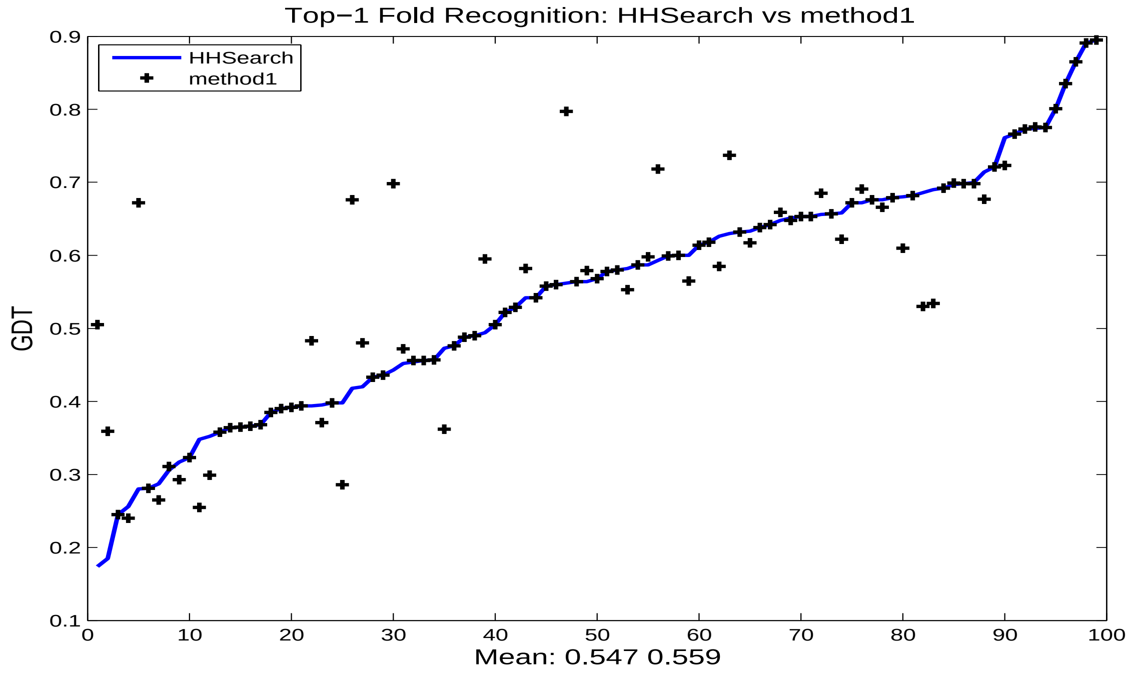

Figure 3.15: Top-1 Fold Recognition: Protocol1 vs HHSearch Default. Blue line shows the base line performance of HHSearch default procedure in terms of top-1 fold recognition performance. Black dots are the corresponding performance of the new protocol.

Chapter 4

Introduction To Protein Model Quality Assessment

Currently, almost all protein structure prediction methods, comparative protein modeling and ab initio modeling, adopt a sampling-selection strategy. In this strategy, the first step is to generate a large number of candidate models with a sampling procedure; and then apply structural quality assessment methods to identify the most native-like conformations. This is reasonable as it always increases the chance of producing candidate models with better quality. For this protocol to work, it is required that the sampling procedure is capable of producing at least some near-native conformations and the scoring method is able to identify more native-like structures from the structural model pool. Accordingly, this makes protein structure quality assessment (QA) play an important role in protein structure prediction.

4.1 Quality Measures for Protein Structures

A number of measures have been proposed by publications to measure the similarity or distance between two protein structures. In order to calculate the structural distance or similarity between two structures, structural alignment is required when we consider the comparison is sequence independent, in which the positional correspondence between the comparing structures is unknown. On the other hand, when the comparison is called sequence dependent, we only need to optimize the structural translation and rotation between the structures to calculate the score. In QA scenario, we are assuming the structure comparison is sequence dependent in the following scores.

Let $d_{1,2}(i,j)$ be the distance between the ith residue of structure 1 and the jth residue of structure 2, and L is the protein length. The following scores are typical and often used in protein structure prediction.

- **Root Mean Square Deviation** (RMSD) between two structures is defined as

$$RMSD = \sqrt{\frac{\sum_{i=1}^{L} d_{1,2}^2(i,i)}{L}} \tag{4.1}$$

When the difference between two structures is not too big, RMSD is a good measure.

- **Global Distance Test Total Score** (GDT) was proposed by Zemla in [20]. The score is defined as

$$GDT = \frac{n_1 + n_2 + n_4 + n_8}{4 * L} \tag{4.2}$$

Where $n_i, i = 1, 2, 4, 8$ is the number of positions where the distance between the two residues is less than $i\mathring{A}$. To calculate this, it is required to search for the optimal superimposition four times, independently. Unlike RMSD, GDT score is a similarity score between two structures. If we define

$$GDT_i = \frac{n_i}{L} \tag{4.3}$$

where i can be any number which is usually an integer within (0 8]. Thus, the GDT score defined above can be rewritten as

$$GDT = \frac{GDT_1 + GDT_2 + GDT_4 + GDT_8}{4} \tag{4.4}$$

- **TMScore** was proposed by the work of [50] to extend GDT score so that the score is less dependent on the protein size for randomly selected structure pairs.

$$TMScore = max\left[\frac{1}{L}\sum_{i=1}^{L}\frac{1}{1+(\frac{d_i}{d_0})^2}\right] \tag{4.5}$$

where d_i is the distance between the ith pair residues after the optimal superimposition and d_0 is a scale to normalize the match difference. One of the advantages of TMScore over GDT is that only one time searching for optimal spatial superimposition is needed.

4.2 Introduction to Protein Structure Quality Assessment

4.2.1 Types of Protein Structure Quality Assessment

Protein structure Quality Assessment (QA) can have different types from different perspectives. Global quality assessment is to assign a score to each decoy to indicate its overall structural quality, while local quality assessment assigns a score to each position in the structure. QA score can can be interpreted either as a geometrical distance or similarity, which measures the actual distance to the corresponding native structure, or a confidence score which tells how likely the predicted decoy is close to the native. For example, the score given by OPUS-CA [17] can only be thought of as confidence, meaning the more negative the score is, the better quality the structure might have. QA methods can also be divided into two categories depending on the input structures. If the method relies on the decoy itself only, we call it a single model QA method, otherwise it is a consensus method, which takes a set of peer structure decoys (predicted decoys from the same sequence) as the input. For example, OPUS-CA is a method that can evaluate each individual protein structure while the consensus methods always need a set of structures.

4.2.2 Measures for Quality Assessment Methods

A lot of work has been done for protein structure assessment. Selection and correlation performance are typically used to compare different QA methods. For a given set of protein decoys of the same sequence, QA method is used to assign a score to each decoy. In the meanwhile, each structure candidate has its GDT score to the native structure. For selection performance we consider the structural quality of the selected top-1 structure and the average quality of the selected top-5 structures, in terms of the GDT score to the native structure. For correlation performance, we only consider Pearson and Spearman correlation between the assigned score and actual GDT score.

4.3 Existing Methods for Protein Model Quality Assessment

As mentioned above, QA includes global quality assessment and local quality assessments. In this selection, we mainly review the current methods in global quality assessment as the methods in this thesis work mainly focus on this category.

4.3.1 Global Quality Assessment Methods

A lot of work have been done in this area since Critical Assessment of Protein Structure Prediction (CASP) [19] which takes place every two years.

Single Model QA Methods:

Single model QA methods evaluate the structural quality of the single protein structure from the input. These methods fall into different categories according to the underlying scoring function.

Physical-based energy functions [51, 52, 53, 54, 55, 56, 57, 58, 59] compute the energy of a protein structure based on physics principles at the atomic level. Physical energies are often too sensitive to small atomic changes, and hence they are not widely used in practical model selection.

Knowledge-based scoring functions, such as OPUS-CA [17], DFIRE [18], DDFIRE [60], RW [61] and QMEAN [62, 63] score the models based on the statistical information of structural attributes in known native structures. These scoring functions are widely used in protein structure prediction. However, knowledge-based scoring functions can only reflect some aspects of protein structures. For example, OPUS-CA uses the distance-dependent energies from the C-alpha atoms of a model, while RW is a side-chain orientation dependent potential. While some success is achieved, overall they have limited discerning power for ranking structural models.

Several machine learning based methods have been proposed to evaluate single structure quality, such as [64, 62] and MULTICOM series for single QA [65, 66, 67]. For example, in [65], a method was proposed to assign GDT score to a structure model by comparing its actual secondary structure, relative solvent accessibility, contact map, and beta sheet structure with their counterparts predicted from its primary sequence.

QA Methods Based on Structure Set:
QA methods of this category take advantages of the mutual distance or similarity between decoys from a common set and assume that the a decoy which is more structurally similar to other candidates in the decoy set has better quality with respect to the native structure. For example, naive consensus GDT is defined as follows. Given a set of predicted decoys $d_1, .., d_N$ for a certain protein, the consensus GDT

score for each decoy

$$CGDT_i = \frac{\sum_{j=1}^{N} GDT(i,j)}{N} \tag{4.6}$$

where $GDT(i,j)$ is the GDT score between decoy i and decoy j calculated by structure comparison and N is the number of decoys in the set. Naive consensus GDT method takes all the decoys from the set as the reference set to calculate CGDT score. Several methods tried to use different schemes to create better reference set to improve the performance of CGDT score. For example [68] adaptively selected reference decoys based on the attributes of the whole decoy set and excludes those decoys that are too similar or too different from the reference set. This approach has been the most successful for model quality assessment in CASP, where the model pool contains the top predictions submitted by the attending groups. However, when it is difficulty to find out the structurally similar templates from PDB [1] database for the query protein, the resultant decoy set is no more dominated by a set of decoys, let alone dominated by good decoys. In these situations, naive consensus methods often do not work well as it only considers the structural or geometrical information. To address the problem, a lot of work have been done to combine the structural consensus information with sequence or structural features such as single QA scores [69, 70, 62, 71, 72] using machine learning methods and clustering methods such as MULTICOM-REFINE [66, 67] and ModFOLD [73, 74].

4.3.2 Local Quality Assessment

Local quality assessment is also very important, especially for the structure refinement. Given a set of structure candidates, local quality of each structure can be predicted based on consensus structure. For example, MULTICOM-REFINE [66, 67] and IntFold-TS [75] calculates the local quality score as the average absolute difference between each residue in the decoy and the residue in the decoy from a selected reference set. For a single structure, its local quality can also be predicted using the sequence-structure relationship, for example, the match scores of secondary structures and solvent accessibility comparison [76].

Chapter 5

Protein Structural Model Selection Based on Protein-dependent Scoring Function

5.1 Introduction

Knowledge based scoring functions have performance inconsistencies for different proteins. We believe that combining several scoring function can result in better performance as they complement each other to some extent. Although consensus methods do not work well for a decoy set that does not contain predominantly good models, one may overcome this by selecting a subset of good models using scoring functions.

Based on these considerations, in this study, we proposed a two-stage optimization approach to take advantages of scoring functions, consensus method and machine learning. In the first protein-dependent optimization, different "noisy" scoring functions were combined to improve the sensitivity of scores for model selection. In this step, each target protein has a pool of structural models without knowing the native structure. For each protein, a subset of models was selected using basic scoring functions to remove likely poor models. Then weights for these scoring functions were optimized on the selected model set of each protein. Ideally, we should use the real GDT-TS score [20] (one of the most widely used scores for protein quality) of models to optimize the weights. Due to lack of native structures, we replaced the real GDT-TS score with a consensus GDT-TS score, which is an estimate of GDT-TS using a consensus approach. The sum of these scores with the optimized weights can be directly used to rank models. However, it was still "noisy" due to the errors introduced by scoring functions and the consensus method. In the second stage optimization, we integrated the weighted scoring functions, correlations of these scores to consensus GDT-TS, model quality computed by consensus method and structural features to train an Support Vector Machine (SVM) that maps these features to the real GDT-TS scores based on separate protein targets with structural model pools and known native structures. Through the two sequential optimizations, the resulting score can gain sufficient discerning power to outperform basic scoring functions and consensus method for model selection.

We have applied this new method to two benchmarks and demonstrated that the weighted sum of individual scoring functions improved the top-1 and top-5 model selection performance, and a following SVM gained further improvement.

5.2 Methods

An overview of our method is presented in Figure 5.1. The first step was to compute the basic scores for each model using scoring functions. Then for each protein, the best weights for scoring functions were obtained through the protein-specific optimization on the subset (at most top 300) models selected by the average rank based on basic scores. The resultant weighted sum (S1 in Figure 5.1) can be directly used to rank the models. The basic scores and weights were integrated into the second stage optimization using an SVM which was trained on models from different proteins with the real GDT-TS score of each model as the target value.

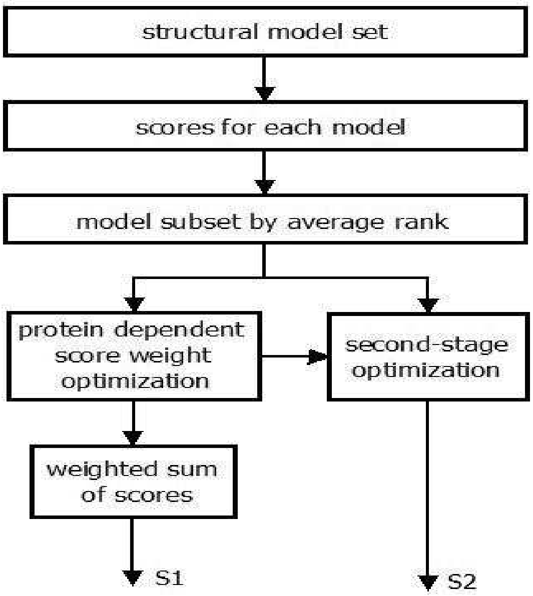
structural model set
scores for each model
model subset by average rank
protein dependent score weight optimization
second-stage optimization
weighted sum of scores
S1
S2

Figure 5.1: Method Flowchart

5.2.1 Scoring Functions

In this method, five published protein structure quality assessment (QA) scores were selected, namely OPUS-CA [17], OPUS-PSP [77], DFIRE [18], DDFIRE [60] and RW [61]. These scores evaluate structure models from different perspectives. Also we computed two additional statistical based scores, i.e., environment fitness score and secondary structure similarity score, which are widely used in threading-based protein structure predictions [26, 25].

Environment Fitness Score

This score measures the propensity of an amino acid type a to appear in a structural environment env_j on the model. The environment type is specified by the secondary structure type (H: helix, E: beta sheet, or C: coil) and solvent accessibility type (B: buried, I: intermediate or E: exposed). The environment fitness score is given by

$$envfitness = \sum_{j=1}^{N} \sum_{a=1}^{20} prob(env_j, a) \times prob(j, a) \tag{5.1}$$

where N is the protein sequence length. $prob(env_j, a)$ is the probability of amino acid type a to appear in structural environment env_j obtained through statistical analysis on a set of training native structures [26, 25]. It is worth mentioning that these structures had no overlap with the ones used in the following benchmark tests. $prob(j, a)$ is the probability of amino acid type a occurring at position j of the protein, which can be calculated from the sequence profile generated by PSI-BLAST [12].

Secondary Structure Similarity Score

For each model, we computed its actual secondary structure based on its 3D coordinates using DSSP [39] . We also used PSIPRED [37, 38] to predict the secondary structure from its amino acid sequence. The similarity between these two secondary structures is a good indication of model quality. Higher secondary structure similarity usually means better model quality. Suppose the secondary structure type at position j of model is S_d, and the corresponding predicted secondary structure from sequence by PSIPRED is S_p with confidence value P, the score is defined as

$$sssimilarity = \sum_{j=1}^{N} prob_j(S_d, S_p, P) \tag{5.2}$$

where $S_d, S_p \in \{H, E, C\}, P \in [0, 9]$ and $prob(S_d, S_p, P)$ is the probability of S_d being predicted as S_p with confidence value P, obtained from a training dataset whose proteins had no overlap with the ones used in the following benchmark tests.

5.2.2 Protein-dependent Weights Optimization

Let $s_1, s_2, .., s_7$ be the seven scores of a model, and $w_1, w_2, .., w_7$ be the weights for the scores. We optimized the weights by minimizing

$$L_2 = \sum_{i_1, i_2} \left[\sum_{j=1}^{7} w_j (s_j^{i_1} - s_j^{i_2}) - [GDT(i_1) - GDT(i_2)] \right]^2$$

where $w_j < 0$ and i_1, i_2 are two structural models of the same protein. $s_j^{i_1}$ is score j of structure model i_1 and $GDT(i_1)$ is the GDT-TS score of model i_1.

In practice, GDT-TS score is not available as we do not have the native structure. So we used consensus GDT-TS score, $cgdt()$, to approximate the real $GDT()$ score.

A reference set R containing the top 300 models was selected according to the average rank using the seven basic scores. cgdt of a model is defined as the average GDT-TS score to the remaining models in R. Thus the weights were optimized on R by minimizing

$$L_2 = \sum_{i_1, i_2 \in R} \left[\sum_{j=1}^{7} w_j (s_j^{i_1} - s_j^{i_2}) - [cgdt(i_1) - cgdt(i_2)] \right]^2$$

Let $x_j^k = s_j^{i_1} - s_j^{i_2}$ and $y_k = cgdt(i_1) - cdgt(i_2)$, we have

$$L_2 = \sum_k \left[\sum_{j=1}^{7} w_j x_j^k - y_k \right]^2, w_j < 0 \tag{5.3}$$

Further, let $W = [w_1, .., w_7]^T$ and $X_k = [x_1^k, .., x_7^k]^T$, Eqn. (5.3) becomes

$$\begin{aligned} L_2 &= W^T \sum_k X_k X_k^T W - 2W^T \sum_k y_k X_k \qquad (5.4) \\ &+ \sum_k y_k^2, \ W < 0 \end{aligned}$$

Minimization of Eqn. (5.4) was solved by quadratic programming. Before optimization, all the scores were normalized to Z-score. Z-score of score S is defined as $Z = \frac{S - avg(S)}{dev(S)}$, where $avg(S)$ is the mean value and $dev(S)$ is the standard deviation in the structural model pool. Each scoring function has its "direction"; for example, OPUS-CA is "negative" compared to GDT-TS, which means lower OPUS-CA values usually have higher GDT-TS scores. In the actual optimization, the "directions" of seven scores were all adjusted to be "negative." Also, due to the noise in the training data, weights were constrained to be less than -0.0001 to keep the optimization from reversing or disabling any scores. After optimization, weights were obtained for each score and the score S_1 in Figure 5.1 was $S_1 = \sum_{j=1}^{7} w_j s_j$.

5.2.3 Second Stage Optimization

This optimization was implemented as an SVM. The input features for each model included:

- Weighted scores $w_j s_j, j = 1, .., 7$.

- Spearman correlation of each score $s_j, j = 1, .., 7$ to consensus GDT-TS score. The correlations of different scores indicate their relative performance on models of a specific protein.

- Naive consensus GDT-TS score *cgdt*.

- Another secondary structure score to strengthen the similarity between the actual secondary structure in model and the predicted one from sequence. It is defined as $SSIden = \frac{\sum_{j=1}^{N} \delta(SS_p, SS_d)}{N}$, where N is protein sequence length and

$$\delta(SS_p, SS_d) = \begin{cases} 1 & SS_p = SS_d \\ 0 & SS_p \neq SS_d \end{cases}$$

- Solvent accessibility (SA) matching scores, which is similar to $SSIden$. $SAIden = \frac{\sum_{j=1}^{N} \delta(SA_p, SA_d)}{N}$, where N is protein sequence length and

$$\delta(SA_p, SA_d) = \begin{cases} 1 & SA_p = SA_d \\ 0 & SA_p \neq SA_d \end{cases}$$

SA_p is the predicted solvent accessibility by SSPro [40] and SA_d is computed from models by DSSP with cutoff 25% (above which means the exposed state and otherwise the buried state).

Although the secondary structure and solvent accessibility information were used in the seven scores, SSIden and SAIden were more direct to help SVM to learn the "weak" relationship between features and real GDT-TS score. The SVM was trained using SVMLight with a linear kernel.

5.2.4 Dataset

We applied the method to two benchmarks produced by different model generation methods. Benchmark1 was from Yang Zhang's lab (http://zhanglab.ccmb.med.umich.edu/decoys/), generated by the I-TASSER ab initio modeling tool, containing 56 proteins. The other one, benchmark2, included models generated by Robetta or Rosetta, containing 34 CASP8 proteins. Each protein in both benchmarks had hundreds of decoys. Figure 5.2 shows the maximum, average and minimum GDT-TS score of models of each protein for both benchmarks. The best model of each protein had a GDT-TS score greater than 0.4, which ensured that the pool contained some reasonably good models.

5.3 Results

In the test, each score was used to rank the models of a given protein. We used four metrics to compare the performance of each scoring method. In the following tables, ”GDT1” is the average GDT-TS score of top 1 model; ”avgGDT5” is the average of

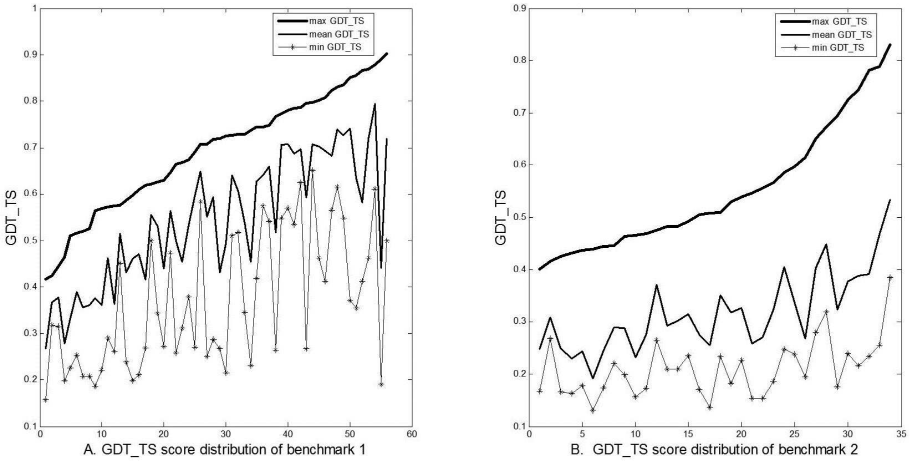

Figure 5.2: Model quality measured by GDT-TS score to the native structure. The X-axis is the proteins of each benchmark sorted by the GDT-TS score of the best model. (A) Model quality distribution of benchmark1. (B) Model quality distribution of benchmark2.

the mean GDT-TS score of top 5 models. ”Pearson” indicates the Pearson correlation to real GDT-TS and ”Spearman” is the Spearman correlation to real GDT-TS score. Table 5.1 and 5.2 compares seven basic scores mentioned above, avezscore, averank and S1 on benchmark1 and benchmark2, where ”avezscore” is the sum of the seven scores after normalization; ”averank” is the average rank using seven basic scores. ”S1” is the weighted sum of basic scores. The term averank was used to select the top 300 models for each protein to optimize the weights for S1. Table 5.3 shows the selection performance of cgdt and S2 on the subset models selected by averank.

As shown in Table 5.1 and 5.2, weighted sum with the optimized weights improved over seven basic scores, in top-1 and top-5 selection performance. For example, for benchmark2, the best scoring function was DDFIRE, which had GDT1 performance of 0.3976 and avgGDT5 of 0.3833, while S1 achieved GDT1 of 0.4012 and avgGDT5 of 0.3977. Furthermore, weight optimization improved over avezscore and averank

	Benchmark1			
	GDT1	avgGDT5	Pearson	Spearman
GDT-TS	0.6918	0.6737	1.0000	1.0000
OPUS-CA	0.5935	0.5904	0.4952	0.4159
OPUS-PSP	0.5670	0.5715	0.2893	0.2906
DFIRE	0.5984	0.5882	0.5332	0.4416
DDFIRE	0.5984	0.5883	0.5328	0.4411
RW	0.5927	0.5855	0.4909	0.4178
envfitness	0.5604	0.5691	0.3805	0.2985
sssimilarity	0.5836	0.5823	0.3578	0.2938
avezscore	0.5966	0.5919	0.5486	0.4530
averank	0.5970	0.5895	0.5126	0.4562
S1	**0.5989**	**0.5953**	**0.5824**	**0.4841**

Table 5.1: Comparison of scores based on their performance.

	Benchmark2			
	GDT1	avgGDT5	Pearson	Spearman
GDT-TS	0.5504	0.5281	1.0000	1.0000
OPUS-CA	0.3769	0.3705	0.2980	0.2709
OPUS-PSP	0.3171	0.3253	0.0993	0.0941
DFIRE	0.3389	0.3277	0.0723	0.0786
DDFIRE	0.3976	0.3833	0.3050	0.2718
RW	0.3707	0.3738	0.2987	0.2727
envfitness	0.3501	0.3396	0.1050	0.0962
sssimilarity	0.3571	0.3623	0.2366	0.2152
avezscore	0.3856	0.3823	0.3291	0.2987
averank	0.3861	0.3707	0.3200	0.2969
S1	**0.4012**	**0.3977**	**0.3709**	**0.3489**

Table 5.2: Comparison of scores based on their performance.

in selection performance, especially for benchmark2, as our optimization was carried out on the subset selected by averank. For Pearson and Spearman, we can see from Table 1 that S1 had the best correlation to the real GDT-TS score among the scores being compared on both benchmarks. For example, for benchmark1, although the selection improvement of S1 over the best of other scores was small, the improvement in correlation was quite significant. In Figure 5.3 we took the protein 1SHF from benchmark1 as an example to show the score distribution. It is evident that S1 had a much better correlation to real GDT-TS than sssimilarity and the top model selected by S1 was better than the one by sssimilarity.

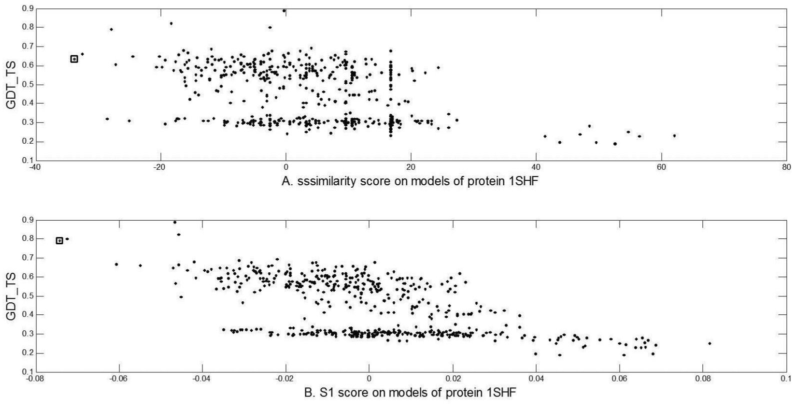

Figure 5.3: Score distributions for models of protein 1SHF from benchmark1. (A) Score distribution of sssimilarity with respect to GDT-TS. (B) Score distribution of S1 with respect to GDT-TS. The point highlighted in the box is the top model selected by the score.

Table 5.3 shows that after selecting the top 300 models for each protein using averank, the GDT-TS loss between the best model in the 300-model set and the best model in the entire pool was acceptable for benchmark1; the average GDT-TS loss was only $0.6918 - 0.6892 = 0.0026$. For benchmark2, the best models of all proteins were kept in the selected top-300 model set, i.e., with 0 GDT-TS loss. Table 5.3

	Benchmark1		Benchmark2	
	GDT1	avgGDT5	GDT1	avgGDT5
GDT-TS	0.6892	0.6713	0.5504	0.5273
cgdt	0.6047	0.6030	0.4351	0.4217
S2	**0.6098**	**0.6034**	**0.4446**	**0.4220**

Table 5.3: Comparison of scores based on reference set. "S2" corresponds to the SVM output in Figure 5.1.

also shows the leave-one-out performance of the SVM. This research trained different models for benchmarks 1 and 2 as they were generated by different methods and had quite different structural characteristics and distributions which were reflected by the diverse performances of basic scores. In leave-one-out training and testing, all proteins were tested using one model while the remaining were used as training data. Table 5.3 shows that S2 improved over cgdt on both benchmarks, especially in GDT1 performance. For benchmark1, GDT1 of S2 was 0.6098, which gained about half a GDT-TS point ($0.6098 - 0.6047 = 0.0051$) over cgdt (0.6047). For benchmark2, the improvement over cgdt in GDT1 was $0.4446 - 0.4351 = 0.0095 \cong 0.01$. On the other hand, S2 had significantly better GDT1 and avgGDT5 performance than basic scores. Especially, for benchmark2, the best basic scoring function was DDFire, whose GDT1 was 0.3976, while S2 had GDT1 of 0.4446. The improvement was $0.4446 - 0.3976 = 0.047$.

5.4 Discussion

Our new approach combined the advantages of various methods and avoided some of their limitations. Existing scoring functions such as OPUS-CA and DFIRE do not work consistently well for model selection of different proteins especially when models are generated by different methods. Consensus method depends only on the dataset itself and does not use any information from native structures. In order to improve the selection performance, for each protein, we trained the weights for each score on a reference set which was selected to enrich the overall quality of the smaller pool. The resultant weighted score was less noisy and more correlated with the real GDT-TS score. With the weighted scores, it is more advantageous for the second stage optimization to learn the weak intrinsic correlation between input features and real model quality.

However, several factors may affect the performance of our method. One such factor is model distribution. S1 and S2 had more GDT-TS loss between the selected top-1 model and the best model in the pool in benchmark2 than in benchmark1. Specifically, GDT-TS loss of S2 in benchmark1 was $0.6918 - 0.6098 = 0.082$; while for benchmark2, the GDT-TS loss was $0.5504 - 0.4446 = 0.1058$. Comparing the two distributions of model pools in Figure 5.2, it is evident that the gap between max and mean GDT-TS in benchmark2 was much bigger than that in benchmark1. The distribution difference also affects the performance of other scores in the same way. For benchmark2, GDT1 of the real GDT-TS score was 0.5504, while all basic scores were less than 0.4, losing more than 0.15, significantly bigger than that in benchmark1.

For the second stage optimization, selection of features and learning method directly affects the performance of S2. Although S2 is not significantly better than cgdt on either of the two benchmarks, the S2 method has some merit. In particular, cgdt and basic scoring functions have different properties and combining them theoretically may improve the performance. Furthermore, the performance of cgdt depends on the distribution of the model pool or how the model pool is generated. The model pool generated in CASP or by the tools that guarantee good sampling of structural conformation can lead to good performance of cgdt; otherwise the performance of cgdt may not be good. In addition, this research concluded that the S2 method has significant room for improvement. We are exploring a better way to do the second stage optimization and combine the two stages. For example, one may use the priori general information of model quality vs. a given scoring function and use that information to guide optimization. The SVM here was developed to demonstrate that integrating weighted scores, their statistical features and structure-related features into optimization over different proteins can improve the performance over any individual feature. On the other hand, more advanced machine learning techniques, such as random forests may further enhance the performance.

There are some limitations of our method. Given that it is based on training from a model pool, it may not be applicable to simultaneously assess models from different generation methods as they may have different characteristics or distributions. For example, our method may not be applicable to the model pool generated by different servers in CASP. Our method is mainly designed for model selection with a single tool which is most practical in protein structure prediction applications.

Chapter 6

Protein Structural Model Selection by Combining Consensus and Single Scoring Methods

6.1 Introduction

Consensus and knowledge-based scoring functions reveal different but complementary aspects of structural models. Consensus method utilizes geometric information from the decoy set only, without taking advantages of the biophysical properties within and between primary sequences and 3D structures. In [72], we developed a protein-dependent scoring method to combine consensus and single scoring functions for decoy selection. In this method, the optimal weights for each component scores were obtained in the first optimization step. Then the weighted scores and other sequence-structure features were combined by an support vector machine (SVM) in the second step. This method achieved improvement over naive consensus GDT (CGDT) score and single scoring functions in selection performance.

However, it still had room for further improvement for two reasons. On one hand, it mapped feature scores to the actual structure quality in terms of native structure, which is relatively difficulty to machine learning methods such as SVM or neural network to capture. On the other hand, it is computationally intensive to optimize the weights in the first step. Here we proposed a new method to combine consensus GDT and knowledge-based scoring functions to obtain better discerning power for QA. First, a consensus method called Position Specific Probability Sum (PSPS) was developed as one of the feature scores. Here, the structural state of each residue was represented by the bond angles of four consecutive residues in a decoy. Thus, each decoy was represented by a sequence of structure code. A probability score was calculated for each decoy of a set based on consensus. Although this method alone did not have outstanding performance in decoy selection, it was quite different from all other methods, and outperformed CGDT when combined with other methods such as RW [61], OPUS-CA [17] and DDFIRE [60]. Second, a two-stage method was developed to perform QA. We trained two neural-network models to capture the underlying correlation among different features (scoring functions). Specifically, for every two decoys, the first neural-network model decided whether they were structurally close or not, and subsequently the second model determined which one was better than the other in term of GDT score to the native. After the comparison between all pairs of decoys, we calculated a score for each decoy in the pool based on the number of winning times.

We applied this method to three benchmark data sets from different protein structure prediction methods and demonstrated significant improvements over CGDT and state-of-art single scoring functions in terms of best model selection performance and Spearman correlation to actual GDT score.

6.2 Method

6.2.1 Position Specific Probability Sum (PSPS) Score

Each decoy in a decoy set was transformed to a sequence of structural alphabet based on the study in [78]. For each residue x_k, we calculated angle triplet $(\theta_k, \tau_k, \theta_{k+1})$ of four consecutive C-alpha atoms, where θ_k is the bend angle of (x_{k-2}, x_{k-1}, x_k), τ_k is the dihedral angle of $(x_{k-2}, x_{k-1}, x_k, x_k k+1)$ and $\theta_{k+1}$is the bend angle of (x_{k-1}, x_k, x_{k+1}). x_k was assigned to one of the 17 clusters (states) according to the following Gaussian

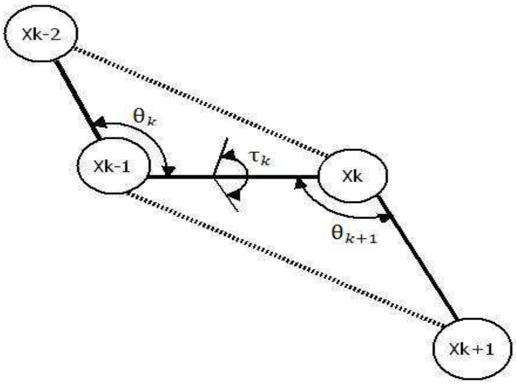

Figure 6.1: Angles of four residues

Mixture Model:

$$\begin{aligned} P(C_i|x_t) &\propto \pi_i P(x_t|C_i) \\ &\propto \pi_i |\Sigma_i|^{-\frac{1}{2}} \cdot e^{-\frac{1}{2}(x_t-u_i)'\Sigma_i^{-1}(x_t-u_i)} \end{aligned} \quad (6.1)$$

and,

$$i = \arg\max_{1\leq i\leq 17} P(C_i|x_t) \tag{6.2}$$

For more details about this, please refer to Appendix A.

After we had all the 4-mer sequences of the decoys for the same protein, we calculated Position Specific Frequency Matrix (PSFM). $PSFM(i, j)$ is the occurring frequency of state i at sequence position j, where i is the state index from 1 to 17, representing the 17 clusters; j is the residue position from 3 to $L - 1$ (L is the length of protein). This matrix was counted in the model pool and normalized by dividing the number of decoys. We then got Position Specific Probability Sum (PSPS):

$$PSPS(k) = \sum_{j=3}^{L-1} PSFM(SA_j^k, j) \tag{6.3}$$

where k is the decoy index and SA_j^k is the cluster state of position j in the structure code (state) sequence of decoy k.

6.2.2 Combine Consensus and Single Scoring Functions

For a set of decoys of a target protein, the input features for every decoy-pair were the respective differences between OPUS-CA, RW, DDFIRE, PSPS and CGDT of the two decoys.

$$S_{(k)}^{ij} = S_{(k)}^{i} - S_{(k)}^{j} \tag{6.4}$$

where i, j are decoy indexes, and k represents different scores.

Two neural-network models were used to compare a decoy pair. Model 1 was trained to determine whether two decoys were significantly different in terms of the GDT score to their native. We chose the cutoff to be 0.025, which meant if the GDT difference of two decoys was larger than 0.025, these two decoys were treated as significantly different. Model 2 was used to predict whether one decoy was better than the other. To train this model, considering the training error, we removed those of pairs whose GDT difference is less than 0.01 from training data. Model 2 was tested only on the pairs that were predicted to be significantly different by Model 1. After the comparison between all pairs of decoys, the final score, named as PWCom, for each decoy was simply the number of winning times during the pair-wise comparison. The training and testing were done in a leave-one-out manner at protein (target) level, which meant each target (decoy set) was tested on the models trained on all other targets (decoy sets).

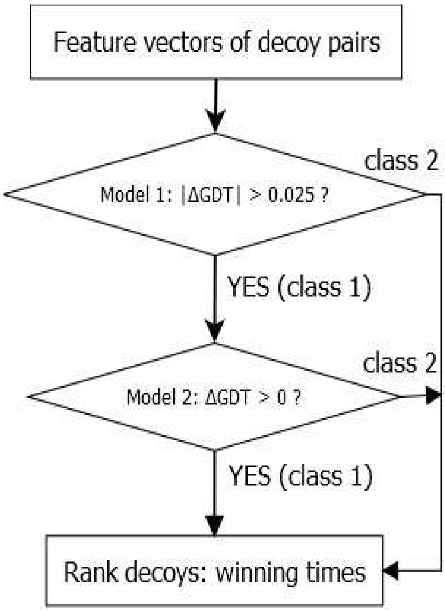
Feature vectors of decoy pairs
Model 1: |ΔGDT| > 0.025 ?
class 2
YES (class 1)
Model 2: ΔGDT > 0 ?
class 2
YES (class 1)
Rank decoys: winning times

Figure 6.2: Neural Networks for Pairwise Comparison

6.2.3 Dataset

We applied the method to three benchmark datasets from different model prediction methods. Each target (protein) had hundreds of decoys. The best decoy in each target had a GDT score greater than 0.4, which ensured that the pool contained reasonably good decoys. The first dataset contained 56 targets with decoys generated by I-TASSER ab initio modeling method (http://zhanglab.ccmb.med.umich.edu/decoys/). The second dataset consisted of 35 CASP 8 targets predicted by Rosetta or Robetta. The third dataset contained 50 CASP 9 targets with decoys generated by our in-house template-based model generation tool MUFOLD. Figure 6.3, 6.4 and 6.5 show the GDT distribution information, i.e. maximum, average and minimum GDT of each dataset respectively.

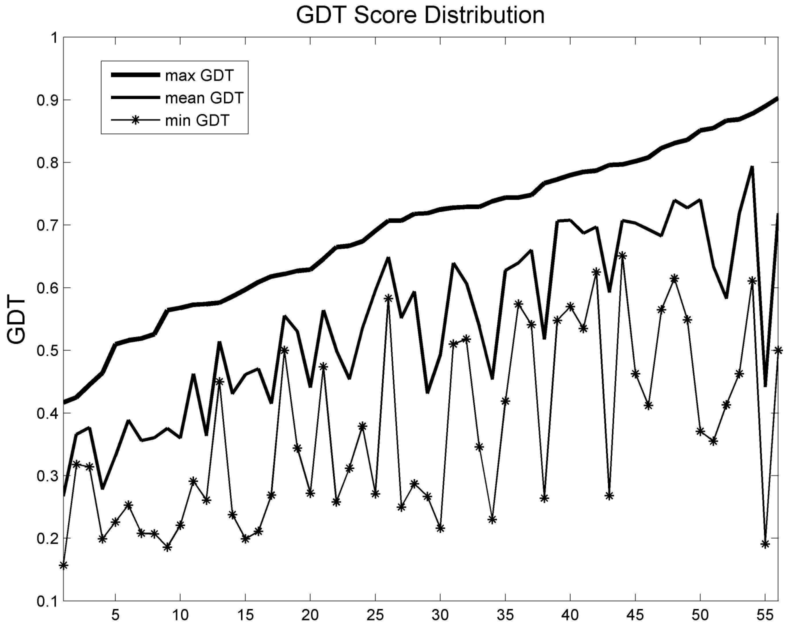

Figure 6.3: GDT score distribution of Benchmark 1

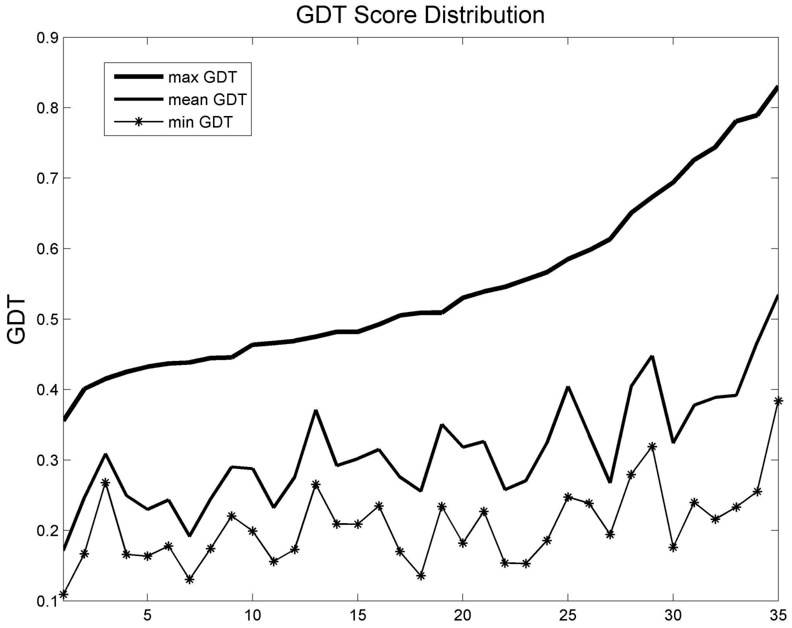

Figure 6.4: GDT score distribution of Benchmark 2

6.3 Result

In the test, each score was used to rank the decoys of a given protein. We studied the selection performance using three measures to compare each method. In the following comparison tables, "GDT1" is the average GDT score of top 1 model; "avgGDT5" is the average of the mean GDT score of top 5 models; and "Spearman" is the average Spearman correlation coefficient.

6.3.1 Performance Statistics

As shown in Table 6.1, 6.2 and 6.3, CGDT has better performance than all other single scoring functions in terms of three measures. Specifically, in benchmark 1, although CGDT's top-1 selection performance is not significantly better than that

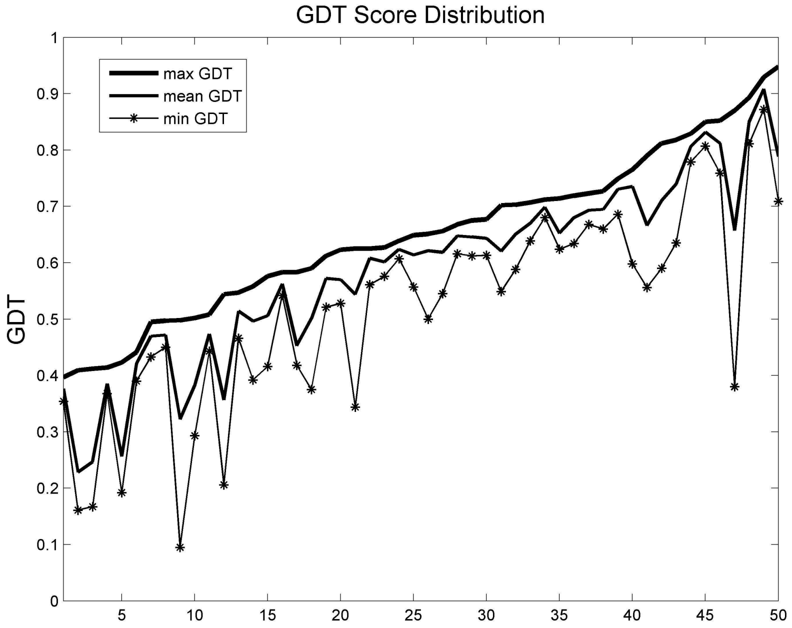

Figure 6.5: GDT score distribution of Benchmark 3

	Top1	**Best5**	**Mean5**	**Pearson**	**Spearman**
GDT	0.6946	0.6946	0.6767	1.0000	1.0000
CGDT	0.6058	0.6287	0.6045	0.7125	0.5845
DDFIRE	0.6006	0.6387	0.5906	0.5328	0.4405
OPUS-CA	0.5959	0.6367	0.5925	0.4949	0.4156
RW	0.5954	0.6381	0.5879	0.4912	0.4173
PSPS	0.5847	0.6161	0.5734	0.4213	0.3302
PWCom	0.6104	0.6353	0.6065	0.6169	0.6034
WQA [72]	0.6098		0.6034		

Table 6.1: Performance on Benchmark 1.

	Top1	**Best5**	**Mean5**	**Pearson**	**Spearman**
GDT	0.5449	0.5449	0.5219	1.0000	1.0000
CGDT	0.4255	0.4622	0.4060	0.5303	0.5588
DDFIRE	0.3901	0.4666	0.3788	0.3065	0.2726
OPUS-CA	0.3763	0.4551	0.3663	0.3012	0.2742
RW	0.3662	0.4567	0.3696	0.3026	0.2766
PSPS	0.3435	0.4173	0.3534	0.2482	0.2462
PWCom	0.4529	0.4796	0.4309	0.5313	0.5616
WQA [72]	0.4446		0.4220		

Table 6.2: Performance on Benchmark 2.

	Top1	**Best5**	**Mean5**	**Pearson**	**Spearman**
GDT	0.6503	0.6503	0.6431	1.0000	1.0000
CGDT	0.6023	0.6160	0.6042	0.3105	0.3199
DDFIRE	0.6091	0.6255	0.6094	0.3315	0.3049
OPUS-CA	0.6054	0.6246	0.6085	0.2611	0.2395
RW	0.6008	0.6215	0.6056	0.2345	0.2233
PSPS	0.5987	0.6166	0.6002	0.2457	0.2307
PWCom	0.6131	0.6271	0.6136	0.3328	0.3377

Table 6.3: Performance on Benchmark 3.

of other feature scores, its correlation (Spearman: 0.5845) is much higher than the others, among which DDFIRE is the best (Spearman: 0.4403). In benchmark 2, CGDT is significantly better than OPUS-CA, DDFIRE, RW and PSPS in terms of all three measures. Its top-1 selection performance (average GDT: 0.4255) has more than 3 GDT points than DDFIRE (0.3901), which is the best among remaining feature scores. In benchmark 3, the top-1 selection performance of all feature scores are similar, but in terms of Spearman correlation, CGDT is still the best (0.3199) with DDFIRE at the second place (0.3049).

From Table 6.1, 6.2 and 6.3, we can see that PWCom is significantly and consistently better than CGDT in three benchmarks. Notably, in benchmark 2, the top-1 GDT performance of PWCom is much higher than that of CGDT, with the improvement of 0.4529 - 0.4255 = 0.0274. In the other two benchmarks, PWCom score still improves in top-1 average GDT over CGDT, and even more over single scoring functions. As for Spearman correlation, PWCom is consistently better than CGDT in all three benchmarks.

6.3.2 Case Study

From the average performance, CGDT is consistently better than single scoring functions such as OPUS-CA, RW and DDFIRE. Here we show some individual cases from these benchmark datasets to see more detailed comparison. The bigger black spot in the following figures are the selected best decoy according to the scores.

	Top1	**Mean5**	**Pearson**	**Spearman**
GDT	0.6295	0.6179	1.0000	1.0000
CGDT	0.5446	0.5268	0.7681	0.7551
DDFIRE	0.4464	0.4696	0.2879	0.2675
OPUS-CA	0.5670	0.5330	0.4083	0.3870
RW	0.5134	0.5411	0.2025	0.1437
PSPS	0.4330	0.5035	0.1764	0.1739
PWCom	0.5804	0.5339	0.7674	0.7556

Table 6.4: Comparison of 1NE3 from benchmark 1

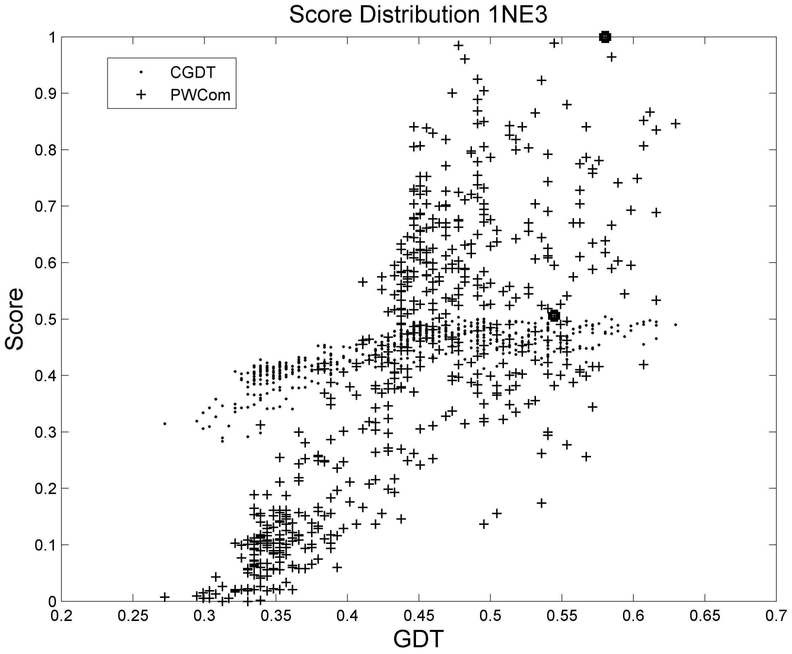

Figure 6.6: Distribution of CGDT and PWCom for 1NE3 from benchmark 1. The big black spot on the top is the selected best decoy according to PWCom and the one at the bottom according to CGDT

	Top1	**Mean5**	**Pearson**	**Spearman**
GDT	0.5900	0.5816	1.0000	1.0000
CGDT	0.5000	0.4896	0.4798	0.3924
DDFIRE	0.5770	0.5808	0.7965	0.8158
OPUS-CA	0.5670	0.5670	0.8640	0.8886
RW	0.5770	0.5744	0.7504	0.7709
PSPS	0.5350	0.5150	0.3532	0.3013
PWCom	0.5810	0.5808	0.8624	0.8742

Table 6.5: Comparison of T0527 from benchmark 3

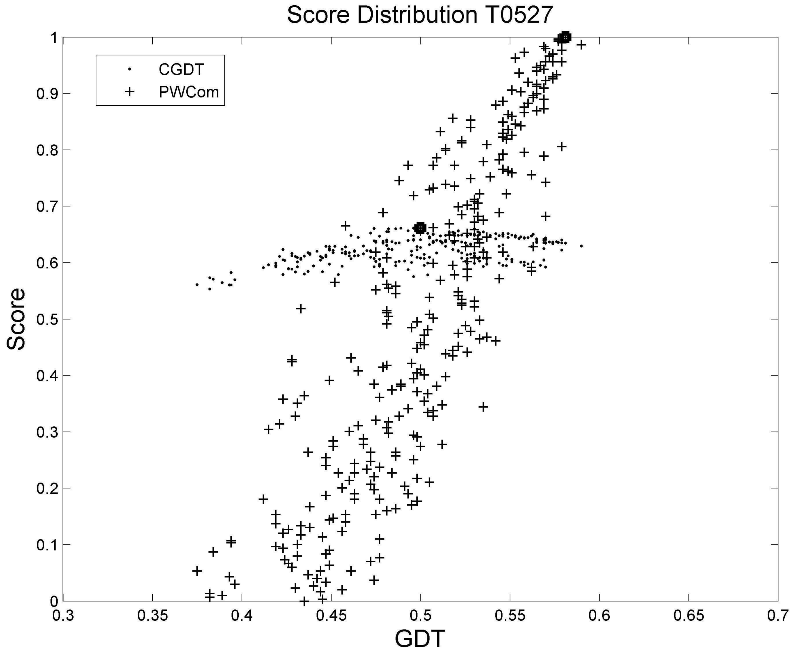

Figure 6.7: Distribution of CGDT and PWCom for T0527 from benchmark 3. The big black spot on the top is the selected best decoy according to PWCom and the one at the bottom according to CGDT

	Top1	**Mean5**	**Pearson**	**Spearman**
GDT	0.7810	0.7591	1.0000	1.0000
CGDT	0.7762	0.6852	0.8921	0.9098
DDFIRE	0.4000	0.4724	0.4226	0.3677
OPUS-CA	0.2595	0.4586	0.1838	0.1514
RW	0.3786	0.4029	0.2846	0.2600
PSPS	0.6071	0.5919	0.4971	0.4990
PWCom	0.7333	0.6819	0.8471	0.8954

Table 6.6: Comparison of T0396 from benchmark 3

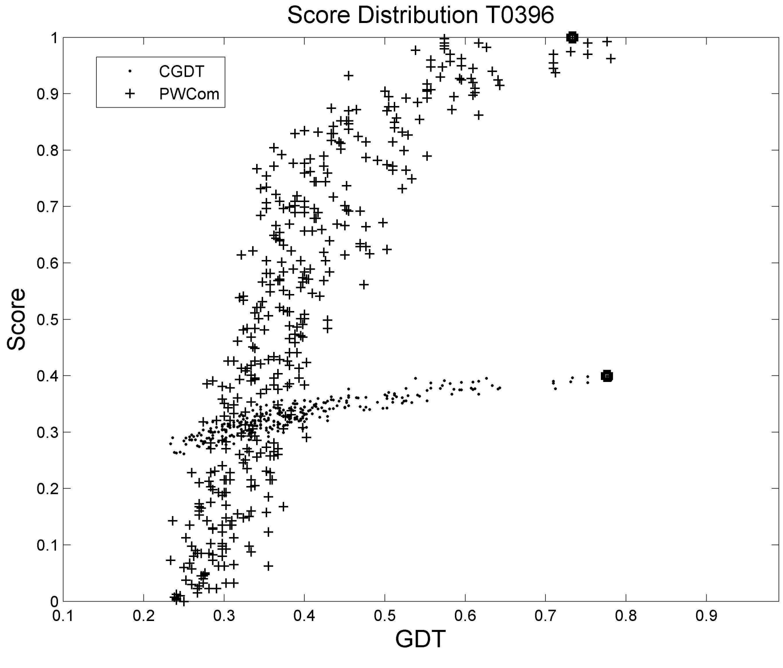

Figure 6.8: Distribution of CGDT and PWCom for T0396 from benchmark 2. The big black spot on the top is the selected best decoy according to PWCom and the one at the bottom according to CGDT

6.4 Discussion

Our new approach combined the advantages of consensus GDT method and single scoring functions through pairwise comparison and a two-stage machine-learning scheme. Consensus GDT method depends on the decoy distribution and relies on geometric information of protein structures only, while single scoring functions produce a wide range of values for different decoys, which makes their scores unstable and noisy. Our method tries to capture the correlation between score differences and actual structural difference as well as the complementarity among these scores. The resulting score (PWCom) is less noisy and more correlated to the real GDT score with respect to the native structure.

Our test result shows that PWCom was better than CGDT or single scoring functions in selection performances (GDT1 or avgGDT5) and correlations. PWCom is also better than our previous method WQA. This may be because WQA trained a SVM to directly map feature scores like CGDT, OPUS-CA score etc. to actual GDT scores of decoys, which is less stable generally when applied to different kinds of structural models. In addition, the weights of WQA for single scoring functions were optimized through quadratic programming, which required much more computation than PWCom.

PWCom combines CGDT and single scoring functions. Its performance is affected by the performances of the individual scores. For example, in the target shown in Figure 6.8 and Table 6.6, PWCom is worse than CGDT. Like CGDT and WQA, PWCom was also inevitably affected by the decoy distribution. Comparing Table 6.2 to Tables 6.1 and 6.3, we can see PWCom score got more improvement over other scores in benchmark 2 than those in benchmark 1 and 3. For example, in benchmark

2, the top-1 selection performance of PWCom was 0.4529, while the best of others was CGDT (0.4255). The improvement was 0.4529 - 0.4255 = 0.0274; while in benchmarks 1 and 3, the improvement was less significant. Comparing the decoy distributions of benchmark 2 to 1 and 3, the gap between maximum and mean GDT curve in Figure 6.4 was much bigger than that of Figure 6.3 and 6.5 [72]. And also, for quite a few targets in benchmarks 1 and 3, the gap between maximum and minimum GDT was quite small. This may explain that the average top-1 selection performance of single scoring functions was close to that of CGDT in these two benchmarks. In spite of this, in terms of Spearman correlation, CGDT was still better than single scoring functions.

This method still has significant room to improve as the training errors and parameter optimization problem may exist. We empirically chose 0.025 as the cutoff for neural-network model 1 and 0.01 to screen training data for neural-network model 2. Large-scale training and testing may help find better values for these cutoffs and other parameters in this approach. On the other hand, in terms of model training itself, we trained two neural-network models to predict whether two decoys are similar and which one is better than the other. An alternative method is regression, which might help further improve over classification methods. Finally, testing more feature scores and their combinations may also lead to more significant improvements.

6.5 Modification For CASP 10 Quality Assessment

We modified this method specifically to attend the QA session in CASP10, as shown in the flowchart. The basic idea is to compare any two decoys in terms of their structure quality first and then combine all the comparisons for QA of each decoy. First, the difference between feature vectors of a decoy-pair A and B were input to two independent neural network models to decide whether A or B is closer the native structure, in terms of GDT score. The first model was to judge whether two decoys are significantly different. If yes, the second model was used to decide which one of the two was better.

The feature vector for each decoy included

1. Structural environment fitness score between sequence and decoy structure.
2. Secondary structure (SS) matching score between the true secondary structure (SS_d) of decoy computed by DSSP
3. Solvent accessibility (SA) matching scores
4. Naive consensus GDT Scores of each decoy at eight thresholds. $CGDT_i, i = 1, 2, .., 8$ using the definition of 4.3 and 4.6
5. Mean square error between predicted angles and actual decoy angles. The Mean Square Error (MSE) between the true ϕ and ψ angles and the predicted ones respectively, which is computed by SPINE [79].

The neural networks were trained and tested on CASP9. 44 targets were used as test cases and the remaining were used as training data. For each decoy set, the top

150 were selected using the naive consensus GDT on the original data set.

	$top1$	$\overline{top5}$	Spearman
GDT	0.6412	0.6243	1.0000
CGDT	0.5861	0.5851	0.8408
PWCom	0.5958	0.5904	0.8499

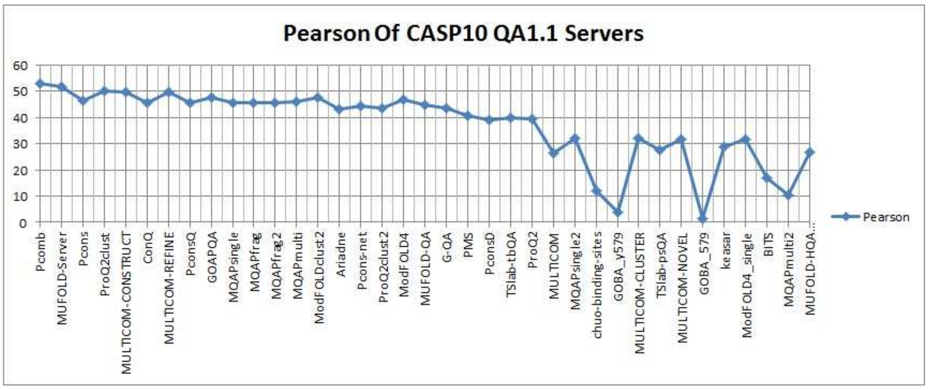

Figure 6.9: Pearson Score of CASP10 QA Servers

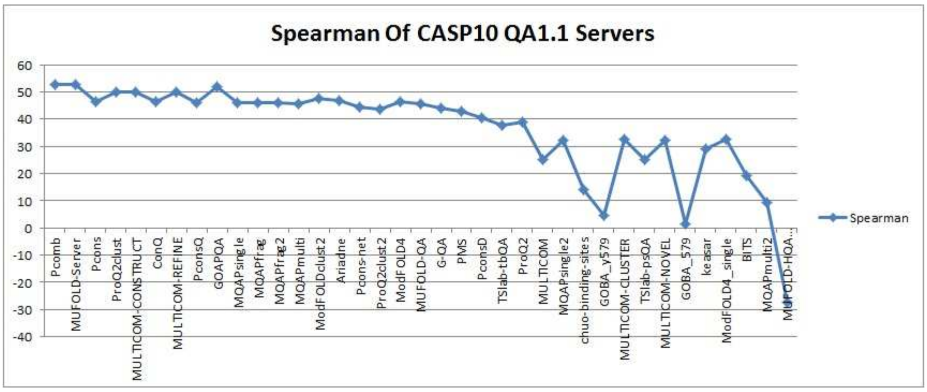

Figure 6.10: Spearman Score of CASP10 QA Servers

	Pearson	**ZScore of Pearson**	**FisherZ**	**ZScore of FisherZ**	**Spearman**
Pcomb	52.92	52.77	68.21	54.14	52.77
MUFOLD-Server	51.49	44.61	65.94	44.54	52.61
Pcons	46.39	38.44	58.94	37.30	46.58
ProQ2clust	50.11	37.71	65.48	38.42	50.00
MULTICOM-CONSTRUCT	49.62	36.90	65.36	38.68	49.80
ConQ	45.71	36.61	55.21	32.15	46.33
MULTICOM-REFINE	49.80	36.56	65.03	37.44	50.12
PconsQ	45.67	35.60	57.83	34.21	45.97
GOAPQA	47.43	32.23	63.12	40.93	51.84
MQAPsingle	45.70	31.65	59.39	34.29	45.87
MQAPfrag	45.70	31.65	59.39	34.29	45.87
MQAPfrag2	45.70	31.65	59.39	34.29	45.87
MQAPmulti	45.90	30.43	60.06	31.04	45.80
ModFOLDclust2	47.56	29.34	61.33	29.01	47.70
Ariadne	43.23	27.49	56.91	31.47	46.67
Pcons-net	44.47	26.98	57.97	26.77	44.49
ProQ2clust2	43.32	25.93	54.98	25.23	43.54
ModFOLD4	46.67	23.73	59.49	21.73	46.43
MUFOLD-QA	44.59	22.62	57.31	23.55	45.50
G-QA	43.48	17.87	55.81	19.86	44.01
PMS	40.71	11.24	46.72	4.79	42.73
PconsD	39.02	8.34	47.98	4.40	40.40
TSlab-tbQA	39.80	5.98	48.58	4.16	37.75
ProQ2	39.35	5.20	44.38	-2.38	39.12
MULTICOM	26.30	-17.13	29.75	-22.57	25.10
MQAPsingle2	32.16	-18.28	40.68	-15.64	32.29
chuo-binding-sites	11.89	-20.03	15.34	-18.77	14.24
GOBA-y579	3.86	-20.84	4.01	-21.66	4.44
MULTICOM-CLUSTER	32.05	-23.36	35.46	-30.42	32.70
TSlab-psQA	27.61	-23.94	30.36	-29.99	25.28
MULTICOM-NOVEL	31.74	-25.83	35.45	-32.09	32.38
GOBA-579	1.25	-30.05	1.37	-28.81	1.41
keasar	28.74	-31.96	32.01	-36.62	28.95
ModFOLD4-single	31.71	-32.13	39.32	-35.13	32.83
BITS	17.09	-83.04	20.12	-82.06	19.11
MQAPmulti2	10.24	-98.37	14.33	-94.91	9.50
MUFOLD-HQA	-26.79	-256.55	-28.95	-231.64	-27.33

Table 6.7: Scores of CASP10 QA Servers

Chapter 7

A New Hidden Markov Model for Protein Quality Assessment Using Protein Sequence-Structure Compatibility

7.1 Introduction

In Chapter 5 and 6, we proposed two differnt methods to combine single model scoring functions and consensus GDT information for protein structure quality assessment (QA) purpose. Single model scoring functions capture the sequence-structure relationship among proteins and play a critical role in protein structure prediction, such as fold recognition, threading alignment (or sequence-structure alignment), and protein structure quality assessment. In this work, we developed a new Hidden Markov Model (HMM) to assess the compatibility of protein sequence and structure for capturing their complex relationship. More specifically, the emission of the HMM consists of protein local structures in angular space, secondary structures, and sequence profiles. This model has two capabilities: (1) encoding local structure of each position by jointly considering sequence and structure information, and (2) assigning a global score to estimate the overall quality of a predicted structure, as well as local scores to assess the quality of specific regions of a structure, which provides useful guidance for targeted structure refinement.

To measure the sequence-structure compatibility, the structure environment of a protein residue is specified by a number of variables. Then a score is assigned for the observation of an amino acid type occurring in the structure environment. Some simple measures have been widely used in threading alignment methods [80, 81, 14, 13]. For example, the secondary structure matching score is the match ratio between the predicted secondary structure from an amino acid sequence and the actual ones calculated from the template structure. Another one is the environmental fitness score, which measures the propensity of an amino acid type to appear in the structure environment specified by three types of secondary structures (Helix, Sheet and Coil) or three types of solvent accessibilities (Buried, Intermediate and Exposed). More advanced work has been done to address this problem, which mainly differ in the definition of structure environment and the method to calculate the compatibility score (probability or pseudo-energy) [82, 83, 84, 85, 86]. For example, in [83], three-dimensional profiles were derived from native structures to measure the compatibility in which the structural environment was defined by parameters such as the area of the side chain that is buried and the secondary structure type; in [86], more complex structural environment was defined in which side chain packing and hydrogen bonding were used as one of its four measurement functions; a neural network was trained to predict the probability of observing an amino acid type given the structural environment [82].

It is commonly observed that proteins have recurrent local sequences and structure patterns. The sequence-structure dependency at local levels leads researchers to use the Hidden Markov Model (HMM) approach to describe the proteins. In [87, 88], an HMM was used to compress protein three-dimensional conformations into a one-dimensional series of letters of a structural alphabet, where the emission of the HMM at each state is a multi-dimensional Gaussian distribution for the distance configuration of four consecutive neighboring C_α atoms. In [89], a more complex HMM based method, HMMSTR was proposed to capture local sequence-structure correlations, in which four types of emissions were defined, i.e. amino acid types, secondary structure types, backbone angle region (i.e. using the (ϕ, ψ) plot to partition the protein chain into several non-overlapping regions) and structural context descriptor (for example, distinguishing a hairpin turn from a diverging turn). However, the structure information contained in this HMM is not informative enough as it only contains discretized backbone angle region types and secondary structures.

A number of knowledge-based scoring functions such as OPUS-CA [17], DFIRE [18] and RW [61] for protein structure quality assessment can also be considered as sequence structure compatibility measures at global levels. Most of these scores are weighted sums of several energy terms obtained through statistics over native structures. For example, OPUS-CA uses the distance distributions of residue pairs and DFIRE constructs residue-specific all-atom potential of mean force from a database of native structures.

More advanced descriptions of local structures are important for improving HM-

M's capability of capturing sequence structure relationship. For this purposes, we defined new emission functions for the HMM to describe the local sequence-structure relationship. The structural emission contains information for every four consecutive C_α atoms, which is represented as three-dimensional Gaussian distributions in the angular space. Another important emission is about the sequence profile, which contains the distribution of 20 types of amino acids and the insertion and deletion during the evolution process. The HMM model has two capabilities: (1) encoding local structure of each position by considering of the local sequence-structure relationship and (2) assigning a global score to estimate the overall quality of a predicted structure, as well as local scores to assess the quality of a specific structural segment.

The new model was tested and compared with the state-of-art single structure QA methods. Test results demonstrated that the model can achieve better overall selection performance than the comparing QA methods.

7.2 Methods

Our goal is to construct a new Hidden Markov Model to encode the compatibility between protein sequence and structure, and capture their complex relationships. First, the emission of the HMM is defined based on protein local structures in the angular space, secondary structures, and sequence profiles. Second, with a training data set, the proposed HMM was trained using the Expectation Maximization (EM) algorithm.

7.2.1 Sequence and Structure Representation

For each protein, we calculated the sequence profile matrix $PSFM[]$ and $SEQ[]$ from the output alignments of PSI-BLAST [12] running against the non-redundant (NR) sequence database (released in 2010, `ftp://ftp.ncbi.nih.gov/blast/db/`) three rounds with an E-value cutoff of 0.001. Each row of the matrix $PSFM[]$ is a vector of 21 dimensions containing frequencies for 20 types of amino acids and indels (insertions and deletions) in the multiple sequence alignment (MSA), while $SEQ[]$ only contains amino acids distribution information with each row being 20 dimensions.

The local structure of a protein is represented in the angular space according to the work of [78]. Specifically, for each residue x_k in a protein structure, we calculated an angle triplet $(\theta_k, \tau_k, \theta_{k+1})$ for four consecutive C_α atoms $(x_{k-2}, x_{k-1}, x_k, x_{k+1})$, where θ_k is the bend angle of (x_{k-2}, x_{k-1}, x_k), τ_k is the dihedral angle of $(x_{k-2}, x_{k-1}, x_k, x_{k+1})$ and θ_{k+1} is the bend angle of (x_{k-1}, x_k, x_{k+1}), as shown in Figure 7.4. Let $x_k \equiv (\theta_k, \tau_k, \theta_{k+1})$, a protein of length L is represented by a list of x_k, where k goes from 3 to $L-1$.

The probability distribution of an angle triplet x for the entire structure space was approximated by a Gaussian mixture model of 17 components [78], i.e.,

$$P(x) = \sum_{i=1}^{17} \pi_i N_i(x; u_i, \Sigma_i) \ ; \tag{7.1}$$

$$N_i(x; u_i, \Sigma_i) = (2\pi)^{-3/2} |\Sigma_i|^{-1/2} e^{\frac{1}{2}(x-u_i)\cdot\Sigma_i^{-1}\cdot(x-u_i)} \tag{7.2}$$

where $N_i(x; u_i, \Sigma_i)$ is the i-th normal distribution, π_i is the corresponding weight, and u_i, Σ_i are the mean vector and covariance matrix, respectively.

7.2.2 HMM Definition

Let $Y = [y_1, y_2, .., y_T]$ and $O = [o_1, .., o_T]$ be the state sequence and observation sequence of length T, respectively. The basic form of HMM, denoted by Λ, can be written as the joint probability of Y and O,

$$p(Y, O|\Lambda) = \pi_{y_0} \prod_{t=1}^{T} a_{y_{t-1}y_t} \prod_{t=1}^{T} b_{y_t}(o_t) \tag{7.3}$$

where y_t is the state of position t, $a_{y_{t-1}y_t}$ is the transition probability from state y_{t-1} to y_t and $b_{y_t}(o_t)$ is the emission probability for state y_t. In this work, the emission probability is defined as

$$\begin{aligned} b_{y_t}(o_t) \quad = \quad & \left[\sum_{k=1}^{17} w_{y_t,k} \cdot N_k(x_t; u_k, \Sigma_k)\right] \left[\sum_{a=1}^{21} f_{y_t,a} \cdot PSFM[t, a]\right] \\ & \left[\sum_{b=1}^{20} s_{y_t,b} \cdot SEQ[t, b] \cdot env(b, SS_d, SA_d)\right] \end{aligned} \tag{7.4}$$

The first part of Eqn. 7.4 describes the structure information, where $N_k(x_t; u_k, \Sigma_k)$ is the k-th Gaussian function, whose parameters were taken from Eqn. 7.2 and x_t is the angle triplet defined above. The second part of Eqn. 7.4 is the sequence profile distribution, where $PSFM[]$ is the sequence profile matrix. The third part of Eqn. 7.4 describes the sequence-structure distribution, where $evn(b, SS_d, SA_d)$ is the probability score of amino acid type b appearing in the structure environment specified by three types of secondary structures SS_d and three types of solvent accessibilities SA_d [80, 81]. For the simplicity of implementation, currently only parameters $w_{y_t,k}$, $f_{y_t,a}$ and $s_{y_t,b}$ in the emission function need to be trained by the learning procedure. Therefore the number of states is set to 17 by default [78], which can be optimized by

Bayes Information Criteria (BIC) or other model selection techniques such as cross validation. We have tried different number of states, the test results did not show any significant improvement.

7.2.3 Scoring Structures by HMM

Once the HMM is given, we can assign a score to measure the global sequence structure compatibility of a protein by

$$V = \arg\max_{Y*} P(Y, O|\Lambda) \tag{7.5}$$

or

$$Z = \sum_{Y} P(Y, O|\Lambda) \tag{7.6}$$

where Λ is the model, O is the observation and Y is the state sequence. Practically, the probability given by Eqn. 7.6 is more robust than that of Eqn. 7.5. Throughout this work, we use HMM.Z to denote the score defined by Eqn. 7.6.

7.2.4 Training Data Set

Considering the diversity of structural space, each test protein will have its own training dataset. First, for each protein in testing data, we use PSI-BLAST to search the sequence against the PDB [1] database to find out significant templates, remove those templates having more than 70% sequence identity to test sequence. If no or too few templates remains, we add a random subset from the following default data set to constitute the training data set of about 200 chains for this protein. The default data set for training is extracted from PDB according to the following steps:

1. Filtering the entire PDB database with the following setting:

   - X-Ray structure with resolution less than 2.0 Å.
   - All residues have 3D coordinates, at least for backbone atoms.
   - Sequence length $L \in [50\ 300]$.

2. Remove all chains that have sequence similarity higher than 70% to any test sequence using BLAST [12].

3. Remove redundant proteins within the training data set by decreasing the mutual sequence similarity to 40% using CD-Hit [49].

With this data set, the proposed HMM is trained using the Expectation Maximization (EM) algorithm.

7.2.5 Test Data Set

We tested the method in protein structure selection scenario using Global Distance Test score (GDT) [20] as structure similarity measure. GDT is defined as $\frac{N_1+N_2+N_4+N_8}{4L}$, where $N_i, i = 1, 2, 4, 8$ is the number of positions with distance less than $i\AA$ after optimal structural superimposition and L is the protein length. Therefore, GDT value being 1 means two structures are exactly the same. We applied the method to four benchmark datasets from different protein structure prediction methods. The first dataset, I-TASSER-DATA, contains 56 targets (proteins) with decoys generated by I-TASSER *ab initio* method [7, 8] (`http://zhanglab.ccmb.med.umich.edu/decoys/`). The second one, Modeller-DATA, has 55 targets, with decoys generated by Modeller

[10]. In both datasets, each target has $\sim$ 500 decoys, and the best decoy for each target has a GDT score greater than 0.4, which ensures that the pool contains at least some good-quality decoys. Figure 7.4 (a) and (b) show the GDT distribution information, i.e., maximum, average and minimum GDT of I-TASSER-DATA and Modeller-DATA, respectively. The third benchmark data has 20 targets, containing FISA, LMDS_V2 and SEMFOLD from the Decoys 'R' Us decoy set [90]. The fourth one is HG_STRUCTAL from Decoys 'R' Us containing 29 targets.

7.3 Results

We compared the score HMM.Z with the state-of-art QA tools, OPUS-CA, DFIRE and RW, all of which make use of global contact information in protein structures. We also compared the score HMM.Z with the secondary structure matching score ($SSMatch$) and environmental fitness ($Fitness$) which is the summation of compatibility score of all positions in a protein structure. In the test, scores were used to rank the decoys of a given protein. In the following tables, we use the criteria below to study the selection and ranking performance:

- Top1: the GDT score of the top-1 selected model;
- Top5: the best GDT of selected top 5 models;
- Mean5: the average GDT score of the top 5 models;
- Pearson: Pearson correlation coefficient between the QA score and the true GDT score,

- Spearman: Spearman correlation coefficient between the QA score and true GDT score.

7.3.1 Performance of Global QA

Tables 7.1 and 7.2 show the global QA performance of score HMM.Z, compared with OPUS-CA, DFIRE and RW on I-TASSER-DATA and Modeller-DATA, respectively. We can see that HMM.Z achieved the best average top-1 selection performance on both datasets and the best correlation (Pearson and Spearman) to GDT score on Modeller-DATA. In particular, in Table 7.1 HMM.Z has comparable performance to the three QA methods in which OPUS-CA is the best. But in Table 7.2, HMM.Z achieved the best top-1 selection performance (GDT: 0.60), which is significantly better than that of OPUS-CA (0.58) and RW (0.57). Figure 7.4 compares the top-1 selection performance of HMM.Z to that of OPUS-CA on I-TASSER-DATA and Figure 7.4 compares HMM.Z to DFIRE on Modeller-DATA. We can find that for many targets, the decoys selected by our method are significantly better than those from OPUS-CA or DFIRE. In Figure 7.4, although the average performance is similar to that of OPUS-CA, we can see that for quite a number of targets HMM.Z selected almost the best model in the pool. The result in Figure 7.4 shows that HMM.Z outperformed DFIRE, which ranked the best in the three QA methods on Modeller-DATA. Table 7.3 compares the global QA performances on FISA, LMDS_V2 and SEMFOLD together. As we can see, HMM.Z achieves the best selection performance with top-1 selection performance of 0.485 which is 0.016 higher than the second best method, DFIRE, and 0.021 higher than OPUS-CA and RW. Figure 7.4 shows the detailed comparison between HMM.Z and DFIRE. Table 7.4 shows the average performance on HG_STRUCTRAL data set. HMM.Z has nearly the same average performance as OPUS-CA, DFIRE and RW, all of which are close to the limit. From Figure 7.4, we see that except for one case, HMM.Z can select almost the best decoy from the decoy pool. Tables 7.1-7.4 also compare the global QA performance of HMM.Z with *Fitness* and *SSMatch*. Overall, HMM.Z is consistently better than *Fitness* and *SSMatch* in terms of selection and correlation performance on four benchmark datasets.

7.4 Discussion and Conclusions

Our Hidden Markov Model is modeled in the sequence-structure space, in which the emission contains sequence profile information and continuous (instead of discrete) structural content. As one of its advantages, HMM considers the dependency between adjacent local sequences and structures. The emission of HMM contains rich information about the sequence profile, secondary structures, solvent accessibilities as well as local conformation represented in the angular space. The model for each test protein is trained on its homologous structures (if available) obtained by template searching, which greatly reduce the noise in the training procedure and help capture the true relationship between the sequence and the native structure. From the test results, comparing to the three single model QA methods OPUS-CA, DFIRE and RW, our test results have shown clear improvement of score HMM.Z in selection performance on the second (Modeller-DATA) and third (FISA + LMDS_V2 + SEMFOLD) datasets and comparable performance on the first one (I-TASSER-DATA) and the fourth one (HG_STRUCTAL). From the detailed comparisons, we can conclude that for a significant number of cases HMM.Z is able to select almost the best model from the pool and achieve significant better selection performance than other comparing scores, which means our HMM method is more sensitive in selecting near-native structures.

However, our HMM method has room for improvement. In a few cases HMM.Z are significantly worse than the corresponding best method. One example is the 30th target in Figure 7.4, which is 2CR7 from I-TASSER-DATA. And another example is the one in Figure 7.4 from the FISA + LMDS_V2 + SEMFOLD data set. We manually checked the case of 2CR7. HMM mis-selected a protein decoy whose local structures are very similar to the native one, but having a different packing, as shown by Figure 7.4. Table 5 shows the pairwise GDT score between the native structure and the top-1 models selected by all the methods. RW and DFIRE also selected an incorrect decoy similar to the one selected by HMM.Z, while OPUS-CA chose a decoy with correct packing. This indicates that adding global pairwise contact information into our method for HMM.Z might lead to further improvement. We are investigating those cases that HMM.Z loses more than 10 GDT points to the best decoy for further possible improvement. As one of the future work, we will derive informative scores from this method for local structure assessment and compare with existing local QA methods.

Our HMM method can be used as a component tool for protein structure prediction to evaluate the global structure quality of predicted decoys. It will be released when the stand-alone tool is ready.

Figures and Tables

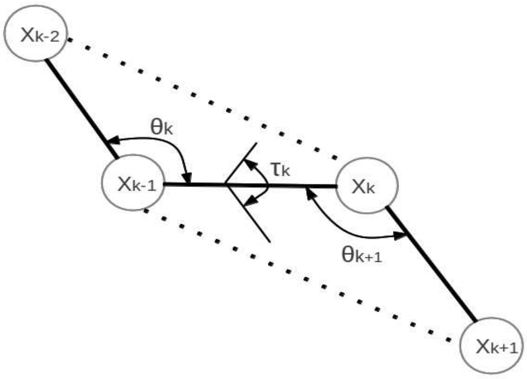

Figure 7.1: Angles of four consecutive C_α atoms.

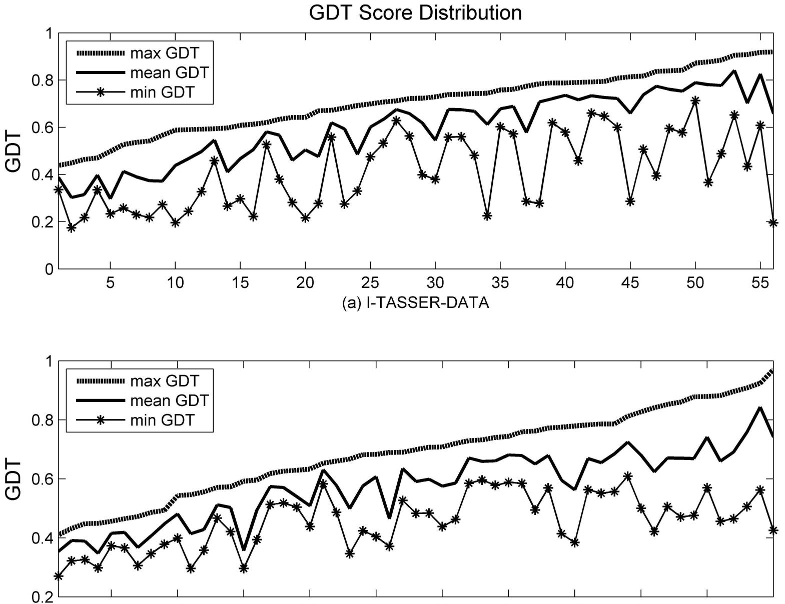

Figure 7.2: Decoy distribution of I-TASSER-DATA (a) and Modeller-DATA (b). The dashed curve shows the maximum GDT score, the solid curve shows the mean GDT score, and the stared curve shows the minimum GDT score in the pool for each target.

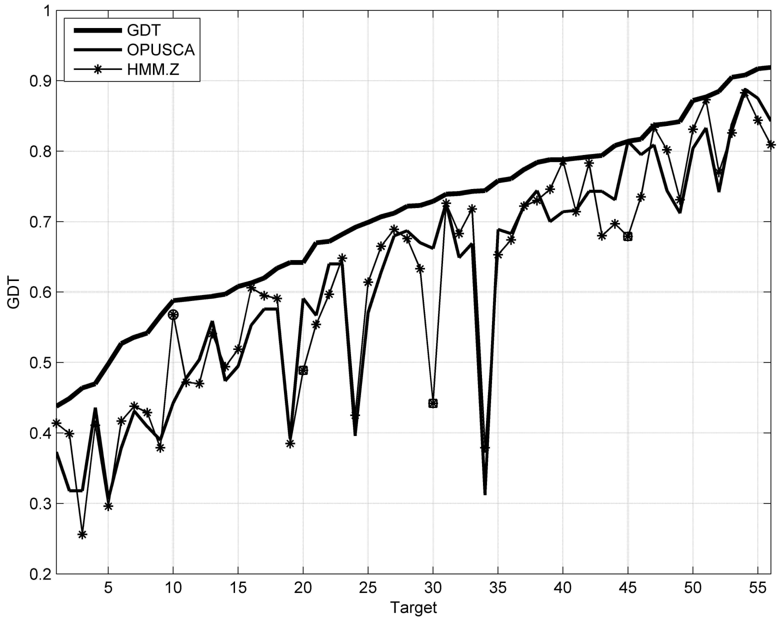

Figure 7.3: Detailed comparison of global QA on I-TASSER-DATA. The widest curve represents the best true GDT score of the decoy for each target. The middle curve shows the performance of OPUS-CA. The thinnest one represents the GDT score achieved by our method HMM.Z. The circled stars indicate that our method HM-M.Z performs significantly better than OPUS-CA on the corresponding targets. The boxed stars show that HMM.Z significantly underperforms over OPUS-CA on the corresponding targets.

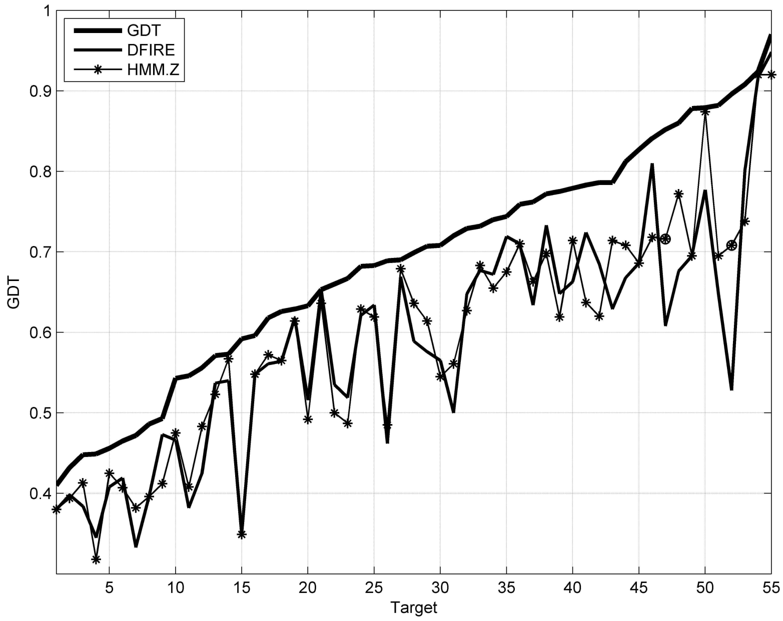

Figure 7.4: Detailed comparison of global QA on Modeller-DATA. The widest curve represents the best true GDT score of the decoy for each target. The middle curve shows the performance of DFIRE. The thinnest one represents the GDT score achieved by our method HMM.Z. The circled stars indicate that our method HMM.Z performs significantly better than DFIRE on the corresponding targets. The boxed stars show that HMM.Z significantly underperforms over DFIRE on the corresponding targets.

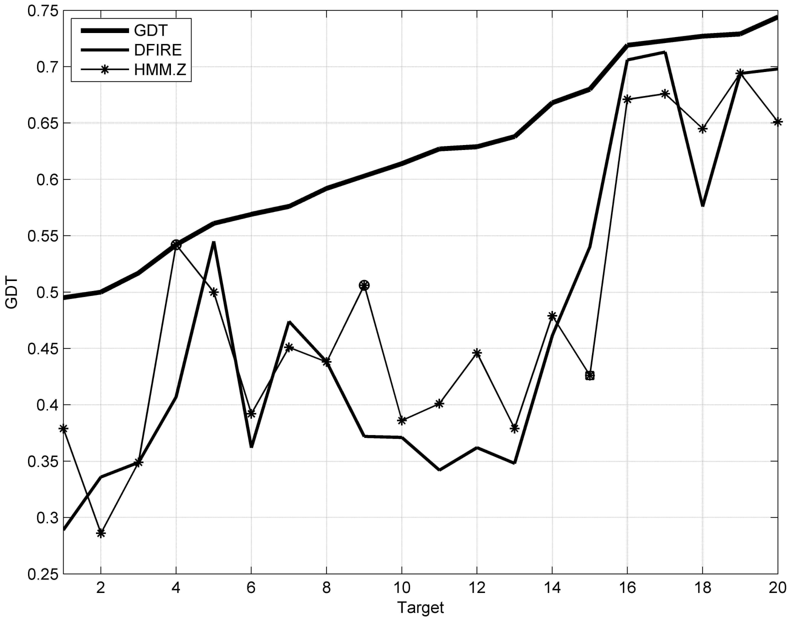

Figure 7.5: Detailed comparison of global QA on combined set of FISA, LMDS_V2 and SEMFOLD data. The widest curve represents the best true GDT score of the decoy for each target. The middle curve shows the performance of DFIRE. The thinnest represents the GDT score achieved by our method HMM.Z. The circled stars indicate that our method HMM.Z performs significantly better than DFIRE on the corresponding targets. The boxed stars show that HMM.Z significantly underperforms over DFIRE on the corresponding targets.

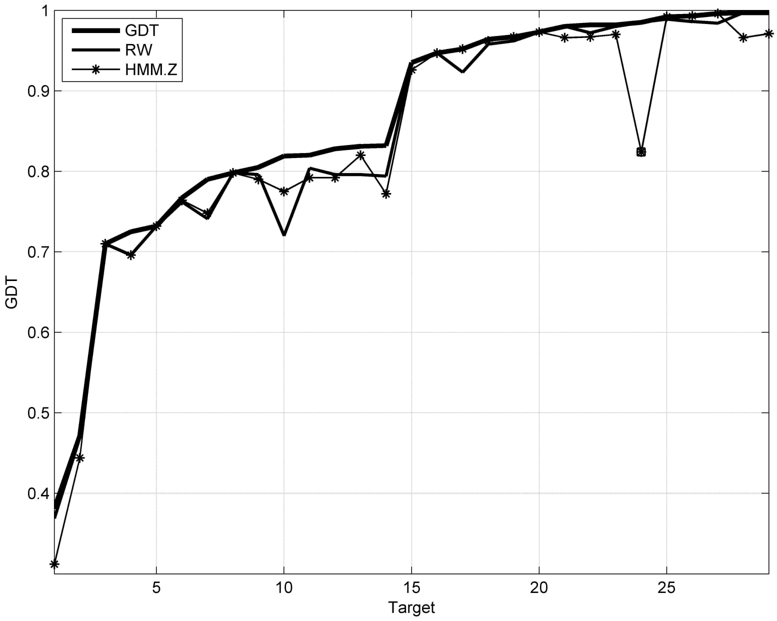

Figure 7.6: Detailed comparison of global QA on HG_STRUCTAL data. The widest curve represents the best true GDT score of the decoy for each target. The middle curve shows the performance of RW. The thinnest one represents the GDT score achieved by our method HMM.Z. The circled stars indicate that our method HMM.Z performs significantly better than RW on the corresponding targets. The boxed stars show that HMM.Z significantly underperforms over RW on the corresponding targets.

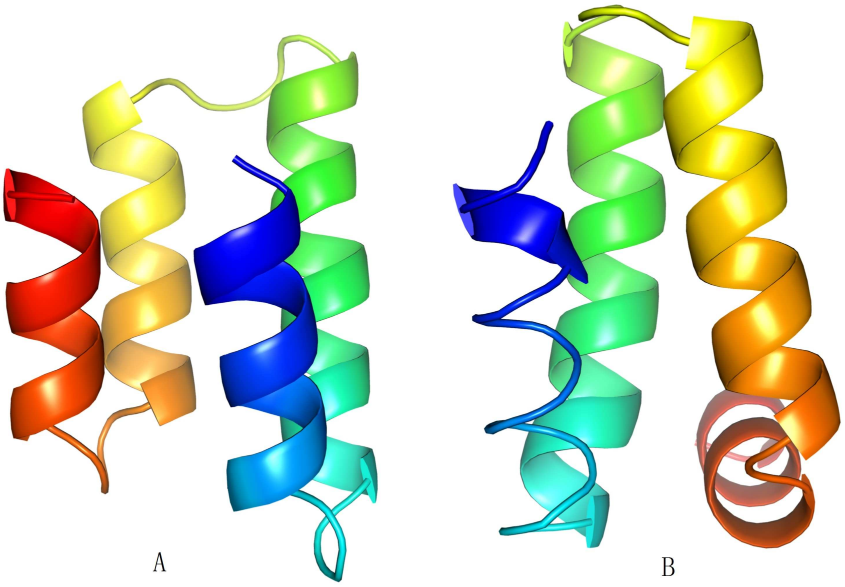

Figure 7.7: The sub-figure (A) shows the native structure of protein 2CR7 and (B) is the top-1 model selected by HMM.Z.

	Top1	**Best5**	**Mean5**	**Pearson**	**Spearman**
GDT	0.705	0.705	0.693	1.000	1.000
OPUS-CA	0.614	0.646	0.613	**0.322**	**0.237**
DFIRE	0.609	0.641	0.608	0.312	0.231
RW	0.610	0.636	0.609	0.278	0.196
Fitness	0.607	0.641	0.606	0.176	0.119
SSMatch	0.617	0.651	0.616	0.216	0.166
HMM.Z	**0.615**	**0.651**	**0.616**	0.265	0.192

Table 7.1: Global QA performance on I-TASSER-DATA.

	Top1	**Best5**	**Mean5**	**Pearson**	**Spearman**
GDT	0.688	0.688	0.675	1.000	1.000
OPUS-CA	0.579	0.627	0.584	0.192	0.175
DFIRE	0.587	0.623	0.585	0.175	0.157
RW	0.569	0.613	0.574	0.104	0.093
Fitness	0.558	0.621	0.567	0.018	0.020
SSMatch	0.578	0.624	0.580	0.075	0.067
HMM.Z	**0.594**	**0.631**	**0.593**	**0.227**	**0.205**

Table 7.2: Global QA performance on Modeller-DATA.

	Top1	**Best5**	**Mean5**	**Pearson**	**Spearman**
GDT	0.623	0.623	0.598	1.000	1.000
OPUS-CA	0.464	0.517	0.450	0.274	0.274
DFIRE	0.469	0.525	**0.468**	**0.288**	**0.282**
RW	0.463	0.524	0.465	0.268	0.268
Fitness	0.470	**0.542**	0.465	0.190	0.186
SSMatch	0.467	0.519	0.451	0.172	0.166
HMM.Z	**0.485**	0.525	0.464	0.236	0.218

Table 7.3: Global QA performance on data of FISA + LMDS_V2 + SEMFOLD.

	Top1	**Best5**	**Mean5**	**Pearson**	**Spearman**
GDT	0.860	0.860	0.836	1.000	1.000
OPUS-CA	0.840	0.858	0.823	0.779	0.739
DFIRE	0.844	0.856	0.824	0.806	0.756
RW	**0.847**	**0.858**	**0.824**	**0.812**	**0.759**
Fitness	0.826	0.854	0.803	0.740	0.592
SSMatch	0.789	0.845	0.795	0.680	0.625
HMM.Z	0.839	0.857	0.813	0.780	0.721

Table 7.4: Global QA performance on HG_STRUCTAL data.

	Native	OPUS-CA	DFIRE	RW	HMM.Z
Native	1.000	0.662	0.525	0.525	0.442
OPUS-CA		1.000	0.521	0.521	0.463
DFIRE			1.000	1.000	0.762
RW				1.000	0.762
HMM.Z					1.000

Table 7.5: Pairwise GDT of selected top-1 models for protein 2CR7 from I-TASSER-DATA. Native means the native structure of protein 2CR7,OPUS-CA means its selected top-1 model, and similarly for DFIRE, RW and HMM.Z.

Chapter 8

Protein Structural Model Assessment Based on Conditional Random Field

8.1 Introduction

In Chapter 5 and 6, we proposed two methods to combine knowledge based scoring functions with consensus GDT [20] score from different perspectives, in which the combining methods treated the input scoring functions as black boxes. And those scores can be replaced by any other scoring functions. In Chapter 7, we presented a new hidden moarkov model (HMM) based method to evaluate protein structure quality which did not rely on any other "black-box" scores and can be used as a scoring function to evaluate single models, just like OPUS-CA [17] and DFIRE [18]. Although the score given by the HMM model reflected the sequence-structure relationship of proteins to some extent, it can be improved from two different perspectives. The first one is the score given by the HMM is a confidence score, which does not indicate the exact structural quality of protein models. On the other hand, in terms of the descriptive power, HMM has its shortcomings when compared to conditional random field (CRF), which is a modern extension of HMM. From the HMM model analysis in Chapter 7 and Appendix B, we can see that the biggest shortcoming of HMM is its inconvenience to handle complex features and the simple form of the transition probability matrix, which limits the learning capability of HMM.

In this work, we formulate the protein structural quality assessment problem as sequence sequential labeling (SSL) problem and train a CRF model from sequence or structural features to predict the actual GDT score of decoys. In more details, the structural quality at each position of the decoy structure is labeled by one of the predefined states. We train a CRF model and predict the structural quality state at each position of the decoy structure. The first advantage of this method is that CRF is capable of handling complex sequence and structural features from different resources. The second one is the predicted structural state for each position can be used as local quality assessment for that position, and the overall quality score is simply the sum of the state sequence.

8.2 Method

8.2.1 Define the Target States

In this method, the protein quality assessment (QA) problem is formulated as SSL problem. Therefore, the first step is to define the states to indicate the structural quality at each position of the structure. One straightforward way to define the states is to discretize the distance between each residue to the corresponding one in the native structure after optimal superimposition. Let d_j denote such a distance at position j. For example, we can define state s_j as

$$s_j = \begin{cases} 1 & 0 < d_j <= 1 \\ 2 & 1 < d_j <= 2 \\ 3 & 2 < d_j <= 3 \\ 4 & 3 < d_j <= 4 \\ ... & ... \end{cases}$$

Also, there is another way to define the local structural states, which is based the GDT [20] score. For every decoy structure, we convert its GDT score with respect to the native structure to a vector during the training stage.

Expand GDT Score as GDT Vector

From the definition of GDT score, we see from Eqn. 4.3 that GDT_i is the sum of a binary vector, each 0/1 element of which indicates whether or not the distance d_j at protein position j is less than the corresponding cutoff i. Let $gdtVec_i$ denote such a binary vector for GDT_i, see Chapter 4 for more details. Then according to Eqn. 4.2 and 4.4, the GDT vector between two decoys p_1 and p_2 can be defined as

$$gdtVec(p_1, p_2) = gdtVec_1 + gdtVec_2 + gdtVec_4 + gdtVec_8 \tag{8.1}$$

The GDT score between p_1 and p_2 simply equals to $\frac{gdtVec(p_1,p_2)}{4*L}$, where L is the protein length. Now, the structural state s_j at position j of decoy p_k is

$$s_j^k = gdtVec(p_k, native)[j] \tag{8.2}$$

where $s_j^k \in \{0, 1, 2, 3, 4\}$. s_j^k is the target for CRF to learn during the training stage.

8.2.2 Model Definition

Similar to the CRF model for threading alignment in Chapter 2, the CRF model for QA is defined by the following conditional probability

$$P_\theta(s|X) = \frac{\sum_{j=1}^{L} F(s_{j-1}, s_j, X)}{Z(X)} \tag{8.3}$$

where $Z(X)$ is a normalization factor to make the right hand part be a probability and $F(s_{j-1}, s_j, X)$ is the potential function that indicates the likelihood of the structural state at protein position j, given the input feature set X.

There are at most 25 potential functions we need to train as there are total 5 different structural states. In reality, we ignore some potential functions as we don't have enough training data examples. For example, the transition $4 \rightarrow 0$ indicates a sudden change in the protein structure, which can be treated as an error. So the potential for this transition is forbidden. Potential functions $F(s_{j-1}, s_j, X)$ in this model is a non-linear function which is represented by a weighted sum of a set of regression trees.

8.2.3 Learning and Predicting with CRF

The model Eqn. 8.3 can be incrementally trained using functional gradient tree boosting methods [36].

$$\begin{aligned} F_m(u,v,X) &= F_0(u,v,X) + w_1 \Delta F_1(u,v,X) +, ..., \\ &+ w_m \Delta F_m(u,v,X) + b_m \end{aligned} \tag{8.4}$$

where b_m is the offset for $F_m(u,v,X)$, which is a regression tree fitted to the gradient at each step.

The features mainly include two parts. The first part is the sequence and structural features of a decoy which directly related to its sequence and structure properties. The second one is the consensus geometrical information obtained from a set of peer candidates. This information is a strong indication of the structural quality when the decoy comes from a structure pool which is dominant of good candidates. The following features are used for all types of state transitions.

1. Sequence profile similarity: sequence profile similarity score at sequence position j between the query sequence and template $\sum_a PSSM(i,a) \times PSFM(j,a)$.

2. Environmental fitness score: this score measures the propensity of an amino acid type a to appear in a structure type, which is specified by the combinations of three types secondary structure (Helix, Beta sheet, and loop) and three types of solvent accessibility (Fully buried, intermediate and fully exposed) [25, 26]. The environment fitness score is given by $\sum_a PSFM(j,a) \times F(env_i, a)$.

3. Secondary structure (SS) Match Score: suppose the secondary structure type

at template position j is SS_d, and a predicted secondary structure for query sequence is SS_p with confidence value C given by PSIPRED, the matching score is the probability of SS_d predicted to be (SS_p, C), which is specified by a look-up table.

4. Solvent accessibility (SA) matching scores: similarly to secondary structure match score, the predicted SA (SA_p) is done by SSPro [40] and the true SA (SA_d) for template is computed using DSSP with cutoff 25% (above which means the exposed state and otherwise the buried state). If the SA state is matched, the score is 1 otherwise 0.
5. Dihedral angle difference: the difference between the predicted ϕ and ψ angles and the actual ones from decoy structure. The predicted ϕ and ψ angles are computed by SPINE [79].
6. Consensus GDT information: the consensus GDT information for each decoy is similar to naive GDT score, as defined by Eqn. 4.6, except that the GDT score is represented by its vector form.

8.2.4 Dataset

We tested the method on two benchmark datasets. The first one contained 56 targets with decoys generated by I-TASSER ab initio modeling method (`http://zhanglab.ccmb.med.umich.edu/decoys/`), which was also used as benchmark dataset in Chapter 7. The second dataset consisted of 62 targets from CASP 10 QA category, section 2, each of which contained 150 decoys. Figures 8.1 and 8.2 show the GDT distribution information, i.e. maximum, average and minimum GDT of each dataset respectively.

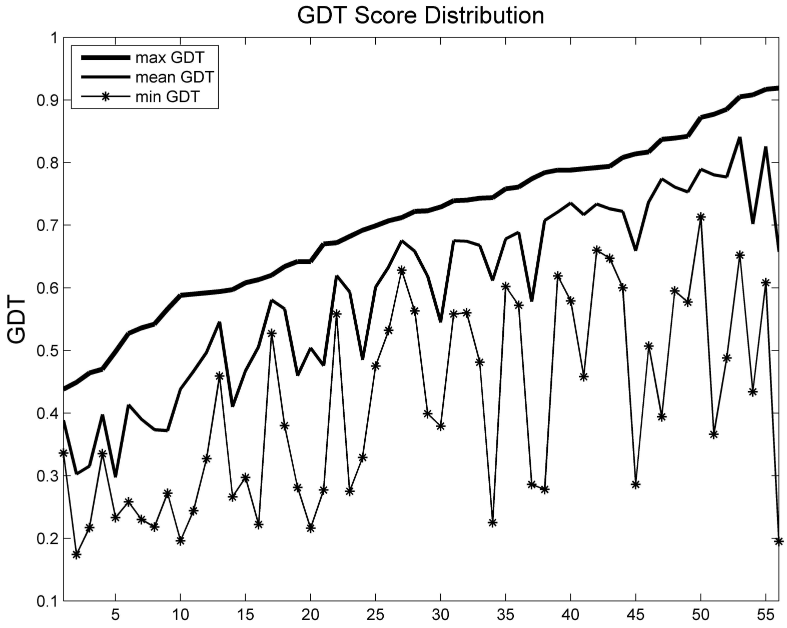

Figure 8.1: Decoy Distribution of Dataset 1

8.3 Benchmark Result

In the test, we compared the QA performance of the new score, i.e. CRFCom, to naive consensus GDT (CGDT) and three knowledge based scoring functions, namely, OPUS-CA, DFIRE and RW. Each score was used to rank the decoys of a given protein. We studied the selection performance using three measures. In the following comparison tables, "GDT1" is the average GDT score of top 1 model selected out by each QA method; "avgGDT5" is the average of the mean GDT score of top 5 models; and "Spearman" is the average Spearman correlation coefficient.

Table 8.1 and 8.2 compared the average performance of different QA methods on the two datasets respectively. Dataset 2 was not tested using OPUS-CA as the method failed to output scores for quite a number of decoys due to the structural

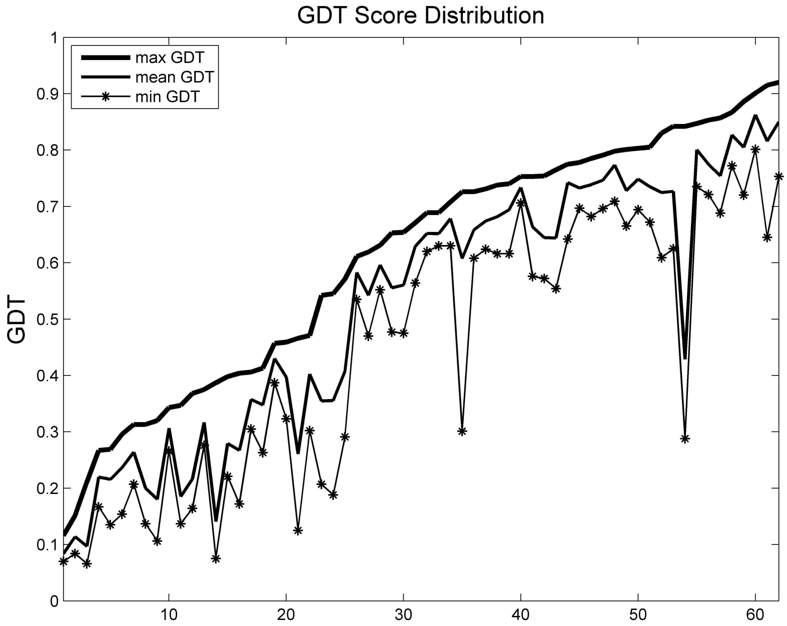

Figure 8.2: Decoy Distribution of Dataset 2

abnormality or too many steric conflicts in CASP models. From these two tables, we can see that the new method, i.e. CRFCom, achieved significantly better performance than knowledge based scoring functions such as OPUS-CA, RW and DFIRE. More specifically, for dataset 1, CRFCom has achieved top-1 selection 0.631, significantly better than that of OPUS-CA (0.614), which is the best one among the three scoring functions. On dataset 2, CRFCom has top-1 selection performance of 0.562, which is about 1 GDT point better than that of RW (0.554). Besides the better top-1 selection performance, CRFCom also has higher correlation to GDT score than OPUS-CA, DFIRE and RW on these two datasets.

Although CRFCom has better performance than the three knowledge based scoring functions, it has not achieved significant improvement over CGDT score in selection performance. And in terms of correlation, CRFCom is worse than that of

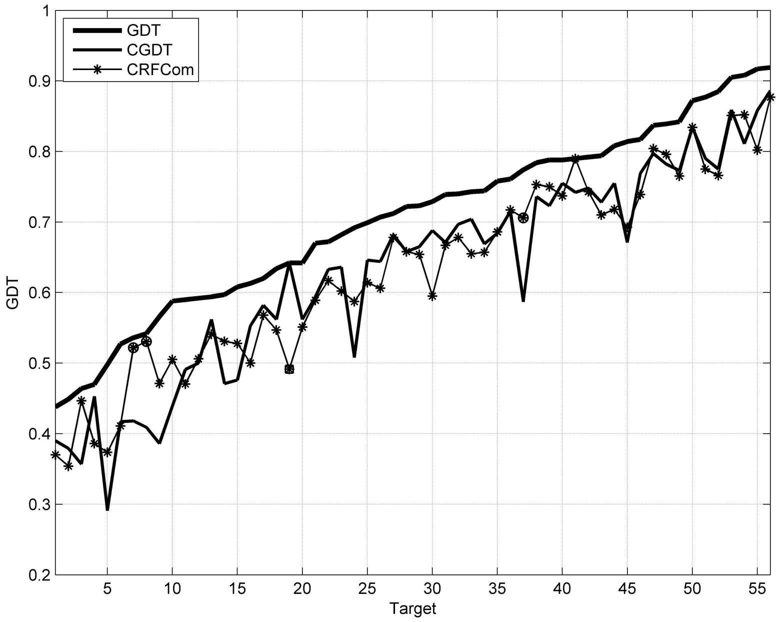

Figure 8.3: Detailed Comparison of Top-1 Selection between CGDT and CRFCom on Dataset 1. The points in circles are targets on which CRFCom has selected a much better model than that of CGDT, the vice are points shown in square boxes.

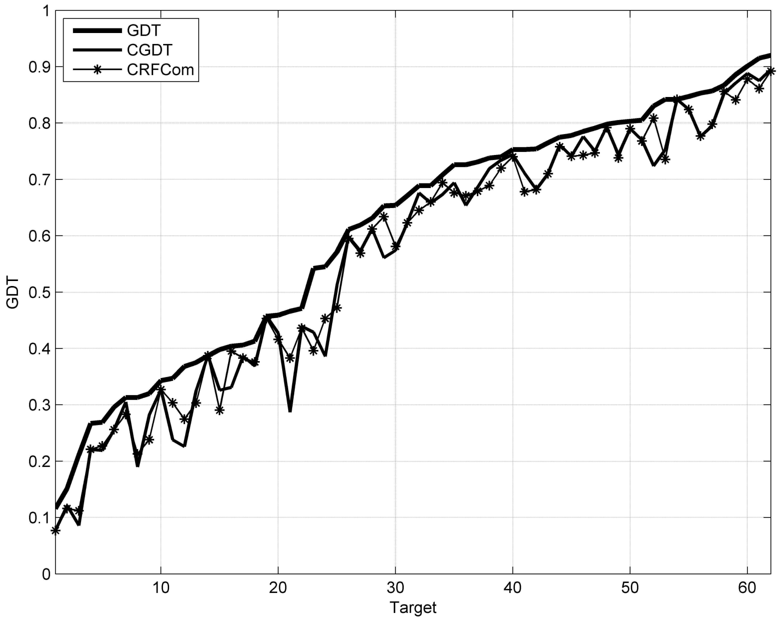

Figure 8.4: Detailed Comparison of Top-1 Selection between CGDT and CRFCom on Dataset 2

	Top1	**Best5**	**Mean5**	**Pearson**	**Spearman**
GDT	0.705	0.705	0.693	1.000	1.000
OPUSCA	0.614	0.646	0.613	0.322	0.237
DFIRE	0.609	0.641	0.608	0.312	0.231
RW	0.610	0.636	0.609	0.278	0.196
CGDT	0.629	0.641	0.625	0.540	0.407
CRFCom	0.631	0.641	0.614	0.420	0.308
HMM.Z	0.615	0.651	0.616	0.265	0.192

Table 8.1: QA performance on Dataset 1.

	Top1	**Best5**	**Mean5**	**Pearson**	**Spearman**
GDT	0.608	0.608	0.598	1.000	1.000
DFIRE	0.547	0.571	0.549	0.362	0.372
RW	0.554	0.577	0.553	0.282	0.315
CGDT	0.560	0.575	0.560	0.543	0.546
CRFCom	0.562	0.574	0.553	0.469	0.474
PWCom	0.564	0.576	0.562	0.566	0.580

Table 8.2: QA performance on Dataset 2.

CGDT, for example, on dataset 1, CGDT has Pearson correlation of 0.540, while CRFCom has only 0.420.

Figures 8.3 and 8.4 shows the detailed comparison between CRFCom and CGDT in top-1 selection performance. As we can see that for most targets from the two datasets, CRFCom has similar performance, except for the noticeable three points in Figure 8.3, in which CRFCom has selected out much better models than CGDT.

8.4 Analysis and Conclusion

The new method combines consensus GDT score with sequence and structural features to predict the real GDT score of decoys. One of the advantages of this method is that the output score is a approximation of the actual GDT score of a decoy, which directly indicates the distance of the decoy to its native structure. The scores produced by OPUS-CA, RW and DFIRE can only be used as a confidence, for which the absolute value does not has actual meaning. Another advantage of this method is that the score can be used for local quality assessment, as CRFCom score is the summation of the local QA score at each protein position.

Table 8.1 also compares the performance of CRFCom to the score of HMM.Z, which is described in Chapter 7. CRFCom has better performance than HMM.Z in both selection and correlation performance. Two reasons may account for this. The first reason for this is that conditional random field is more convenient to adopt complex and redundant features and thus more capable of learning the underlying intricate relationship between protein sequence and structures. The second reason is CGDT score, as one of major features of CRFCom, captures the global contact information of the decoy from the decoy set. This reason also accounts for the comparison in Table 8.2 between CRFCom and PWCom, which is described in Chapter 6. In this comparison, PWCom has achieved slightly better performance than CRFCom in terms of selection and correlation performance. PWCom combines CGDT with several powerful knowledge based score functions, most of which were build upon global contact potential within proteins. This also indicates that we need incorporate more global contact information in order to further improve the performance of this method. Although PWCom has achieved better performance than CRFCom, especially in correlation performance, CRFCom has its own advantages over PWCom. The first one is CRFCom is much faster than PWCom which does N^2 comparisons, where N is the number of decoys. The other advantages is that CRFCom can be used for local structural assessment.

Chapter 9

Conclusion and Future Work

9.1 Conclusions

In this dissertation, we have presented several methods developed for protein structure prediction and the tools have been applied to MUFOLD, which is platform for template based protein modeling.

The conditional random field based threading alignment tool is the state-of-art alignment method, which is more capable of combining the complex sequence and structural features from query sequence and template structure respectively, and helps build good alignment for proteins with less sequence similarity. The test result has demonstrated the improvement in alignment accuracy over HHSearch, which is based on profile hidden markov model. And this tool serves as a platform for continuous improvement in future development.

To ensure good alignment quality and fold recognition performance, sequence profile plays an important role. In order to improve the sequence profile quality, the proposed protocol based on Pfam database and PSI-BLAST achieved better fold recognition comparing to the default procedure of PSI-BLAST and HHSearch.

Protein structure quality assessment (QA) is one of the most important steps of protein structure prediction. In Chapters 5 and 6, we discussed two methods to combine naive consensus GDT score and knowledge based scoring functions in order to achieve better selection and correlation performance. More specifically, in Chapter 5, the scoring functions weighted by the optimal weights obtained by the quadratic programming achieved better decerning power than the original scoring functions. The pairwise comparison based method proposed in Chapter 6 avoided directly mapping the sequence and structure features to the actual GDT score of protein models and compared all pairs of involving structure candidates to determine their ranking in terms of structure quality.

In Chapter 7, a new Hidden Markov Model (HMM) based method was proposed to capture the sequence structure compatibility, which is the essence of knowledge based scoring functions. The HMM was designed to capture the complex relationship among sequence profile, secondary structures and three dimensional structure information, from which, the score can be used as a confidence measure to indicate the structural quality of single protein models.

In Chapter 8, we proposed a new conditional random field based method for structural quality assessment. This method combined consensus GDT information with sequence and structure features to predict the actual structural quality state at each protein position through the strong descriptive power of conditional random field. The score can be used for both global and local structure quality assessment.

9.2 Future Work

Although these methods have achieved improvement in benchmark tests, there are still large rooms to improve their performance. For remote homologous proteins, the alignment quality is still not enough. More specifically, The conditional random field based threading alignment tools can be further improved by adopting even more powerful machine learning methods and more informative sequence and structure features.

Sequence profile quality critically affects alignment accuracy and fold recognition performance. The Pfam based sequence profile generation protocol has better fold recognition performance than HHSearch default procedure, but the accuracy of top-1 alignment is worse than that of HHSearch default. The reason for this might be that HHSearch parameters are not optimal any more on the new sequence profiles. And the protocol itself also has a lot of aspects to tune.

Protein structure quality assessment methods, especially the methods based on hidden markov model and conditional random field are promising to be good scoring functions. As we discussed previously, global contact information is effective in discriminating near native structures from fair predictions. Introducing more informative features and better tuning the learning model could lead to further improvement to these methods.

Appendix A

CLE: A Structure Alphabet

A.1 Protein Structure Representation By Pseudo-bond Angles

For each protein structure, only Ca atom of residues are chosen as the representative points. In this representation, two adjacent residues in a protein sequence are virtually bonded, forming pseudo-bond.

The virtual bond bending angle θ defined for three contiguous points (a, b, c) is the angle between the vectors $r_{ab} = r_b - r_a$, i.e. $\theta = \frac{\mathbf{r}_{ab} \cdot \mathbf{r}_{bc}}{|\mathbf{r}_{ab}\mathbf{r}_{bc}|}$. The angle of θ is $[0, 2\pi]$. The virtual torsion angle τ is defined for four contiguous points (a, b, c, d) is the dihedral angle between the plans *abc* and *bcd*. The range of τ is $(-\pi, \pi]$. For four consecutive Ca atoms $(X_{k-2}, X_{k-1}, X_k, X_{k+1})$, three angles are defined $(\theta_k, \tau_k, \theta_{k+1})$, where θ_k is the bend angel of (X_{k-1}, X_k, X_{k+1}) and τ_k is the dihedral angle of $(X_{k-2}, X_{k-1}, X_k, X_{k+1})$ and θ_{k+1} is the bend angle of (X_{k-1}, X_k, X_{k+1}), as shown Fig. A.1.

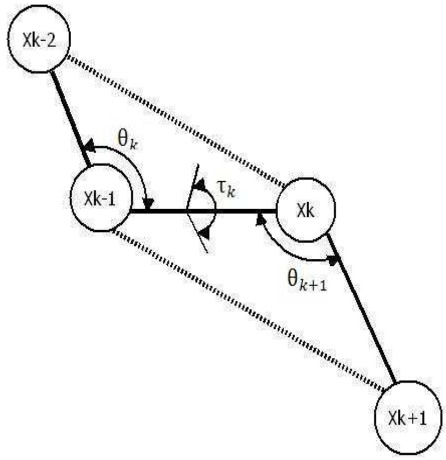

Figure A.1: Dihedral Angles of Four Points

Let the $x_k \equiv (\theta_k, \tau_k, \theta_{k+1})$, a protein of length L is represented a list of x_k, where k goes from 3 to $L-1$. So, total number of x_k is $L-3$.

A.2 Gaussian Mixture Model for the Angle Probability Distribution

The method clusters the angle-triplet of the four consecutive residues into 17 groups. The probability distribution of point x is given by a Gaussian mixture model

$$P(x|M) = \sum_{i=1}^{17} \pi_i N(u_i, \Sigma_i) \tag{A.1}$$

where $N(u, \Sigma)$ is the normal distribution.

$$N(u, \Sigma) = (2\pi)^{-3/2} |\Sigma|^{-1/2} e^{\frac{1}{2}(x-u)\cdot\Sigma^{-1}\cdot(x-u)} \tag{A.2}$$

The probability for a point to belong to the i-th category C_i according to the Bayes formula is

$$\begin{aligned} P(C_i|x) &\propto \pi_i P(x|C_i) \\ &\propto \pi_i |\Sigma_i|^{-\frac{1}{2}} \cdot e^{-\frac{1}{2}(x-u_i)'\Sigma_i^{-1}(x-u_i)} \end{aligned} \tag{A.3}$$

and,

$$i = \arg\max_i P(C_i|x) \tag{A.4}$$

The parameters of this model are trained using (Expectation-Maximization) EM algorithm, which are shown in Table A.1.

			$\mathbf{u}$			Σ^{-1}					
State	π	$\vert\Sigma\vert^{-1/2}$	θ	τ	θ'	$\theta\theta$	$\tau\theta$	$\tau\tau$	$\theta'\theta$	$\theta'\tau$	$\theta'\theta'$
I	8.20	1881	1.52	0.83	1.52	275.40	-28.30	84.30	106.90	-46.10	214.40
J	7.30	1797	1.58	1.05	1.55	314.30	-10.30	46.00	37.80	-70.00	332.80
H	16.20	10425	1.55	0.88	1.55	706.60	-93.90	245.50	128.90	-171.80	786.10
K	5.90	254	1.48	0.70	1.43	73.80	-13.70	21.50	15.50	-25.30	75.70
F	4.90	105	1.09	-2.72	0.91	24.10	1.90	10.90	-11.20	-8.80	53.00
E	11.60	109	1.02	-2.98	0.95	34.30	4.20	15.20	-9.30	-22.50	56.80
E	7.50	100	1.01	-1.88	1.14	28.00	4.10	6.20	2.30	-5.10	69.40
C	5.40	78	0.79	-2.30	1.03	56.20	3.80	4.20	-10.80	-2.10	30.10
D	4.30	203	1.02	-2.00	1.55	30.50	9.10	8.70	6.00	5.70	228.60
A	3.90	66	1.06	-2.94	1.34	26.90	4.60	4.90	9.50	-5.00	54.30
B	5.60	133	1.49	2.09	1.05	163.90	0.60	3.80	2.00	-3.70	32.30
G	5.30	40	1.40	0.75	0.84	43.70	2.50	1.40	-7.00	-2.90	34.50
L	3.70	144	1.47	1.64	1.44	72.90	2.10	4.80	1.90	-7.90	72.90
M	3.10	74	1.12	0.14	1.49	25.30	3.20	3.10	9.90	0.90	83.00
N	2.10	247	1.54	-1.89	1.48	170.80	-0.70	3.70	-4.10	3.10	98.70
P	3.20	206	1.24	-2.98	1.49	48.00	8.20	7.30	-4.90	-6.60	155.60
Q	1.70	25	0.86	-0.37	1.01	28.40	1.50	1.20	3.40	0.10	19.50

Table A.1: Parameters for CLE Model

Appendix B

Hidden Markov Model (HMM)

A hidden Markov model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. Its configuration λ is defined as follows:

- N: number of states
- L: number of observations
- $x_t, t = 0, 1, .., L-1$: the observation at position t, X be the corresponding vector
- $y_t, t = 0, 1, .., L-1$: state at position t, and Y be the corresponding vector
- $\pi_i = P(y_0 = i), i = 0, .., N-1$: initial probability of being state i
- $T_{ij} = P(y_{t+1} = j | y_t = i), i = 0, .., N-1, j = 0, .., N-1, t = 0, .., L-2$: transition probability satisfying $\sum_{j=0}^{N-1} T_{ij} = 1$

- $b_j(x_t), j = 0,..,N-1, t = 0,..,L-1$: emission probability of observing x_t at state j

We need to define the forward and backward variable for model inference.

Forward variable $a_t(i) = P(x_1,..,x_t, y_t = i|\lambda)$ is the joint probability of the partial observation $x_1,..,x_t$ and state y_t at position t.

1. $a_0(i) = \pi_i b_i(x_0), i = 0,..,N-1$
2. $a_{t+1}(j) = b_j(x_{t+1}) \sum_{i=0}^{N-1} a_t(i)T_{ij}, j = 0,..,N-1$

Backward variable $\beta_t(i) = P(x_{t+1},..,x_{L-1}|y_t = i, \lambda)$ is the the probability of observing the remaining observations $x_{t+1},..,x_{L-1}$ given any starting state $y_t = i$.

1. $\beta_{L-1}(i) = 1$
2. $\beta_t(i) = \sum_{j=0}^{N-1} \beta_{t+1}(j)T_{ij}b_j(x_{t+1})$

The sum of the probability of all possible paths is defined as

$$Z \equiv P(X|\lambda) = \sum_i a_t(i)\beta_t(i), \forall t \tag{B.1}$$

which does not depend on the position of t.

Gamma variable $\gamma_t(i) = P(y_t = i|X, \lambda)$ is defined as the probability of being in state $y_t = i$, given the observation sequence and the model λ. From the forward and backward variables, we have

$$\begin{aligned} \gamma_t(i) &= P(y_t = i|X, \lambda) \\ &= \frac{a_t(i)\beta_t(i)}{Z} \end{aligned} \tag{B.2}$$

Let $\xi_t(i,j)$ be the probability of being $y_t = i$ and $y_{t+1} = j$, given the model and the observation sequence.

$$\begin{aligned}\xi_t(i,j) &= P(y_t = i, y_{t+1} = j | X, \lambda) \\ &= \frac{a_t(i) T_{ij} \beta_{t+1}(j) b_j(x_{t+1})}{Z}\end{aligned} \tag{B.3}$$

It is easy to see that

$$\gamma_t(j) = \sum_{j=1}^{N} \xi_t(i,j) \tag{B.4}$$

Appendix C

Expectation-Maximization (EM) Algorithm

EM is a general method of finding the Maximum Likelihood Estimation (MLE) estimate of parameters of an underlying distribution with hidden variables or missing data. Let's define the following three variables

- x: the observed data
- y: the hidden variable
- θ: the distribution to estimate.

C.1 Derivation Of EM Algorithm

According to the Bayes principle, we have

$$logP(x|\theta) = logP(x, y|\theta) - logP(y|x, \theta) \tag{C.1}$$

where $P(x, y|\theta)$ is the likelihood function $L(\theta; x, y)$. Multiply the equation by $P(y|x, \theta^t)$ and sum up over all possible y,

$$logP(x|\theta) = \sum_y P(y|x, \theta^t)logP(x, y|\theta) - \sum_y P(y|x, \theta^t)logP(y|x, \theta) \tag{C.2}$$

where θ^t is the current available model. Let

$$Q(\theta|\theta^t) = \sum_y P(y|x, \theta^t)logP(x, y|\theta) \tag{C.3}$$

In order for $logP(x|\theta)$ to be larger than $logP(x|\theta^t)$, we only need try to let the following equation greater than zero.

$$logP(x|\theta) - logP(x|\theta^t) = Q(\theta|\theta^t) - Q(\theta^t|\theta^t) + \sum_y P(y|x, \theta^t)log\frac{P(y|x, \theta^t)}{P(y|x, \theta)} \tag{C.4}$$

The last term is the relative entropy and it is always non-negative. So

$$logP(x|\theta) - logP(x|\theta^t) \geq Q(\theta|\theta^t) - Q(\theta^t|\theta^t) \tag{C.5}$$

So, to maximize $logP(x|\theta)$, we only need to choose θ to maximize $Q(\theta|\theta^t)$.

C.2 EM for HMM

In HMM, we want to maximize the likelihood

$$logP(x|\theta) = \sum_y logP(x, y|\theta))$$

where y is path states, which is hidden. x, y are vectors of the same length of the number of observations. In the following, we use t as the position index in sequence and θ' to denote the previous model.

$$\begin{aligned}
Q(\theta|\theta') &= \sum_y P(y|x,\theta') log P(y,x|\theta) \\
&= \sum_y P(y|x,\theta') \times log\left[\pi_{y_0} \prod_{t=1}^{T} T_{y_{t-1}y_t} \prod_{t=1}^{T} b_{y_t}(x_t)\right] \\
&= \sum_y P(y|x,\theta') \times \left[log(\pi_{y_0}) + \sum_{t=1}^{T} log(T_{y_{t-1}y_t}) + \sum_{t=1}^{T} log[b_{y_t}(x_t)]\right] \\
&= \sum_i \gamma_0(i) log(\pi_i) + \sum_i \sum_j \sum_t \xi_t(i,j) log(T_{ij}) + \sum_y P(y|x,\theta^t) \sum_{t=1}^{T} log[b_{y_t}(x_t)] \\
&\equiv Q_\pi + Q_t + Q_e \qquad \text{(C.6)}
\end{aligned}$$

$P(y|x,\theta')$ is just a constant weight when y is fixed. Q_π, Q_t are easy to maximize using Lagrange multiplies with corresponding constraints. Q_e is sometimes difficulty to optimize, depending on the function of $b_{y_t}(x_t)$.

$$Q_\pi = \sum_i \gamma_0(i) log(\pi_i) + \lambda(\sum_i \pi_i - 1) \qquad \text{(C.7)}$$

$$\frac{\partial Q_\pi}{\partial \pi_i} = \frac{1}{\pi_i}\gamma_0(i) + \lambda = 0 \qquad \text{(C.8)}$$

Summing over i, we have $\sum_i \gamma_0(i) + \lambda \sum_i \pi_i = 0$. So, we have $\lambda = -1$ which means that

$$\pi_i = \gamma_0(i) \qquad \text{(C.9)}$$

For Q_t, we have

$$Q_t = \sum_i \sum_j \sum_t \left[\xi_t(i,j) log(T_{ij}) \right] + \sum_i \left[\lambda_i (\sum_j T_{ij} - 1) \right] \tag{C.10}$$

Let

$$Q_t^i = \sum_j \sum_t \left[\xi_t(i,j) log(T_{ij}) \right] + \lambda_i (\sum_j T_{ij} - 1) \tag{C.11}$$

$$\frac{\partial Q_t^i}{\partial T_{ij}} = \sum_t \xi_t(i,j) \frac{1}{T_{ij}} + \lambda_i = 0 \tag{C.12}$$

Summing over j, we have

$$\sum_t \sum_j \xi_t(i,j) = -\lambda_i \sum_j T_{ij} \tag{C.13}$$

which gives

$$\sum_t \gamma_t(i) = -\lambda_i \tag{C.14}$$

Substitute λ_i into Eqn. C.12, we have

$$T_{ij} = \frac{\sum_t \xi_t(i,j)}{\sum_t \gamma_t(i)} \tag{C.15}$$

The optimization of Q_e depends on the the emission function $b_{y_t}(x_t)$. Usually the emission function needs to be careful design so that the optimization is feasible.

Bibliography

[1] J. L. Sussman, D. Lin, J. Jiang, N. O. Manning, J. Prilusky, O. Ritter, and E. E. Abola. Protein data bank (pdb): database of three-dimensional structural information of biological macromolecules. *Acta Crystallogr D Biol Crystallogr*, 54(Pt 6 Pt 1):1078–84, 1998.

[2] D. Petrey and B. Honig. Protein structure prediction: inroads to biology. *Mol Cell*, 20(6):811–9, 2005.

[3] D. Cozzetto and A. Tramontano. Advances and pitfalls in protein structure prediction. *Curr Protein Pept Sci*, 9(6):567–77, 2008.

[4] F. S. Domingues, W. A. Koppensteiner, and M. J. Sippl. The role of protein structure in genomics. *FEBS Lett*, 476(1-2):98–102, 2000.

[5] D. Baker and A. Sali. Protein structure prediction and structural genomics. *Science*, 294(5540):93–6, 2001.

[6] D. E. Kim, D. Chivian, and D. Baker. Protein structure prediction and analysis using the robetta server. *Nucleic Acids Res*, 32(Web Server issue):W526–31, 2004.

[7] S. Wu, J. Skolnick, and Y. Zhang. Ab initio modeling of small proteins by iterative tasser simulations. *BMC Biol*, 5:17, 2007.

[8] A. Roy, A. Kucukural, and Y. Zhang. I-tasser: a unified platform for automated protein structure and function prediction. *Nat Protoc*, 5(4):725–38, 2010.

[9] J. Zhang, Q. Wang, B. Barz, Z. He, I. Kosztin, Y. Shang, and D. Xu. Mufold: A new solution for protein 3d structure prediction. *Proteins*, 78(5):1137–52, 2010.

[10] A. Sali and T. L. Blundell. Comparative protein modelling by satisfaction of spatial restraints. *J Mol Biol*, 234(3):779–815, 1993.

[11] K. T. Simons, R. Bonneau, I. Ruczinski, and D. Baker. Ab initio protein structure prediction of casp iii targets using rosetta. *Proteins*, Suppl 3:171–6, 1999.

[12] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Res*, 25(17):3389–402, 1997.

[13] J. Soding. Protein homology detection by hmm-hmm comparison. *Bioinformatics*, 21(7):951–60, 2005.

[14] J. Peng and J. Xu. Boosting protein threading accuracy. *Res Comput Mol Biol*, 5541:31–45, 2009.

[15] J. Peng and J. Xu. Boosting protein threading accuracy. *Res Comput Mol Biol*, 5541:31–45, 2009.

[16] R. D. Finn, J. Mistry, J. Tate, P. Coggill, A. Heger, J. E. Pollington, O. L. Gavin, P. Gunasekaran, G. Ceric, K. Forslund, L. Holm, E. L. Sonnhammer,

S. R. Eddy, and A. Bateman. The pfam protein families database. *Nucleic Acids Res*, 38(Database issue):D211–22, 2010.

[17] Y. Wu, M. Lu, M. Chen, J. Li, and J. Ma. Opus-ca: a knowledge-based potential function requiring only calpha positions. *Protein Sci*, 16(7):1449–63, 2007.

[18] H. Zhou and Y. Zhou. Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction. *Protein Sci*, 11(11):2714–26, 2002.

[19] J. Moult, J. T. Pedersen, R. Judson, and K. Fidelis. A large-scale experiment to assess protein structure prediction methods. *Proteins*, 23(3):ii–v, 1995.

[20] A. Zemla. Lga: A method for finding 3d similarities in protein structures. *Nucleic Acids Res*, 31(13):3370–4, 2003.

[21] G. Yona and M. Levitt. Within the twilight zone: a sensitive profile-profile comparison tool based on information theory. *J Mol Biol*, 315(5):1257–75, 2002.

[22] B. Rost. Twilight zone of protein sequence alignments. *Protein Eng*, 12(2):85–94, 1999.

[23] M. A. Marti-Renom, M. S. Madhusudhan, and A. Sali. Alignment of protein sequences by their profiles. *Protein Sci*, 13(4):1071–87, 2004.

[24] L. Rychlewski, L. Jaroszewski, W. Li, and A. Godzik. Comparison of sequence profiles. strategies for structural predictions using sequence information. *Protein Sci*, 9(2):232–41, 2000.

[25] Y. Xu, D. Xu, and E. C. Uberbacher. An efficient computational method for globally optimal threading. *J Comput Biol*, 5(3):597–614, 1998.

[26] Y. Xu and D. Xu. Protein threading using prospect: design and evaluation. *Proteins*, 40(3):343–54, 2000.

[27] J. Xu, M. Li, G. Lin, D. Kim, and Y. Xu. Protein threading by linear programming. *Pac Symp Biocomput*, pages 264–75, 2003.

[28] J. Xu, M. Li, D. Kim, and Y. Xu. Raptor: optimal protein threading by linear programming. *J Bioinform Comput Biol*, 1(1):95–117, 2003.

[29] J. Xu, F. Jiao, and L. Yu. Protein structure prediction using threading. *Methods Mol Biol*, 413:91–121, 2008.

[30] H. Zhou and Y. Zhou. Single-body residue-level knowledge-based energy score combined with sequence-profile and secondary structure information for fold recognition. *Proteins*, 55(4):1005–13, 2004.

[31] Jiye Shia, Tom L Blundella, and Kenji Mizuguchi. Fugue: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. *Journal of Molecular Biology*, 310(1):243–257, 2001.

[32] D. T. Jones. Genthreader: an efficient and reliable protein fold recognition method for genomic sequences. *J Mol Biol*, 287(4):797–815, 1999.

[33] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: probabilistic models for segmenting and labeling sequence data. *International Conference on Machine Learning*, 2001.

[34] Chuong B. Do, Samuel S. Gross, and Serafim Batzoglou. Contralign: discriminative training for protein sequence alignment. *RECOMB'06 Proceedings of the 10th annual international conference on Research in Computational Molecular Biology*, pages 160–174, 2006.

[35] J. Peng and J. Xu. Low-homology protein threading. *Bioinformatics*, 26(12):i294–300, 2010.

[36] Thomas G. Dietterich, Adam Ashenfelter, and Yaroslav Bulatov. Training conditional random fields via gradient tree boosting. *Proceedings of the 21st International Conference on Machine Learning (ICML)*, pages 217–224, 2004.

[37] L. J. McGuffin, K. Bryson, and D. T. Jones. The psipred protein structure prediction server. *Bioinformatics*, 16(4):404–5, 2000.

[38] K. Bryson, L. J. McGuffin, R. L. Marsden, J. J. Ward, J. S. Sodhi, and D. T. Jones. Protein structure prediction servers at university college london. *Nucleic Acids Res*, 33(Web Server issue):W36–8, 2005.

[39] W. Kabsch and C. Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–637, 1983.

[40] J. Cheng, A. Z. Randall, M. J. Sweredoski, and P. Baldi. Scratch: a protein structure and structural feature prediction server. *Nucleic Acids Res*, 33(Web Server issue):W72–6, 2005.

[41] 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.

[42] I. N. Shindyalov and P. E. Bourne. Protein structure alignment by incremental combinatorial extension (ce) of the optimal path. *Protein Eng*, 11(9):739–47, 1998.

[43] P. Bork and E. V. Koonin. Predicting functions from protein sequences–where are the bottlenecks? *Nat Genet*, 18(4):313–8, 1998.

[44] M. M. Lee, M. K. Chan, and R. Bundschuh. Simple is beautiful: a straightforward approach to improve the delineation of true and false positives in psi-blast searches. *Bioinformatics*, 24(11):1339–43, 2008.

[45] M. W. Gonzalez and W. R. Pearson. Homologous over-extension: a challenge for iterative similarity searches. *Nucleic Acids Res*, 38(7):2177–89, 2010.

[46] B. H. Kim, Q. Cong, and N. V. Grishin. Hangout: generating clean psi-blast profiles for domains with long insertions. *Bioinformatics*, 26(12):1564–5, 2010.

[47] G. M. Boratyn, A. A. Schaffer, R. Agarwala, S. F. Altschul, D. J. Lipman, and T. L. Madden. Domain enhanced lookup time accelerated blast. *Biol Direct*, 7:12, 2012.

[48] A. Marchler-Bauer, S. Lu, J. B. Anderson, F. Chitsaz, M. K. Derbyshire, C. DeWeese-Scott, J. H. Fong, L. Y. Geer, R. C. Geer, N. R. Gonzales, M. Gwadz, D. I. Hurwitz, J. D. Jackson, Z. Ke, C. J. Lanczycki, F. Lu, G. H. Marchler, M. Mullokandov, M. V. Omelchenko, C. L. Robertson, J. S. Song, N. Thanki, R. A. Yamashita, D. Zhang, N. Zhang, C. Zheng, and S. H. Bryant. Cdd: a conserved domain database for the functional annotation of proteins. *Nucleic Acids Res*, 39(Database issue):D225–9, 2011.

[49] W. Li and A. Godzik. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–9, 2006.

[50] Y. Zhang and J. Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins*, 57(4):702–10, 2004.

[51] Q. Dong, X. Wang, and L. Lin. Novel knowledge-based mean force potential at the profile level. *BMC Bioinformatics*, 7:324, 2006.

[52] Q. Dong and S. Zhou. Novel nonlinear knowledge-based mean force potentials based on machine learning. *IEEE/ACM Trans Comput Biol Bioinform*, 8(2):476–86, 2011.

[53] D. W. Gatchell, S. Dennis, and S. Vajda. Discrimination of near-native protein structures from misfolded models by empirical free energy functions. *Proteins*, 41(4):518–34, 2000.

[54] T. Lazaridis and M. Karplus. Discrimination of the native from misfolded protein models with an energy function including implicit solvation. *J Mol Biol*, 288(3):477–87, 1999.

[55] F. Melo and E. Feytmans. Novel knowledge-based mean force potential at atomic level. *J Mol Biol*, 267(1):207–22, 1997.

[56] F. Melo and E. Feytmans. Assessing protein structures with a non-local atomic interaction energy. *J Mol Biol*, 277(5):1141–52, 1998.

[57] D. Petrey and B. Honig. Free energy determinants of tertiary structure and the evaluation of protein models. *Protein Sci*, 9(11):2181–91, 2000.

[58] A. W. Stumpff-Kane and M. Feig. A correlation-based method for the enhancement of scoring functions on funnel-shaped energy landscapes. *Proteins*, 63(1):155–64, 2006.

[59] S. Miyazawa and R. L. Jernigan. How effective for fold recognition is a potential of mean force that includes relative orientations between contacting residues in proteins? *J Chem Phys*, 122(2):024901, 2005.

[60] Y. Yang and Y. Zhou. Specific interactions for ab initio folding of protein terminal regions with secondary structures. *Proteins*, 72(2):793–803, 2008.

[61] J. Zhang and Y. 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.

[62] P. Benkert, S. C. Tosatto, and T. Schwede. Global and local model quality estimation at casp8 using the scoring functions qmean and qmeanclust. *Proteins*, 77 Suppl 9:173–80, 2009.

[63] P. Benkert, T. Schwede, and S. C. Tosatto. Qmeanclust: estimation of protein model quality by combining a composite scoring function with structural density information. *BMC Struct Biol*, 9:35, 2009.

[64] A. J. Martin, C. Mirabello, and G. Pollastri. Neural network pairwise interaction fields for protein model quality assessment and ab initio protein folding. *Curr Protein Pept Sci*, 12(6):549–62, 2011.

[65] Z. Wang, A. N. Tegge, and J. Cheng. Evaluating the absolute quality of a single protein model using structural features and support vector machines. *Proteins*, 75(3):638–47, 2009.

[66] J. Cheng, Z. Wang, A. N. Tegge, and J. Eickholt. Prediction of global and local quality of casp8 models by multicom series. *Proteins*, 77 Suppl 9:181–4, 2009.

[67] Z. Wang, J. Eickholt, and J. Cheng. Multicom: a multi-level combination approach to protein structure prediction and its assessments in casp8. *Bioinformatics*, 26(7):882–8, 2010.

[68] Q. Wang, K. Vantasin, D. Xu, and Y. Shang. Mufold-wqa: A new selective consensus method for quality assessment in protein structure prediction. *Proteins*, 79 Suppl 10:185–95, 2011.

[69] B. Wallner and A. Elofsson. Prediction of global and local model quality in casp7 using pcons and proq. *Proteins*, 69 Suppl 8:184–93, 2007.

[70] P. Larsson, M. J. Skwark, B. Wallner, and A. Elofsson. Assessment of global and local model quality in casp8 using pcons and proq. *Proteins*, 77 Suppl 9:167–72, 2009.

[71] J. Qiu, W. Sheffler, D. Baker, and W. S. Noble. Ranking predicted protein structures with support vector regression. *Proteins*, 71(3):1175–82, 2008.

[72] Zhiquan. He, Jingfen. Zhang, Yang. Xu, Yi. Shang, and Dong Xu. Protein structural model selection based on protein-dependent scoring function. *Statistics and Its Interface*, Volume 0, 2011.

[73] L. J. McGuffin. The modfold server for the quality assessment of protein structural models. *Bioinformatics*, 24(4):586–7, 2008.

[74] L. J. McGuffin. Prediction of global and local model quality in casp8 using the modfold server. *Proteins*, 77 Suppl 9:185–90, 2009.

[75] L. J. McGuffin and D. B. Roche. Automated tertiary structure prediction with accurate local model quality assessment using the intfold-ts method. *Proteins*, 79 Suppl 10:137–46, 2011.

[76] Z. Wang, J. Eickholt, and J. Cheng. Apollo: a quality assessment service for single and multiple protein models. *Bioinformatics*, 27(12):1715–6, 2011.

[77] M. Lu, A. D. Dousis, and J. Ma. Opus-psp: an orientation-dependent statistical all-atom potential derived from side-chain packing. *J Mol Biol*, 376(1):288–301, 2008.

[78] Weimou. Zheng and Xin Liu. A protein structural alphabet and its substitution matrix clesum. *Transactions on Computational Systems Biology II*, Volume 3680:59–67, 2005.

[79] E. Faraggi, Y. Yang, S. Zhang, and Y. Zhou. Predicting continuous local structure and the effect of its substitution for secondary structure in fragment-free protein structure prediction. *Structure*, 17(11):1515–27, 2009.

[80] Y. Xu, D. Xu, and E. C. Uberbacher. An efficient computational method for globally optimal threading. *J Comput Biol*, 5(3):597–614, 1998.

[81] Y. Xu and D. Xu. Protein threading using prospect: design and evaluation. *Proteins*, 40(3):343–54, 2000.

[82] K. Lin, A. C. May, and W. R. Taylor. Threading using neural network (tune): the measure of protein sequence-structure compatibility. *Bioinformatics*, 18(10):1350–7, 2002.

[83] J. U. Bowie, K. Zhang, M. Wilmanns, and D. Eisenberg. Three-dimensional profiles for measuring compatibility of amino acid sequence with three-dimensional structure. *Methods Enzymol*, 266:598–616, 1996.

[84] S. Sunyaev, E. Kuznetsov, I. Rodchenkov, and V. Tumanyan. Protein sequence-structure compatibility criteria in terms of statistical hypothesis testing. *Protein Eng*, 10(6):635–46, 1997.

[85] H. Sumikawa, K. Fukuhara, E. Suzuki, Y. Matsuo, and K. Nishikawa. Tertiary structural models for human interleukin-6 and evaluation by a sequence-structure compatibility method and nmr experimental information. *FEBS Lett*, 404(2-3):234–40, 1997.

[86] Y. Matsuo and K. Nishikawa. Protein structural similarities predicted by a sequence-structure compatibility method. *Protein Sci*, 3(11):2055–63, 1994.

[87] A. C. Camproux and P. Tuffery. Hidden markov model-derived structural alphabet for proteins: the learning of protein local shapes captures sequence specificity. *Biochim Biophys Acta*, 1724(3):394–403, 2005.

[88] A. C. Camproux, P. Tuffery, J. P. Chevrolat, J. F. Boisvieux, and S. Hazout. Hidden markov model approach for identifying the modular framework of the protein backbone. *Protein Eng*, 12(12):1063–73, 1999.

[89] C. Bystroff, V. Thorsson, and D. Baker. Hmmstr: a hidden markov model for local sequence-structure correlations in proteins. *J Mol Biol*, 301(1):173–90, 2000.

[90] Ram Samudrala and Michael Levitt. Decoys r us: A database of incorrect conformations to improve protein structure prediction. *Protein Science*, 9(7):1399–1401, 2000.

VITA

Zhiquan He is a Ph.D. student in the Department of Computer Science at University of Missouri. He received his Master degree in Communication and Information System from Institute of Electronics, Chinese Academy of Sciences, P. R. China and his Bachelor degree in Electrical Engineering from Xiangtan University, P. R. China. His research interests include algorithm design, artificial intelligence, machine learning, and their applications in the field of bioinformatics, especially in protein structure prediction. He was a recipient of the Shumaker Fellowship for the year of 2011.