NEW CONSENSUS-BASED ALGORITHMS FOR QUALITY ASSESSMENT IN PROTEIN STRUCTURE PREDICTION

A Thesis

presented to

the Faculty of the Graduate School

University of Missouri

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

---

by

KITTINUN VANTASIN

Dr. Yi Shang, Thesis Supervisor

DECEMBER 2010

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

NEW CONSENSUS-BASED ALGORITHMS FOR QUALITY ASSESSMENT IN PROTEIN STRUCTURE PREDICTION

presented by Kittinun Vantasin,

a candidate for the degree of master of Computer Science and hereby certify that in their opinions it is worth acceptance.

_____

Dr. Yi Shang

_____

Dr. Dong Xu

_____

Dr. Ioan Kosztin

ACKNOWLEDGMENTS

First of all, I would like to wholeheartedly thank Dr. Shang, my thesis supervisor, for opportunity that was offered to me to be a part of his research. Without his invaluable and consistent instructions, I would not be able to make this thesis possible. Also, I would like to thank Qingguo Wang, my research-mate, who always co-operate with me on the project from the beginning to the end.

Furthermore, I am glad to be a member of MU-FOLD protein project and I would like to thank Dr. Dong Xu for his suggestion and he also organizes protein project meeting which is an immense source of knowledge and dicussion in this research area. I would like to thank Dr. Ioan Kosztin and everyone in the protein project who show their contributions to the MU-FOLD team.

In addition, I like to thank my lovely family for financial aids and their mental boosts. They have been working hard in order to make my education possible and they have been sending their supports via phone from Thailand every single week throughout my time in US.

I also want to thank my girlfriend, Yutika, for continuous both physical and mental encouragements throughout the time that we met. She has been a great source of power which is indispensable.

... and thank so much for my beloved God who is the ultimate creator and is above and beyond everything.

New Consensus-Based Algorithms For Quality Assessment

In Protein Structure Prediction

Kittinun Vantasin

Dr. Yi Shang, Thesis Supervisor

ABSTRACT

Two of the essential tasks in protein tertiary structure prediction are predicting quality and selecting the best quality model from given model structures. Finding solutions to these problems are fundamental to understanding the nature of proteins and advancing in protein research area. In this thesis, we present efficient algorithms that tackle both problems effectively. The algorithms are developed on the well-known consensus-based idea that has been continuously succesful since CASP6. For assessing the quality of structures, we develop several new methods based on the idea of removing redundant structures and outliers. The algorithms aims at finding suitable reference sets in computing the consensus-score in order to improve the existing algorithms. The methods can use any suitable pair-wise similarity measurement between a pair of models such as GDT-TS and Q score. We also develop a very efficient method for computing Q score for large size problem. In our experimental results, the algorithms are applied to CASP8 dataset and have achieved the superior performance over existing state-of-the-art methods including the top1 method in the QA category of CASP8. For the selecting the best model structure, our new methods are effective and perform better than other

best-performing scoring functions by upto 7.6% based on the actual GDT-TS of top1 selected model to the native structure. The selection result is obtained by our method using Q score are slightly worse than those obtained using GDT-TS, but using pair-wise Q score method is in general about 15 times faster than using pair-wise GDT-TS method.

List of Tables

List of Figures

Contents

Chapter 1

Introduction

Protein structure prediction can be considered as one of the most challenging goals quested by bioinformaticians. Major advantage of understanding protein stucture inevitably benefits the advance in field of medicine and biotechnology. With these importances, the growth of protein structure database is skyrocketing. However, only few of those protein structures have been solved experimentally due to the difficulties of experimental approaches. These approaches, for instance X-ray crystallography or NMR sprctroscopy, are very time-consuming and expensive to accomplish. Fortuitously, computational methods for protein stucture prediction have been raised to attention. They have become more and more predominant and a main interest of many protein research groups around the world during past recent years [1]. To help advance of computational methods, a bi-annual community-wide experiment for protein structure prediction named CASP (Critical Assessment of Techniques for Protein Structure Prediction) started on 1994

has been held. CASP experiment is served as a place for many research groups to apply their techniques to predict an amino acid sequences for which native structure is soon to be known [2].

Thus, ability to assess quality of protein structures that are predicted by computational approaches is inevitably important. To bolster its importance, Quality Assessment category in CASP was officially created in year of 2006 during CASP7.

Moreover, another important aspect in this area is protein structure selection problem. Occasionally, protein structure prediction tools generate a large number of structures with good candidates included but are inconsistent on selecting good ones. As the result, finding reliable methods of choosing high quality protein structure out of large number of quality protein structures is essential in protein structure prediction.

In this thesis, we propose new algorithms, SigW algorithm, StepW algorithm and RectW algorithm for protein structure quality assessment problem and protein structure selection problem.

The new algorithms for quality assessment problem presented are based on the consensus-based approaches which were proven their successes on recent CASP datasets [3] and [4]. Our algorithms use the strength of consensus-based over an "appropriate" reference set. RefSelect has shown that remove redundancy strategy can produce satisfied results over the simple consensus-based algortihm, RefAll [5]. Although ideas are similar, our implementations are different. Comparison with the best performing algorithms on CASP datset evidence that our algorithms

are consistently better than any other algorithms in terms of correlation results on CASP8 dataset. Not only our new algorithms have superior performance in terms of correlation results, but also they can select better protein structures on the same dataset.

Furthermore, this study shows a series of algorithms, QRefAll algorithm, AVGQRefAll algorithm and IRankQRefAll algorithm that base on consensus-based approach, with different protein structure similarity score named Q score [6]. With Q score consensus-based algorithm, we can choose a better protein structure quality than any other state-of-the-art scoring functions can in terms of true GDT-TS score of the top 1 ranked structure.

Organization of this thesis is structured as follows. In chapter 2, we define protein structure quality assessment problem and protein structure selection problem. Also, we review previous works by others on the same area, picking the most well-known and successful methods. In chapter 3, we extensively explain our proposed algortihms for protein structure quality assessment problem and protein structure selection problem. In chapter 4, we describe our plan of experiment, choice of suitable parameters and results on CASP8 dataset. Lastly, conclusion and future work are discussed in chapter 5.

Chapter 2

Protein Structure Prediction Quality Assessment and Selection Problem Formulation and Existing Algorithms

We define the protein structure prediction quality assessment (QA) problem in this chapter. As we discussed in the earlier section, we seperate this section into two different problems which are firstly, protein structure prediction QA problem and secondly, protein structure selection problem. Therefore, in this chapter, we define both problems mathematically.

2.1 Problem Formulation

2.1.1 Protein Structure Prediction QA Problem

Let N be the native protein structure corresponding to a target. Let p be a number of potential stuctures that were predicted by different prediction servers. Let S be a set of such predicted structures for each target. Therefore, we have set S, where $S = \{s_i, 1 \leq i \leq p\}$. Then, for each structure $s_i \in S$, we can use any algorithm to predict a score $X = \{x_i, 0 \leq x_i \leq 1, 1 \leq i \leq p\}$ by not applying any information of N.

The true evaluation of each candidate to the native stucture is quantitatively assessed by its GDT-TS (Global Distance Test Total Score) to the native structure which is denoted by set $Y = \{y_i, 1 \leq i \leq p\}$. GDT-TS metric will be explained in detail in next chapter about protein structure similarity score. The reason why we use GDT-TS as an evaluation measure because it has been using by CASP evaluator as a judging criterion since CASP5 in 2002 [7].

To measure the performance of predicted score, we use a correlation coeffi-cient ρ between $X = \{x_i\}$ and $Y = \{y_i\}$:

$$\rho = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{2.1}$$

The perfect correlation is 1. Thus, predicted score X that perfectly correlate with true GDT-TS to the native structure Y will give correlation score of $\rho = 1$.

Then, we can define X as the following:

Definition 1 *X is the **predicted score** for set S if :*

$$\arg\max_{x} \rho$$

Our objective is to generate predicted score X that is maximized the correlation between a set of predicted quality score X and a set of true quality values Y.

2.1.2 Protein Structure Selection Problem

We also use the same set of variables as we discussed in previous problem.

Definition 2 *s_i is chosen as **top selection protein structure** $\in S$ such that :*

$$y_i = max(Y)$$

Our goal is to pick s_i that is closest to the true top structure based on GDT-TS score of such predicted structure which respect to the native structure.

2.2 Existing Algorithms for Protein Structure Prediction QA and Selection

2.2.1 Energy or Scoring Functions

A number of approaches are used for QA in protein structure prediction, but one of the major idea is to utilize energy or scoring functions. These scoring functions which can be categorized into two main groups are served as predictors of the quality of given protein structures. The first category of scoring functions is physics-based energy functions [8] [9] and the second category is knowledge-based statistical scoring functions [10] [11].

Physics-based energy functions is constructed by applying experimental knowledge which is backed up by physical properties in molecule level. Such knowledge is called energy expression which is made up by components i.e. atomic representation and functional forms. The advantage of physics-based energy functions is that we can evaluate the actual energy based on internal coordination of amino atoms in protein tertiary structure. However, some of drawbacks are first, the calculation is very complicated to perform because many sources of information are required for instance, atomic description of protein structure and solvent, second, it is very computationally expensive and time-consuming for protein folding [12].

Knowledge-based energy statistical scoring functions is, on the other hand, built by using statistical knowledge of experimentally known protein stuctures which can discover misfolded proteins. Generally speaking, knowledge-based

scoring functions are successful in identifying poor-quality protein structure [13]. Another reward of using knowledge-based scoring functions is that they are faster, simpler and more accurate in term of performance than physics-based ones [14]. Nonetheless, knowledge-based scoring functions in some cases contain a lot of noises because of their statistical properties. Thus, sometimes they fail to identify correctly-folded protein structures. Even though they are much simpler than physics-based ones, they still have very complex structures and need many sources of information involved in the calculation. Additionally, due to the fluctuation of protein folding, protein quality prediction by using these scoring functions can be unreliable. As the result of these limitations, we cannot solely depend on these scoring functions.

Some of the most popular knowledge-based energy scoring functions are OPUS-PSP [15], OPUS-Ca [16], DFIRE [17], RAPDF [18] which will be described more in detail in our result section.

2.2.2 Machine Learning-based Algorithms

Next idea that is used to tackle protein quality assessment problem is machine learning-based approach. Famous techniques that fall in this category are, for example, support vector machine method and clustering-based method.

For support vector machine, it needs a learning process. In some cases, both individual and consensus-based features are extracted from training set [19]. Also, structural feature extracted from 3D coordinates of a model and 1D and 2D structural features predictors are included [20]. The goal of training process is to learn

a good function to correctly map input features to the true GDT-TS scores of given models.

Clustering-based method is widely used in many applications in this research area. The underlying hypothesis in here is that clustering concept can potentially select the good quality protein structures. The idea behind method [21] is the number of low-energy conformations nearby the correct folds is greater than the incorrect ones. Hence, it searches for the largest cluster of structurally related low-energy conformations rather than concentrating on the lowest energy conformation. The technique named SCAR [22] uses root-mean-square distance to quatitatively assess between structures, then identify the similar structure groups by applying k-means clustering algorithm and uses cluster centers as representative models. SPICKER [23] which can view as an improved SCAR and another clustering-based approach [24] categorize pool of structures into many clusters by using radius cuts which can be adjustable depending on the nature of the dataset. Again, a cluster with the most number of neighbors is selected and cluster center is constructed and used as a candidate of near-native conformations.

Nevertheless, these machine learning-based techniques are not flawless. The success of these techniques supports by the fact that the models that is highly similar to the others have better quality which is not necessarily true. An obvious disadvantage of using machine learning-based methods is that groups of protein structures are required. Then, they cannot evaluate protein structure individually unlike in the case of scoring or energy functions.

2.2.3 Consensus-based Algorithms

The third method is using consensus-based algorithm. Likewise, consensus-based idea is similar to the clustering based method which use the underlying idea that the structures similarly predicted to others tends to be more correct than dissimilarly predicted ones. However, consensus-based method does not use the "cluster center" like it is usually built in clustering-based method. Instead, the pair-wise similiarity score is computed between each and every pair of predicted structures. Then, such scores are used as crucial information to predict the quality of structures [19]. For example, 3D-Jury system [25] compares all models with each other and calculates similarity score for them. Then, the system neglects to consider some models according to a predetermined cutoff value which follows the idea of consensus-based algorithm. After that, final score is calculated by averaging the similarity score for set of the considered models.

Consensus-based algorithms stand out from other algorithms, especially in recent CASP competitions. Results by Model Quality Assessment Programs (MQAPs) using similarity score with consensus-based method in CASP7 and CASP8 are significantly better than any other methods as appeared in [3] and [4], respectively.

However, none of algorithms is absolutely perfect and there is no exception for consensus-based algorithm. The major disadvantage of it is the incapability of evaluating each protein structure individually. In other words, consensus-based algorithm needs multiple models in order to build a consensus-based score for each structure. It hinders us from assessing each single protein structure one at

a time. Another disadvantage is that if we have a pool that contains majority of bad model structures. This negatively effects the overall result because that fact that consensus-based algorithm assumes that majority of structures in the pool are good. Therefore, it can give a very poor selection and quality assessment results in those kind of cases.

2.2.4 Multi-approach Algorithms

Another successful approach that has been used among the top groups in model quality assessment category in CASP8 is multi-approach algorithms. Multi-approach idea combines two or more aforementioned algorithms in hoping that it could combine good properties from different approaches together. One example in this category is MULTICOM-CLUSTER server in CASP8 [26], it uses their scoring functions to find the good reference set of models. After that, all models are compared with reference set and global and local quality scores will be predicted. Another example that is belong to this category is QMEANclust [27]. It uses consensus-based information and their own in-house scoring function QMEAN. QMEANclust use QMEAN scoring function to filter out some bad models from consideration. Then, it uses consensus-based score on those filtered models as the final prediction score. Last but not least, ModFOLDclust v2.0 [28] incorporates both consensus-based score and artificial neural network-based score in compensation for the weakness of applying each method independently.

Chapter 3

New Consensus-based Algorithms For Protein Structure QA and Selection Problem

In this section, we propose new consensus-based algorithms for both protein structure prediction QA problem and protein structure selection problem. It has been mentioned that consensus-based method have shown its success over QA category on recent CASP competitions. Moreover, it can be very effective for protein structure selection problem, as well. In structure prediction QA problem, our objective is to numerically assess protein structures as closest as possible to the true GDT-TS to the native structure. Meanwhile, in structure selection problem, we rather focus on how to be able to choose the best structure from the pool. These two problems seem to be similar to each other in the sense that if we have a trusty

solution for protein structure QA problem, selecting the best structure should be not too hard to be achieved. So, in this study, we present critical methodology behind our algorithms for both problems. First, we explain some useful protein structure similarity measures. Then, we describe simple consensus-based method as our fundamental idea. Then, later in this section, we will show that with some tweaks on simple consensus-based method, we can achieve better results in both problems.

3.1 Protein Struture Similarity Scores

Structural arrangement is one of the most challenging limitation found in comparing protein structure measures. Even though, there are a lot of measures used to evaluate protein structures similarity comparison(e.g. root mean square diviation or RMSD), they periordically fails to give accurate results because the small perturbations between two protein structures can result in a high RMSD score [29]. This can lead to wrong assessment by suggesting that such two such structures are very dissimilar overall.

In our study, similarity score plays an important role in our consensus-based algorithm because the fact that we base on our primary strategy that we can determine a quality of a structure by using consensus-based score of every other structure. In here, consensus-based score will be computed from similarity score between structures.

3.1.1 GDT-TS

GDT-TS stands for Global Distance Test Total Score which measures global similarity between two given protein structures. The more GDT-TS of the two structures is, the more similar both structures are.

GDT-TS is computed by the following formula:

$$GDT - TS(s_i, s_j) = (P_1 + P_2 + P_4 + P_8)/4 \tag{3.1}$$

where P_d is a percent of residues from s_i that can be superimposed with corresponding residues from s_j under selected distance cutoffs $d, d \in \{1, 2, 4, 8\}$ [29]. Based on formula, the GDT-TS is in range of 0 to 1. For a very similar pair of structures, GDT-TS is close to 1. While, for a very dissimilar pair of structures, GDT-TS is close to 0. The calculation of GDT-TS in this study is computed by using TM-score [30]. So, when we compare similarity value of one structure to every other by using GDT-TS measure, resulting in a pair-wise GDT-TS similarity matrix of every pair of structures in the group.

Definition 3 *Let G be the pair-wise GDT-TS matrix.* $G = [g_{i,j}]_{1 \leq i \leq p, 1 \leq j \leq p}$ *satisfies the following conditions:*

$$g_{i,j} \geq 0 \textbf{ (1)}$$

$$g_{i,j} = g_{j,i} \textbf{ (2)}$$

$$g_{i,i} = 1 \textbf{ (3)}$$

Intuitively, the values on the diagonal of pair-wise similarity score matrix is

equal to 1, because they measure the similarity between one structure to itself. Also, the value of $g_{i,j}$ and $g_{j,i}$ must be the same due to the fact that similarity between the same pair of structure must be equal and independent to the order of input two structures.

3.1.2 Q Score

Though, GDT-TS is a very good indicator detecting how similar two structures are, GDT-TS in some cases misses to locate good candidates [29]. Thus, Q score, as opposed to GDT-TS, is another useful metric that aims to overcome this limitation [6]. To calculate Q score, internal distance matrices of two protein structures are extracted and used to indetify the similarity between such pair of structures.

Q Score Formulations

We use the Q score based on formulation presented in [6]. Let matrix R to be a internal distance matrix. Let assume that both structures have the equal number of cα atoms p. $R = \{r_{ij}, r_{ii} = 0, r_{ij} = r_{ji}\}$ is computed independently for both candidates by using coordinates of each Cα atom of residue i against other Cα atom of all other $p - 1$ residues in a protein structure, resulting of $p(p-1)/2$ non-zero items matrix.

Although other distance measures can be applied, we use Euclidean distance for building inner matrix R in this study. Then, we set one matrix as a reference matrix called $R^0 = [r^0_{ij}]_{p \times p}$. For each pair of residues $(i \neq j)$ in matrix Q is computed by the following.

Q score Square Error Formulation:

$$Q^{sq-e} = [q_{ij}]_{p\times p} = \exp[-|r_{ij} - r^0_{ij}|^2]. \tag{3.2}$$

Also, another variation of Q score formulation is called relative error Q-measure [6] which replaces Equation 3.2 by the following formula.

Q score Relative Error Formulation:

$$\begin{aligned} q^1 &= \exp[-|(r_{ij} - r^0_{ij})/r^0_{ij}|]. \\ q^2 &= \exp[-|(r_{ij} - r^0_{ij})/r_{ij}|]. \\ Q^{rel-e} &= [q_{ij}]_{p\times p} = \frac{(q^1 + q^2)}{2}. \end{aligned} \tag{3.3}$$

Definition 4 *Let Q be the Q score matrix.* $Q = [q_{i,j}]_{1\le i\le p, 1\le j\le p}$ *satisfies the following conditions:*

$$q_{i,j} \ge 0 \textbf{ (1)}$$

$$q_{i,j} = q_{j,i} \textbf{ (2)}$$

$$q_{i,i} = 1 \textbf{ (3)}$$

Notice that matrix Q has utterly same properties as matrix G presented in GDT-TS similarity score. Only diffference is that matrix Q using Q score as protein structure similarity score, while matrix G using GDT-TS as protein structure similarity score.

Overall similarity score between two structures can be computed by averaging every element in $Q = \{q_{ij}\}$ which is refered as Q_{total} in this paper. For a perfectly

matched pair of structures, $|r_{ij} - r^0_{ij}| = 0$ which yields $q_{ij} = 1$ and $Q_{total} = 1$. For a very different pair of structures, $|r_{ij} - r^0_{ij}| >> 0$ which yields $q_{ij} \approx 0$ and $Q_{total} \approx 0$ [6].

Furthermore, in order to get in-depth structural information, we also use Q_{short} and Q_{long} by computing only the q_{ij} that are in range of $|i - j| \leq 20$ and $|i - j| > 20$, respectively. After that, we sort those pairs and average top 20, 40, 60, 80 and 100% of the best pairs that is in range of $|i - j| \leq 20$ for Q_{short} calculation and is in range of $|i - j| > 20$ for Q_{long} calculation [6].

3.1.3 A New Efficient Method for Computing Q score

According to our Q score formulations in last section, computing Q score for a small size protein might be easy and straightforward, but computing for a large size problem can be very time-consuming. Why? Let us think about process of Q score, specifically. Total time complexity of computing Q score (Q_{total}) is (time of computing internal distance R) + (time of averaging every q_{ij}).

Assume that we are consider the same size of a pair of predicted protein structures. Let n be number of $C\alpha$ atoms of each model. The time complexity of computing internal distance matrix R is $O(n^2)$ since we calculate the pair-wise distance between one $C\alpha$ atom to another for all n atoms in each structure. The time complexity of averaging is linear. Therefore, total time complexity is going to be $O(n^2)$. As the result, if we have a large size problem, computing Q score will be very computational expensive.

We well realize this problem, so we design an efficient way to calculate Q

score. This method based on the well-known statistics approach called sampling. Sampling is the statistical practice concerned with the selection of random subset of individual observations within a population of individuals intended to yield some useful knowledge about the population.

We slightly change our process of computing Q score by including sampling step. Firstly, we randomly select m points of our sampling process and record in any suitable data structure as it is called pool of sampling. Each point represents a coordinate of element (i, j) on matrix R, for example point no.1 represents a coordinate (2,4) and point no.2 represents a coordinate (3,20) on matrix R, etc until we reach m points. However, our points are randomly picked with the exception of the following criteria.

- $i \neq j$

  This is to make sure that values of 0 do not dominate the pool of sampling in later calculation because distance between any coordinate to itself is zero, $r_{i,j;i=j} = 0$.

- (i, j) or (j, i) must not be present in ealier picks

  This is to make sure that the coordinates in our pool of sampling is not redundant which coming from picking the same coordinate or swapped coordinate because distance of two coordinates is independent to the order, $r_{i,j} = r_{j,i}$.

- $|i - j| < 20$ or $|i - j| > 20$, for Q_{short} or Q_{long} only

  In cases, we consider using Q_{short} or Q_{long}, this criterion must be met in

order to follow the formulation. If Q_{total} is opted to use, this structural criterion is not necessary and omitted.

Secondly, we only compute the formula in Equation 3.2 or Equation 3.3 on those points in sampling pool. Third, in this case, we have only i points of $q_{i,j}$ to be considered. Finally, average all considered $q_{i,j}$ in the sampling pool and use average value to be final Q score of a pair of structures.

Sampling points

The sucess of sampling process relies on the fact that the characteristic of sampling points and the actual population. Therefore, we need to have significant number of points m which is big enough to hold that property. However, if number m is too large, advantage in term of time execution of applying sampling approach will be reduced significantly. For instance, sampling process loses its advantage if we opt to use $m = 10000$, but using $m = 100$ is too less and information of using too less sampling points is not accurate to the actual population. Hence, we conduct our experiment on CASP8 dataset to select appropriate m value.

In Figure 3.1, it shows the average percentage of error difference between pair-wise Q score matrix of using sampling with different number of points and using no sampling at all. Intuitively, we can see that percentage of error decrease monotonically when applying more number of sampling points. At $m = 2000$, average percentage of error is down to 2% comparing with no sampling process used.

In Figure 3.2 and 3.3, it shows the time execution of algorithm using pair-wise

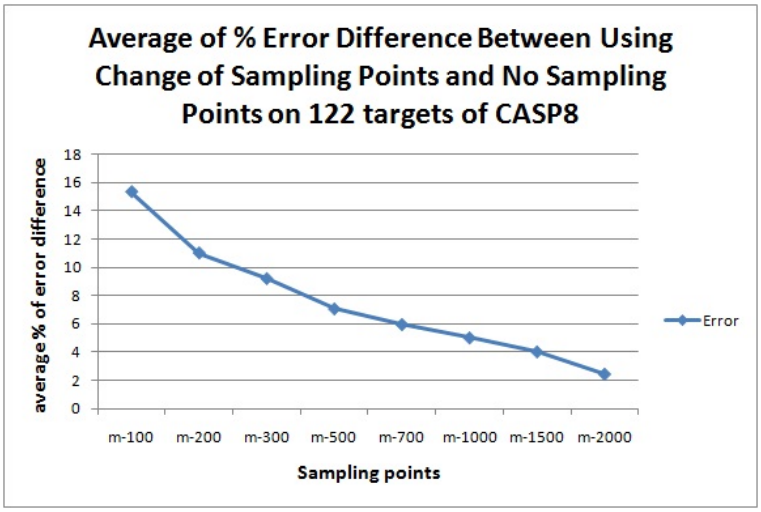

Figure 3.1: Comparison of average % of error difference between pair-wise Q score with different sampling points m (100, 200, 300, 500, 1000, and 2000 points, respectively) used and pair-wise Q score with no sampling points used on 122 targets of CASP8 dataset

Q score sampling 2000 points is significantly faster than using pair-wise GDT-TS. The targets are sorted in descending order by average length of cα atoms of predicted protein structure from participated servers in each target. Total time execution of pair-wise GDT-TS matrix on CASP8 datset is over 194 hours, meanwhile pair-wise Q score with 2000-point sampling is only about 7 hours. On average, pair-wise Q score sampling 2000 points is 20 times quicker than pair-wise GDT-TS per target on CASP8 dataset.

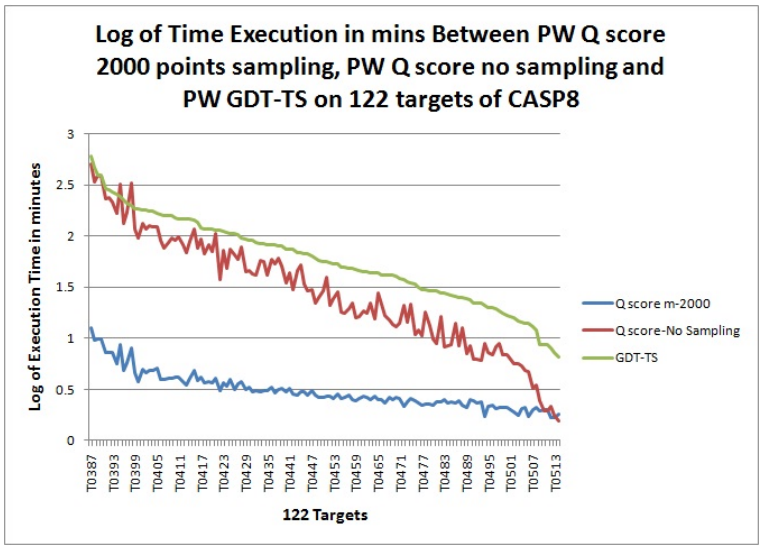

Figure 3.2: Comparison of log of time execution of computing pair-wise Q score using sampling process at 2000 points, time execution of computing pair-wise Q score using no sampling process and time execution of computing pair-wise GDT-TS on the same dataset

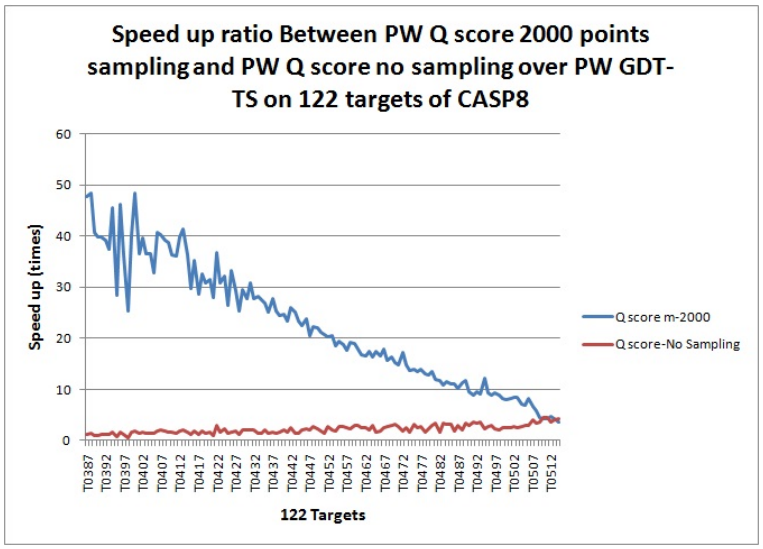

Figure 3.3: Graph represents speed up ratio between using PW Q score 2000 points sampling over PW GDT-TS and PW Q score no sampling over PW GDT-TS on the same dataset

3.2 Simple Consensus-based Algorithms

After, we understand the importance of similarity score in consensus-based algorithm, we like to describe more in detail about the algorithm. The underlying concept of success of using consensus-based algorithm is the fact that the structures that are more similar to the other structures in the same group are more likely to be better quality structures than the less similar ones [31] [19] [25] [5].

Firstly, we begin our discussion with a fairly basic consensus-based algorithm called RefAll [5]. In RefAll, we calculate the quality of given structures by computing any suitable similarity score between one structure and others as known as pair-wise similarity score matrix in this study. Then, we use consensus-based score from such matrix to evaluate each structure. RefAll computes similarity scores between pairs of structures and also uses all the structure as the reference set which is described in pseudo code in Algorithm 1.

Let matrix M be a pair-wise similarity score of set of p structures $S = \{s_i, 1 \leq i \leq p\}$ in a target. Thus, as introduced ealier in in this section, $m_{i,j}$ is the value of similarity score computed by either GDT-TS metric or Q score metric of structure s_i and s_j. The following is the basic properties of matrix M which is exactly the same as matrix G or matrix Q.

We can easily see that in this study, we can alternate of our pair-wise similarity matrix $M = [m_{i,j}]_{p \times p}$ between two choices of similarity scores which are GDT-TS and Q Score as previously discussed (at **(1)** in Algorithm 1). From this point onwards, we will call GRefAll for RefAll with GDT-TS as a similarity score and

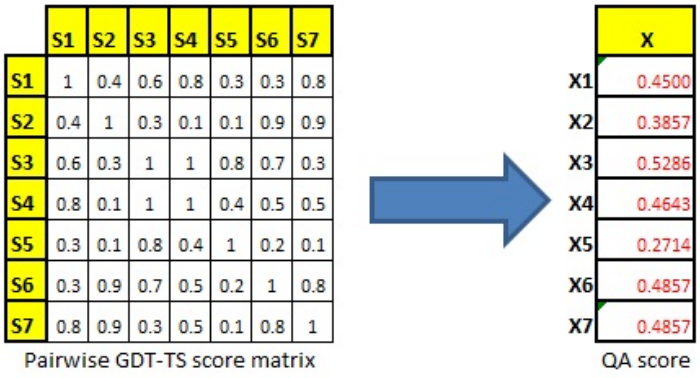

Figure 3.4: The calculation for GRefAll QA consensus-based score is shown. The calculation starts with pairwise GDT-TS score of all protein structures available (figure shows 7 structures for an illustrative example). Then, final QA score is evaluated by averaging all GDT-TS values in each row as presented in Algorithm 1

Algorithm 1 RefAll(S)

Require: Set of protein structures $S = \{s_i, 1 \leq i \leq p\}$

for all $s_i \in S$ **do**

$$x_i = \frac{1}{|S|} \sum_{s_j}^{S} m_{s_i,s_j} \textbf{ (1)}$$

end for

return Set of scores for each candidate, $X = \{x_i, 1 \leq i \leq p\}$

QRefAll for RefAll with Q score as a similarity score.

Specifically, for QRefAll algorithm , the m_{s_i,s_j} in above algorithm is replaced by q_{s_i,s_j} which is item in i^{th} row and j^{th} column of Q matrix. Therefore, if we use Q score at **(1)**, it can be calculated as either Q^{sq-e} by using Equation 3.2 or Q^{rel-e} by using Equation 3.3 and we learned from formulation section that choice of using Q score can alternatively be Q_{total} or Q_{short} or Q_{long}. In summary, QRefAll includes six variations of Q score formulations such as Q^{sq-e}_{short} , Q^{sq-e}_{long} , Q^{sq-e}_{total}

, Q_{short}^{rel-e} , Q_{short}^{rel-e} and Q_{short}^{rel-e} based on formulation used and whether structural infomation is used or not.

On CASP8 dataset, GRefAll algorithm considerably outperforms any other algorithms used in 122 targets of CASP8 dataset in QA category. Moreover, result of GRefAll is even slightly better than the top 5 teams that participating on estimating global model quality assessment category in CASP8 by the pearson correlation result ρ [4] [27]. Note that in the example shown above in Figure 3.4 predicted score X is used to correlate with actual GDT-TS between 7 protein structures and the native structure set Y in order to get ρ.

Table 3.1: Comparison of the Top 5 servers participating in CASP8 competition and simple consensus-based algorithms (RefAll)

Group Name	**Group ID**	**NO. of targets**	**Avg. result(Pearson corr.)**
GRefAll*	N/A	122	0.9290
Pcons_Pcons [32]	239	122	0.9241
ModFOLDclust [28]	31	122	0.9233
SAM-T08-MQAC	56	121	0.9205
QMEANclust [27]	27	121	0.9085
MULTICOM [26]	453	121	0.9021

*GRefAll is the same algorithm as shown in [5]

We have seen that consensus-based algorithm is one of the best solution for protein structure QA problem. It is a promising choice for selection of good protein structure, as well. This can be easily performed right after algorithms finished, the best quality of structure is selected based on the highest value of $x_i \in X$.

Since CASP6, the very first time consensus-based idea was introduced, it shows that a good quality protein structure tends to be more similar to the other predicted structure than a bad quality ones [31]. Also, it strenghtens by CASP7

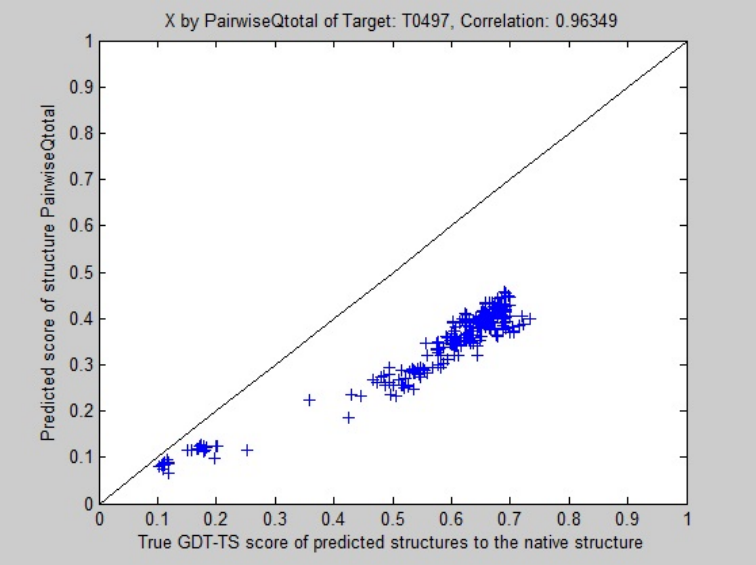

Figure 3.5: Scatter plot between predicted scores by using average of all pair-wise Q_{total}^{sq-e} by QRefAll and GDT-TS score to the native structure of target T0497 on CASP8 dataset

and CASP8 that consensus approach is a methodology used by the leading groups in these competitions and is a suitable option for selecting good protein structures [3] [4]. It has presented in [5] that RefAll algorithm with GDT-TS score as similarity metric clearly outperforms existing best-performing scoring functions, so we will present that QRefAll have some potentials to do that as well.

To strengthen our claim on Q score, Figure 3.5 shows that the correlation between predicted score by using QRefAll with Q_{total}^{sq-e} formulation and GDT-TS score with respect to the native structure. In this example, target T0497 in CASP8 dataset contains 304 candidates from 65 teams. Meanwhile, correlation between predicted scores X by GRefAll algorithm and GDT-TS with respect to native structure is 0.9928 comparing to 0.9635 of Q score on the same target. Despite

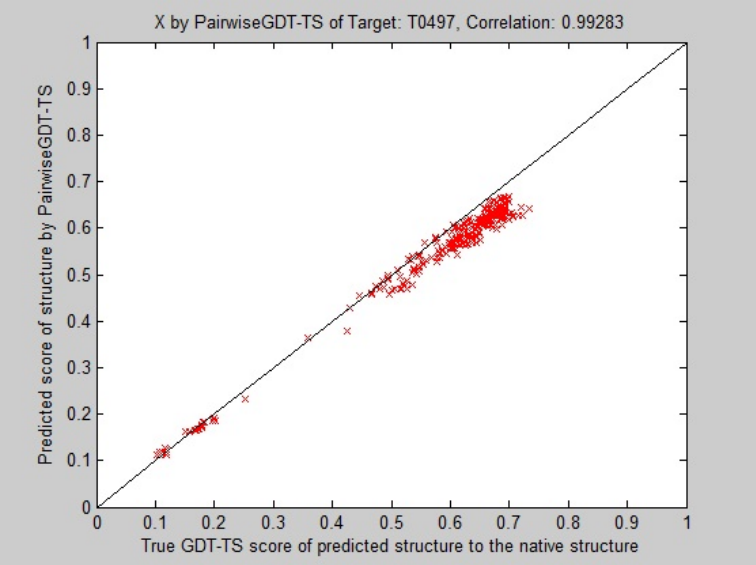

Figure 3.6: Scatter plot between predicted scores by using average of all pair-wise GDT-TS by GRefAll and GDT-TS score to the native structure of target T0497 on CASP8 dataset

the fact that pair-wise GDT-TS has an advantage over pair-wise Q score in terms of correlation result, the correlation of Q score is very high which clearly suggests that Q score can also be a good alternative of ranking structures.

3.3 Multiple Scores Consensus-based Algorithm

In order to take advantage of structural information, we also want to utilize Q_{short}, Q_{long} and Q_{total} as well. So, we propose AVGQRefAll and IRankQRefAll algorithms for combining different Q score metrics together.

Algorithm 2 AVGQRefAll(S)

Require: Set of protein structures $S = \{s_i, 1 \leq i \leq p\}$

for all $s_i \in S$ **do**

$$x_i^t = \frac{1}{|S|} \sum_{s_j}^{S} q_{total\ of\ s_i,s_j} \textbf{ (1)}$$

$$x_i^s = \frac{1}{|S|} \sum_{s_j}^{S} q_{short\ of\ s_i,s_j} \textbf{ (2)}$$

$$x_i^l = \frac{1}{|S|} \sum_{s_j}^{S} q_{long\ of\ s_i,s_j} \textbf{ (3)}$$

$$x_i = \frac{(x_i^t + x_i^s + x_i^l)}{3}$$

end for

return Set of scores for each candidate, $X = \{x_i, 1 \leq i \leq p\}$

Let $RANK$ be a set of integer containing ranking of structures in the same group. $RANK$ can be computed by rank structure based on set of predicted score X. In other words, if $x_i = max(X)$ such that $RANK_i = 1$. IRankQRefAll algorithm shown as Algorithm 3 is also based on the definition of R. Given $RANK_{short}$, $RANK_{long}$ and $RANK_{total}$ be predicted ranks by using predicted score from pair-wise matrix Q_{short}, Q_{long} and Q_{total}, respectively.

Definition 5 *Given* $j, k, p \in \mathbb{N}_1$, $j \leq k$ *and* $j, k \leq p$

$$RANK = \{RANK_i, 1 \leq i \leq p\}$$

$$RANK(j:k) = \{r_j, ..., r_k\}$$

The main difference between AVGQRefAll and IRankQRefAll algorithms is the idea of combining different Q scores together.

The idea of AVGQRefAll algorithm is fairly simple. We calculate predicted scores for Q_{short}, Q_{long} and Q_{total} which are referred as **(1)**, **(2)** and **(3)**, individually. Then, we combine three scores together by averaging them for a final predicted score x_i.

On the other hand, IRankQRefAll algorithm uses intersection operator as method to combine. It is a little more sophisticated than the previous algorithm but simple still. We introduce a (:) symbol as defined in Definition 5 which basically take out a subset of any set at two specified indexes referred as **(4)** in the IRankQRefAll algorithm.

IRankQRefAll calculates predicted score for each Q score as the same way as it does in AVGQRefAll algorithm. However, at the very last step, we take intersection of multiple ranking sets. Different Q score formulations should give distinct ranks of given structures. Even though it is possible that 2 or more Q score types can give both exactly $RANK$ in the same order, chance to occur is very unlikely. Then, we take the first b ranked common index(es) from three ranking systems as a result of set I. After algorithm finishes, best protein structures are selected by based on intersection of rank of corresponding structure. Value b can be an suitable integer regarding to how many top structure we would like to pick

from pool.

Algorithm 3 IRankQRefAll(S)

Require: Set of protein structures $S = \{s_i, 1 \leq i \leq p\}$, No. of top structures b

for all $s_i \in S$ **do**

$$x_i^t = \frac{1}{|S|} \sum_{s_j}^{S} q_{total\ of\ s_i,s_j} \textbf{ (1)}$$

$$x_i^s = \frac{1}{|S|} \sum_{s_j}^{S} q_{short\ of\ s_i,s_j} \textbf{ (2)}$$

$$x_i^l = \frac{1}{|S|} \sum_{s_j}^{S} q_{long\ of\ s_i,s_j} \textbf{ (3)}$$

end for

$RANK_{total} =$ Rank(S, X_{total}) in descending order **(4)**

$RANK_{short} =$ Rank(S, X_{short}) in descending order **(4)**

$RANK_{long} =$ Rank(S, X_{long}) in descending order **(4)**

$I = \emptyset$, $k = 1$

while size$(I) < b$ **do**

$c =$ Intersection$(RANK_{total}(1:k), RANK_{short}(1:k), RANK_{long}(1:k))$ **(4)**

if $c = \emptyset$ **then**

//do nothing here

else

$I \leftarrow c$

end if

$k \leftarrow k + 1$

end while

return Intersection of Rank, I

Regardless, parameter learning process does not pertain to any of Q score-related algorithms, since parameters in each Q score formulation are known and fixed. So, in the result chapter, all Q score related algortihms have been applied based on the formulations given in this chapter.

3.4 Weighted Average Consensus-based Algorithm

As we have seen, simple consensus-based algorithm (RefAll), especially GRefAll, is very successful method on recent CASP datasets. However, there is still room to improve over the current algorithm. So, later in this chapter, we develop our new algorithms based on the success of RefAll and try to add some ideas on top of basic consensus-based RefAll in order to get the better results than using pure RefAll algorithm.

The very first idea that can be adjusted based on RefAll algorithm is usage of the reference set. In RefAll, we assume that all of the structures have the equal contributions on the consensus-based score. Hence, the structure that has more other similar structures will receive more contributions to the consensus-based score than the structure that has less other similar structures. As the result of that, structures that have more other similar structures will appear to have bigger prediction scores, eventually.

With abovementioned concept, it would be a good idea to assign a non-equal contribution based on the similarity score for each structure. It has become the concept of the weighted average concencus-based score which is unlike the simple average consensus-based score of RefAll algorithm.

3.4.1 How to assign weight?

Initially, the idea to assign weight starts from the remove redundancy idea of Ref-Select algorithm in [5]. Even though underlying concept is somewhat similar, we

deploy a different implementation in this study.

So, we need some weight calculation methods. The calculation of weight uses consensus-based scores which can be obtained from pair-wise similarity score matrix. It evaluates from first row of matrix which corresponds to each predicted protein structure and so on. Thus, we develop a couple of varied weighted calculation functions based on our underlying assumptions.

3.4.2 Sigmoid Weighted Function : SigW Algorithm

In this section, we try to reduce the effect of the redundant structures and increase the effect of the dissimilar structures. The definition of "similar" and "dissimilar" here is based on the pair-wise similarity score in matrix M.

Sigmoid Weighted Function

$$Sig(x) = \frac{1}{1 + e^{c(x-0.5)}} \tag{3.4}$$

The input x in above equation is for items in pair-wise similarity score matrix (GDT-TS or Q score) which is in range of interval of zero to one. Thus, in $c = 10$, when we plug the biggest possible value of similarity score which is $x = 1$ into equation resulting $y \approx 0$. Likewise, when we plug the least possible value of similarity score which is $x = 0$, it results $y \approx 1$.

As the result of that, output value is transformed into a weight matrix which meets our strategy of removing effect of redundant structures.

Thus, we propose a new algorithm called SigW algorithm to predict protein

structure quality.

We introduce some more variables i.e. let $W = \{w_i, 1 \leq i \leq p\}$ be the weight vector corresponding to p structures in a target, let $T = [t_{i,j}]_{1 \leq i \leq p, 1 \leq j \leq p}$ be the weight matrix after applying Equation 3.4 to all elements of pair-wise similarity score matrix ($M = [m_{i,j}]_{p \times p}$). Also, keep in mind that index (i, j) of pair-wise similarity score matrix M and weight matrix T always corresponds with same pair of structure s_i and structure s_j.

Algorithm 4 SigW(S)

Require: Set of protein structures $S = \{s_i, 1 \leq i \leq p\}$, Parameter $c > 1$

for all $s_i \in S$ **do**

for all $s_j \in S$ **do**

$$t_{i,j} = \frac{1}{1 + e^{c(m_{s_i,s_j} - 0.5)}} \textbf{ (1)}$$

end for

$$w_i = \frac{\sum_{s_i \neq s_j}^{S} t_{s_i,s_j}}{|S| - 1}$$

end for

for all $s_i \in S$ **do**

$$x_i = \frac{\sum_{s_j, s_i \neq s_j}^{s} (w_{s_i} \cdot m_{s_i,s_j})}{\sum_{s_j}^{S} w_{s_j}} \textbf{ (2)}$$

end for

return Set of scores for each candidate, $X = \{x_i, 1 \leq i \leq p\}$

To elaborate, **(1)** in Algorithm 4 is the calculation of weight matrix $T = [t_{i,j}]$ by applying Equation 3.4 to the pair-wise similarity score matrix M. Weighted average process over weight vector W is executed by calculation in **(2)**. To avoid

confusion of using different similarity measures, SigW algorithm with GDT-TS similarity measure and SigW algorithm with Q score similarity measure are called GSigW and QSigW algorithms, respectively.

This brings us to another interesting question which is how to determine the value of c in the equation. Which one is more suitable, big or small c? First of all, let see the graph presented how change to the weight function of different values of c make according to the Equation 3.4.

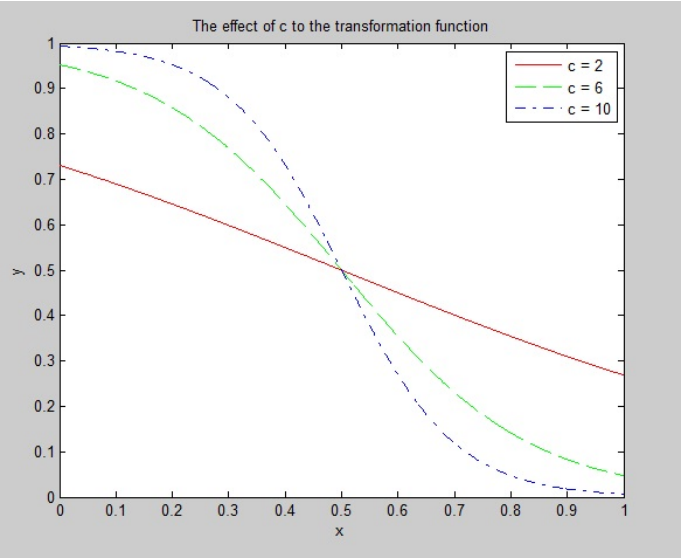

Figure 3.7: The effect of different parameter c to the Sigmoid Weighted Function

According to the graph on Figure 3.7, we can see that the change of c results in how steep the function is. Firstly, we want to cover most of y value in the interval $[0, 1]$, if $x \in [0, 1]$ which is the range of possible similarity scores (maximum = 1, minimum = 0). We decide not use $c = 2$ by the red line in Figure 3.7. However, that doesn't clearly tell us what what the value of c we should use. So, the approach we could do to answer this question is to experiment on how parameter different c values affect to actual correlation results.

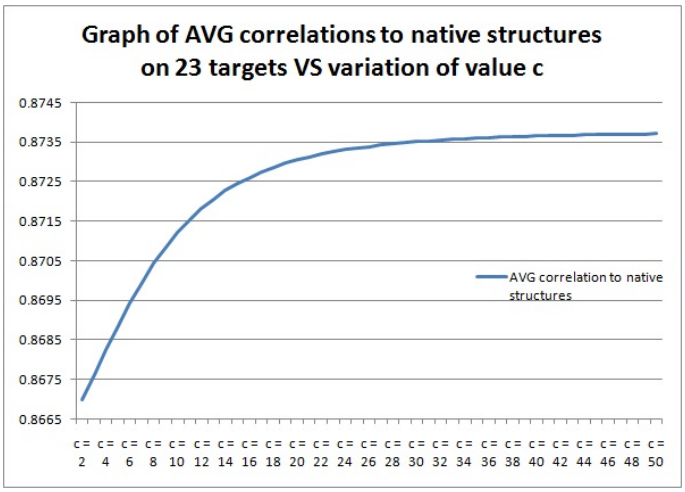

Figure 3.8: Comparison of the variations of parameter c in Equation 3.4 and average of Pearson correlation to the native structures on randomly selected 23 targets of CASP8

Corresponding to the graph on Figure 3.8, we can conclude that when we increase c to the Equation 3.4, the correlation results monotonically increase, as well. However, the curve seems to be stable with the c value reach at a certain point. This evidence leads us to the next discussion in weighted average consensus-based algorithm.

3.4.3 Step Weighted Function : StepW Algorithm

It has been shown that bigger c gave us the better result. When we examine the graph in Figure 3.9, it shows that at $c = 1000$, the function seems to be a step function with 0.5 as a indicator. This still maintains the fact that we want to reduce effect of redundant structures in terms of similarity score to the final consensus-based score. However, with the step function as a weighted function, weight reduction seems to be crisper than the sigmoid one. So, we have more options instead of using Sigmoid-curve as a weight function, we also could use a step function as an alternative weight function.

Similar to previous definition, we need our step function to be a generic function. Therefore, step indicator should be adjustable as we use parameter c in Equation 3.5.

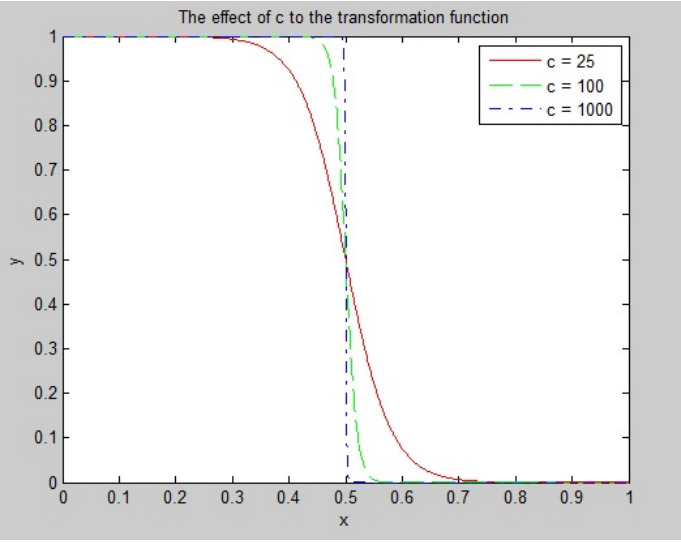

Figure 3.9: The effect of different parameter $c = 25, c = 100, c = 1000$ to the Sigmoid weighted function

So, we mathematically define Step Weighted Function by using parameter c as a cutoff value as the following;

Step Weighted Function

$$Step(x) = \begin{cases} 0 & \text{if } x \geq c; \\ 1 & \text{if } x < c. \end{cases} \tag{3.5}$$

Again, input x in above equation is similarity score like it appears in Equation 3.4. However, when we apply step weighted function, output value y has only binary value (0 or 1) as opposed to Sigmoid weighted function which has possible output values y spanning over most of the value in interval of [0,1] (in big c value, such as $c > 5$). The meaning of c value in Step Weighted Function is that we want to reduce the weight of any pair-wise item in pair-wise similarity matrix that has value over or equal c by giving output value of 0 according to the equation.

Below is an illustrative example of how Step Weighted Function is used on pair-wise GDT-TS score matrix.

Then, we put Step Weighted Function into action as shown in Algorithm 5. StepW algorithm is basically constructed by applying the SigW algorithm (Algorithm 4). The main difference is the approach to build weight matrix T by using Step weighted function Equation 3.5 instead of using Sigmoid Weighted Function. It means that weight vector W of both algorithms will be definitely different, while the rest of calculations are still unaltered by using consensus-based pair-wise similarity score to execute weighted average part. Similarly, we name GStepW and QStepW algorithm for different choices of similarity measures.

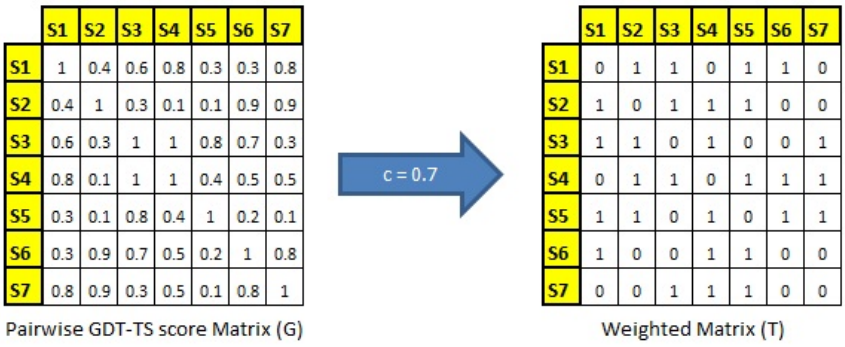

Figure 3.10: Example of applying Step weighted function in Equation 3.5 to matrix G

Algorithm 5 StepW(S)

Require: Set of protein structures $S = \{s_i, 1 \leq i \leq p\}$, Parameter $c > 1$

for all $s_i \in S$ **do**

for all $s_j \in S$ **do**

$$t_{i,j} = \begin{cases} 0 & \text{if } m_{s_i,s_j} \geq c; \\ 1 & \text{if } m_{s_i,s_j} < c. \end{cases} \quad \textbf{(1)}$$

end for

$$w_i = \frac{\sum_{s_i \neq s_j}^{S} t_{s_i,s_j}}{|S| - 1}$$

end for

for all $s_i \in S$ **do**

$$x_i = \frac{\sum_{s_j, s_i \neq s_j}^{s} (w_{s_i} \cdot m_{s_i,s_j})}{\sum_{s_j}^{S} w_{s_j}} \quad \textbf{(2)}$$

end for

return Set of scores for each candidate, $X = \{x_i, 1 \leq i \leq p\}$

3.4.4 Rectangular Weighted Function : RectW Algorithm

As shown in the previous section, assigning different weights is our idea to improve the result. Next, we extend our thoughts to that we not only bring down the effect of similar structures, but we also reduce the effect of dissimilar structures. Since, we know that dissimilar structures also can harm to the final correlation, removing them can be a possible plan to meliorate the result even more. However, building a function to achieve that is not trivial because choice of function can be limitless. Thus, for the sake of simplicity, we make simple addition on the Step Weighted Function into a Rectangular Weighted Function in order to satisfy our ideas.

Hence, we can define Rentangular Weighted Function by the following;

Rectangular Weighted Function

$$Rect(x) = \begin{cases} 1 & \text{if } a < x < b, \text{where } a \neq b \text{ and } b > a; \\ 0 & \text{otherwise.} \end{cases} \tag{3.6}$$

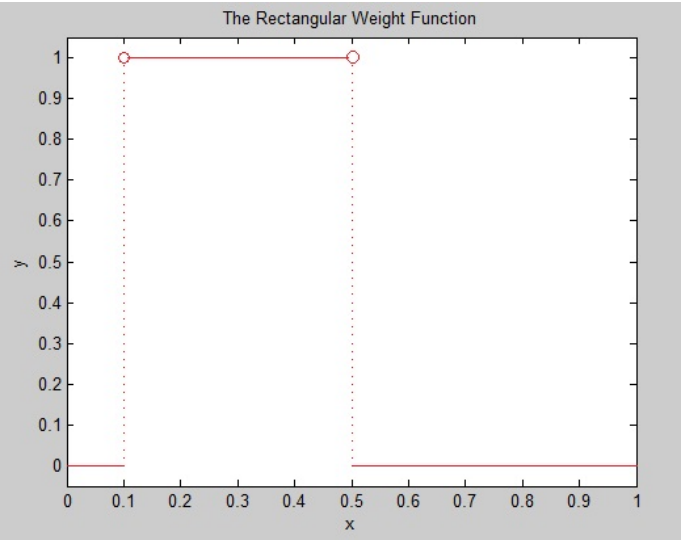

Figure 3.11: Graph of Rectangular weighted function in Equation 3.6 with $a = 0.1$, $b = 0.5$

From the graph presented in Figure 3.11, it shows that Rectangular Weighted Function will contribute weight to 1 only for similarity scores of every pair of structures that is bigger than a ($a = 0.1$ in above example) and as well lower than b ($b = 0.5$ in above example). Meanwhile, the rest of pairs that do not meet that condition will be assigned weight to 0 as it is presented in Equation 3.6.

Here is algorithm with using Rectangular Weight Function for weight calculation;

Algorithm 6 RectW(S)

Require: Set of protein structures $S = \{s_i, 1 \leq i \leq p\}$, Parameter $a,\ b > 1$ and $b > a$

for all $s_i \in S$ **do**

for all $s_j \in S$ **do**

$$t_{i,j} = \begin{cases} 1 & \text{if } a < m_{s_i,s_j} < b; \\ 0 & \text{otherwise.} \end{cases} \quad \textbf{(1)}$$

end for

$$w_i = \frac{\sum\limits_{s_i \neq s_j}^{S} t_{s_i,s_j}}{|S| - 1}$$

end for

for all $s_i \in S$ **do**

$$x_i = \frac{\sum\limits_{s_j, s_i \neq s_j}^{s} (w_{s_i} \cdot m_{s_i,s_j})}{\sum\limits_{s_j}^{S} w_{s_j}} \quad \textbf{(2)}$$

end for

return Set of scores for each candidate, $X = \{x_i, 1 \leq i \leq p\}$

As we can see that, the only main difference between RectW and the rest of algorithms presented in this study is weight matrix calculation T. RectW algorithm use Rectangular Weight Function as weight calculation function. GRectW algorithm and QRectW algorithm distinguish between the use of GDT-TS and Q score as protein structure similarity measure in the similar fashion.

Though, these algorithms could outperform the basic consensus-based algorithm (RefAll), they are inevitably dependable on good parameters. So, how do we decide on choices of paramters? In next section, we have built experimental scheme to find suitable values for parameters such as c in SigW and StepW, also a, b for RectW. It will be discussed in-depth on the approach on how we are able to select appropriate values for those parameters.

Chapter 4

Experimental Scheme and Results

4.1 Dataset and System Specification

We have tested our algorithms on CASP8 dataset. Originally, CASP8 dataset provides 128 targets which correspond to 128 different protein structures. Candidate structures are submitted by different computer-based protein predictors. These predictions are accessible by CASP8 official website [2]. After all of server predictions were submitted, quality assessment (QA) teams are requested to assess quantitatively server predictions and submitted a QA result for predicted structures in each target.

Algorithms were run on Red Hat Enterprise Linux Server release 5.4 (Tikanga) with 8 processors of Intel Xeon(R) CPU E5440 @ 2.83GHz and total memory of 16GB. Linux kernel is x86_64 which is 64-bit architecture. Programming langauges in this thesis are C++, Matlab and Ruby.

In our experiments, we choose 122 targets from CASP8 dataset to evaluate our methods and disregard the other six targets due to cancellation by assessors and/or organizer. Each target's difficulty alters in five categories based on prediction data. These five categories are: the hardest - free modeling (FM), fold recognition (FR), comparative modeling: hard (CM_H), comparative modeling: medium (CM_M) and the easiest - comparative modeling: easy (CM_E) [33] [34]. Also, we measure performance of our methods base on these five categories.

According to our algorithms presented in this study, some of them are sensitive to parameters. Before, we can actually test our algorithms to the dataset, we need to design a training process for our algorithms to be able to learn proper parameters. Thus, we divided 122 targets from CASP8 into two groups: one for training and another for testing.

4.1.1 Target's Difficulty

CASP8's targets differ considerably in terms of prediction's hardness, and particularly our algorithms might perform differently subject to targets' difficulies. It found that average of all items of pair-wise GDT-TS matrix varies on different targets. Thus, we have decided to break 122 targets into 3 categories which are Easy, Medium and Hard by such value.

First of all, we randomly break our targets into two sets as referred in Figure 4.1. Train dataset is composed of 40% out of total targets in CASP8 targets and the rest of targets belong to test dataset. Then, we classify train dataset into 3 categories based on the hardness of targets which use criteria presented in Table

Table 4.1: Distribution of 122 CASP8 targets in each group based on our group criteria

Group Name	NO. of targets	Range of AVG pair-wise GDT-TS score
Hard	30	(0, 0.3]
Medium	38	(0.3, 0.5]
Easy	54	(0.5, 1]

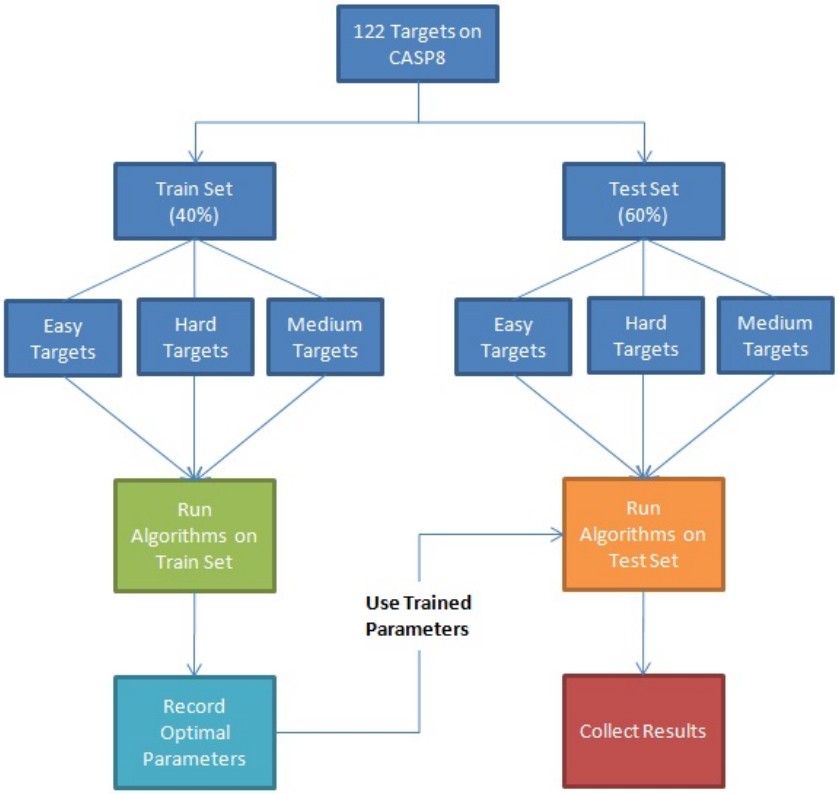

Figure 4.1: Diagram shows the experimental scheme in order to find a set of proper parameters for the algorithms

4.1. After that, we run our algorithms on the train set in order to find set of good parameters according correlation score to the known native structures. When

we finish training process, we record proper parameters one per target's hardness group (one for Hard group, one for Medium group and one for Easy group). This is based on our hypothesis that our algorithm might perform differently depending on variations of target's difficulty. Finally, we use those parameters for applying on the test dataset and presenting the final results.

4.2 Learning Parameters of Algorithms

4.2.1 GSigW Algorithm

The following results were produced by running Algorithm 4 on train dataset. Our training process is performed by running SigW algorithm on different c values in range of $c = [1, 2, 3, ..., 35]$ in Equation 3.4. In each round of one c value, we compute a set of predicted score X. In order to compare performances on different c values, we correlate between set of predicted score with actual GDT-TS score of protein structures to native structure of each target N. So, we use correlation value y described in Equation 3.1 as a decisive value for selecting the best parameters.

Regarding to Graph 4.2 and Graph 4.3 of average of pearson correlation between predicted quality scores and true GDT-TS scores in Hard and Medium group of train dataset, we can see that avearge correlation are monotonically increasing with the bigger values of parameter c in Equation 3.4 in SigW algorithm. This verify the fact that the more steep the sigmoid function is, the better average of correlation score is, in the hard and medium groups.

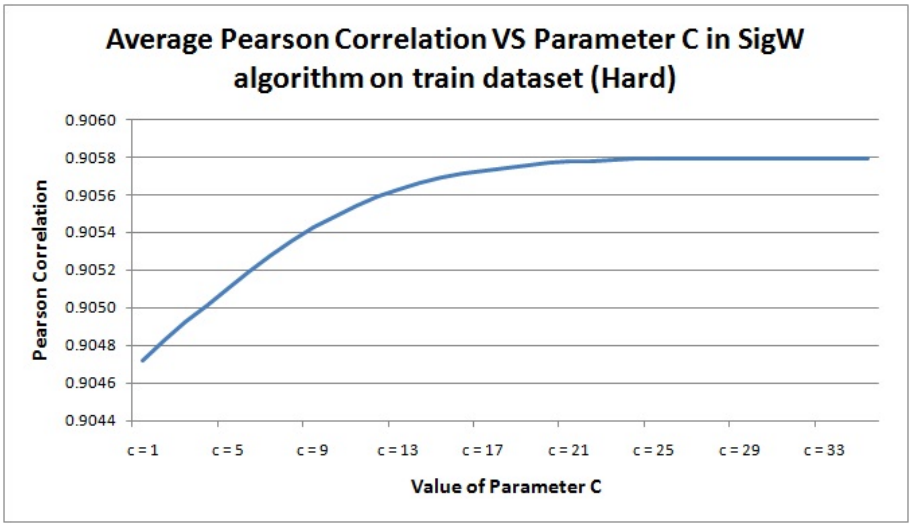

Figure 4.2: Comparison of the average of pearson correlations and values of parameter c on Hard targets (15 targets) in train dataset by SigW algorithm

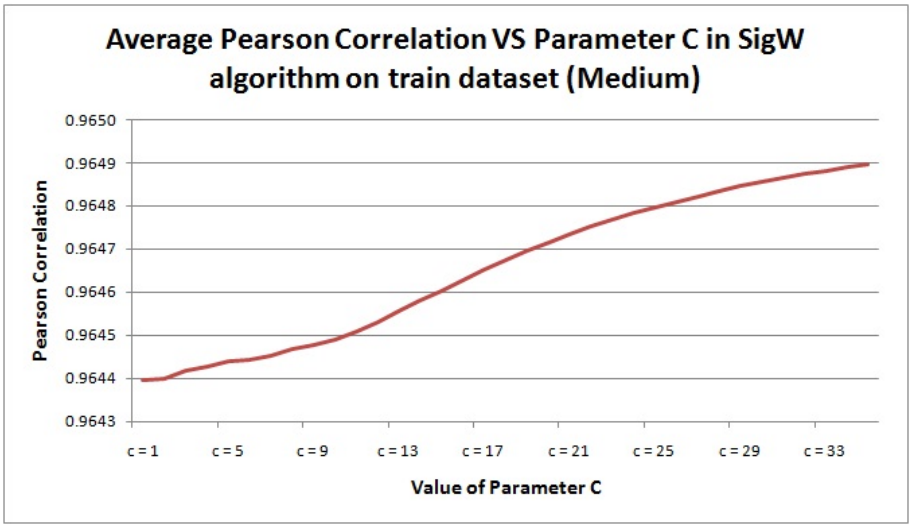

Figure 4.3: Comparison of the average of pearson correlations and values of parameter c on Medium targets (19 targets) in train dataset by SigW algorithm

Graph 4.4 is different from others. We can see that the bigger values of parameter c help increase the average of correlation in range of $c = [1, ..., 13]$, but

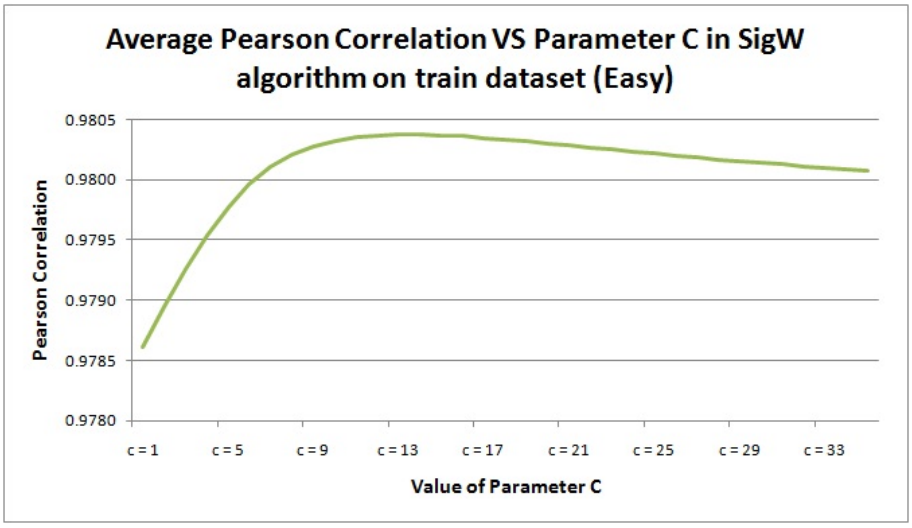

Figure 4.4: Comparison of the average of pearson correlations and values of parameter c on Easy targets (16 targets) in train dataset by SigW algorithm

in range of $c \geq 13$, average correlation of all easy targets group starts to decline and seems to be stable around 0.9800 at $c = 5$ (not shown in the figure). It simply describes that the very steep sigmoid function might not be a perfect choice for dealing with easy group target.

4.2.2 GStepW Algorithm

The following results were produced by running Algorithm 5 on train dataset. The steps of training algorithm are similar as it has been performed in SigW algorithm. In StepW algorithm, meaning of parameter is different than what we use in SigW algorithm. Possible choices of parameter c will be a GDT-TS score value which start from [0.1, 0.2, ... 1] which determine any point of GDT-TS value that we want to reduce the effect of the final weight. Obviously, we excluded $c = 0$ out of our consideration otherwise it would set all of the stuctures' weight to zero (every g_{ij} of matrix G is greater or equal 0.).

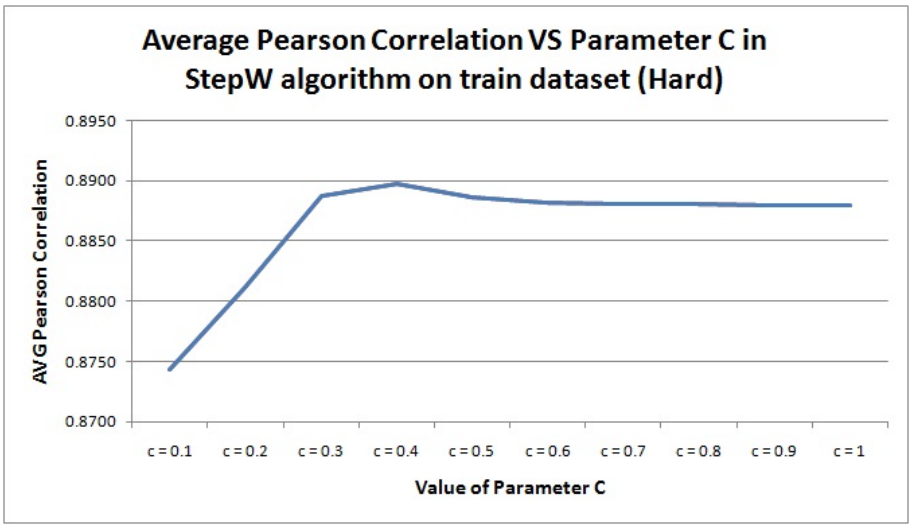

Figure 4.5: Comparison of the average of pearson correlations and values of parameter c on Hard targets (15 targets) in train dataset by StepW algorithm

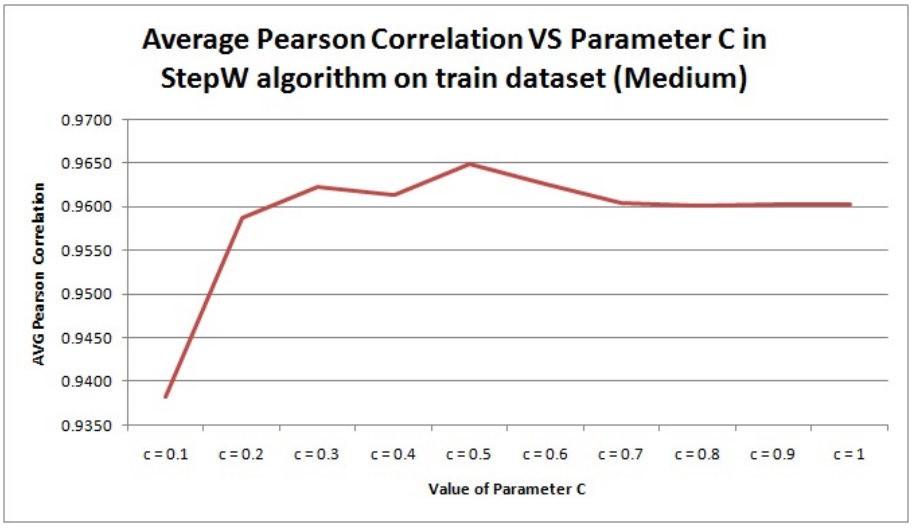

Figure 4.6: Comparison of the average of pearson correlations and values of parameter c on Medium targets (19 targets) in train dataset by StepW algorithm

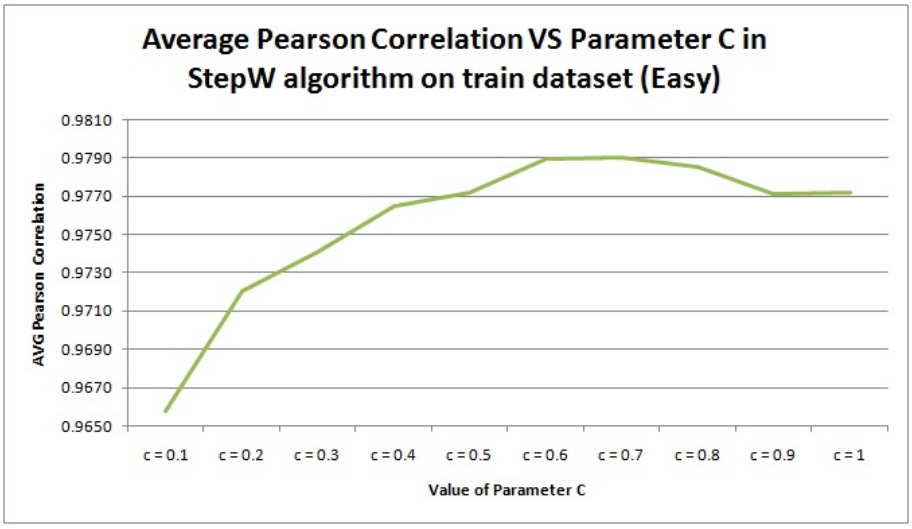

Figure 4.7: Comparison of the average of pearson correlations and values of parameter c on Easy targets (16 targets) in train dataset by StepW algorithm

We can see that each of target group has different optimal parameter values. For hard group, optimal value of c is at 0.4 GDT-TS value based on average of

pearson correlation to the actual GDT-TS score. For medium group, optimal value of c is at 0.5 GDT-TS value. Lastly, for easy group, best value of c is at 0.7 GDT-TS value.

4.2.3 GRectW Algorithm

The following results were produced by executing Algorithm 6 on train dataset. Based on this algorithm, we need two parameters a, b to define how big our rectangular shape is (like shown in Figure 3.11). Originally, our idea started by using all possible combinations $a, b = [0.1, 0.2, 0.3, ..., 1]$ which could be computationally expensive. Thus, we take advantage of learned optimal values from StepW algorithm as one fixed parameter for RectW algorithm. Then, another parameter will be experimented intensively in the same approach as it has been in StepW algorithm by spanning over $[0.1, 0.2, 0.3, ..., 1]$. In this case, we can reduce the computation cost by 10 times lesser than the initial idea.

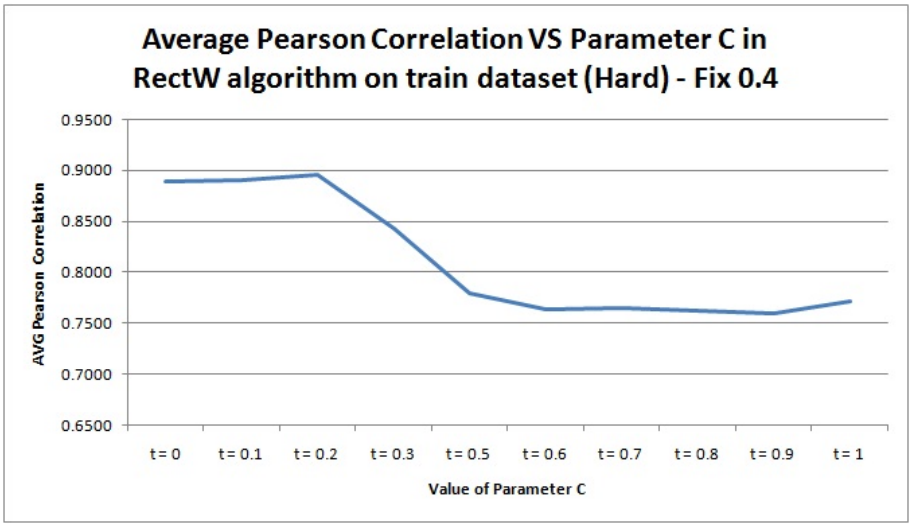

Figure 4.8: Comparison of the average of pearson correlations and values of parameter on Hard targets (15 targets) in train dataset by RectW algorithm

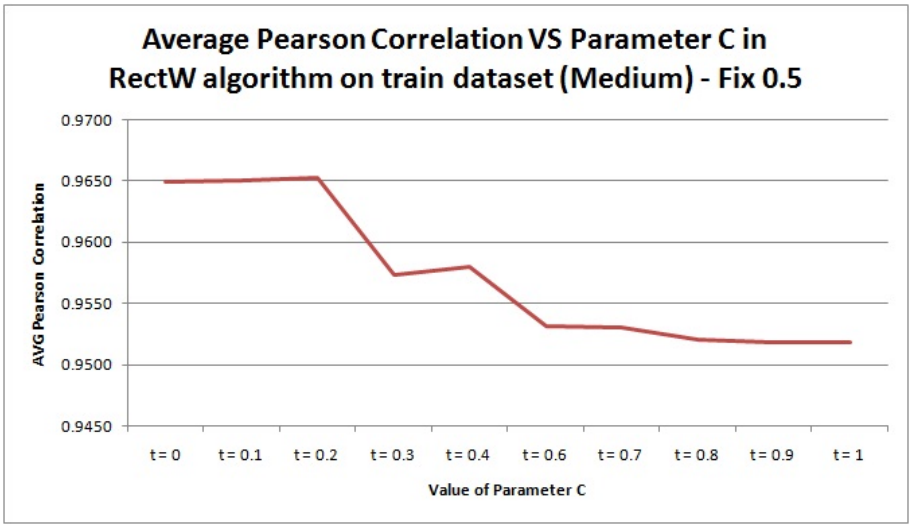

Figure 4.9: Comparison of the average of pearson correlations and values of parameter on Medium targets (19 targets) in train dataset by RectW algorithm

From those graphes, they present the fact that optimal a, b values in Hard group are $a = 0.2,\ b = 0.4$ as in Figure 4.8, for Medium group, they are $a = 0.2,\ b =$

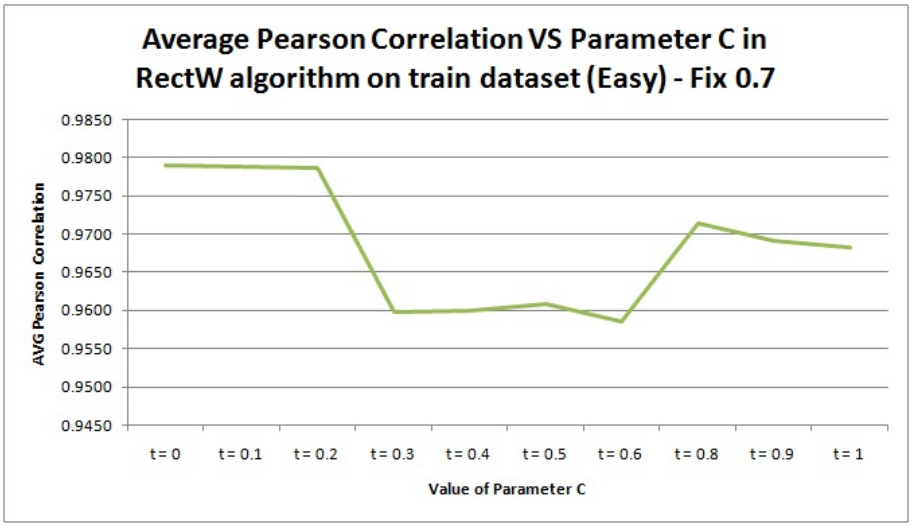

Figure 4.10: Comparison of the average of pearson correlations and values of parameter on Easy targets (16 targets) in train dataset by RectW algorithm

0.5 as in Figure 4.9 and best values are $a = 0,\ b = 0.7$ for Easy group as in Figure 4.10. We can see that smaller band of Rectangular Weighted Function works in the harder cases ($0.2 - 0.4$ and $0.2 - 0.5$), but for easier case, larger band should be used ($0 - 0.7$).

4.3 Experimental Results

As we discussed in problem formulation section, once again we split results into two parts which are QA results and selection results. Be noted that all parameters used in this section has been collected from train dataset as presented in last section.

4.3.1 Protein Structure Prediction QA Results

After, involved algorithms were learned with training process in previous section, results of paremeter are the following;

Table 4.2: Optimal parameters after training in each group based on our group criteria

Group Name	**Range**	**GSigW Alg.**(c)	**GStepW Alg.**(c)	**GRectW Alg.**(a, b)
Hard	(0, 0.3]	35	0.4	0.2, 0.4
Medium	(0.3, 0.5]	35	0.5	0.2, 0.5
Easy	(0.5, 1]	13	0.7	0, 0.7

Note that, RefAll uses all pair-wise GDT-TS score and all structures as reference or called GRefAll in study. RefSelect algorithm is presented in [5] and all of the results of RefSelect is provided by the author of that paper.

The results were performed on test dataset (60% of 122 targets) by using parameters with our group criteria in Table 4.2. Then, results are graphically compared by categorizing test datset into 5 categories which are free modeling (FM), fold recognition (FR), comparative modeling: hard (CM_H), comparative modeling: medium (CM_M) and comparative modeling: easy (CM_E) [33] [34].

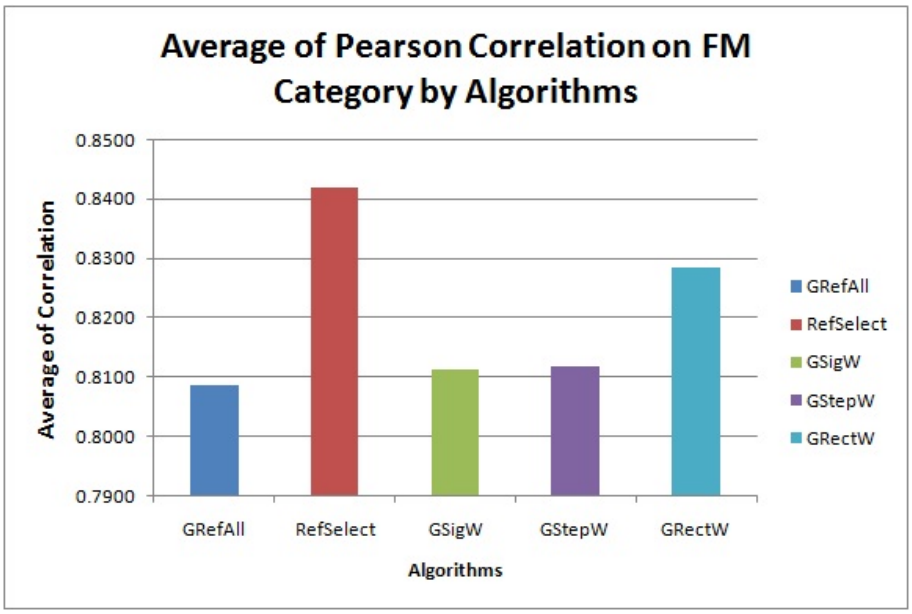

Figure 4.11: Comparison of the average of pearson correlations and algorithms on FM category of train datset

Based on Figure 4.11, we can see that our RectW algorithm with GDT-TS or GRectW algorithm indeed performs better than GRefAll algorithm by about 2.45% on test dataset of FM category. Also, GRectW algorithm outperform other proposed algorithms such as GStepW and GSigW algorithms. GStepW and GSigW are very close in performance in this category, though GStepW is slightly better due to the reason that GStepW is a strongly steep version of GSigW. However, RefSelect is the best performing algorithm in this category. It is better than GRectW algorithm by about 1.64%. The main objective of designing RefSelect algorithm is to focus an improvement on the harder targets which make it very successful on FM category.

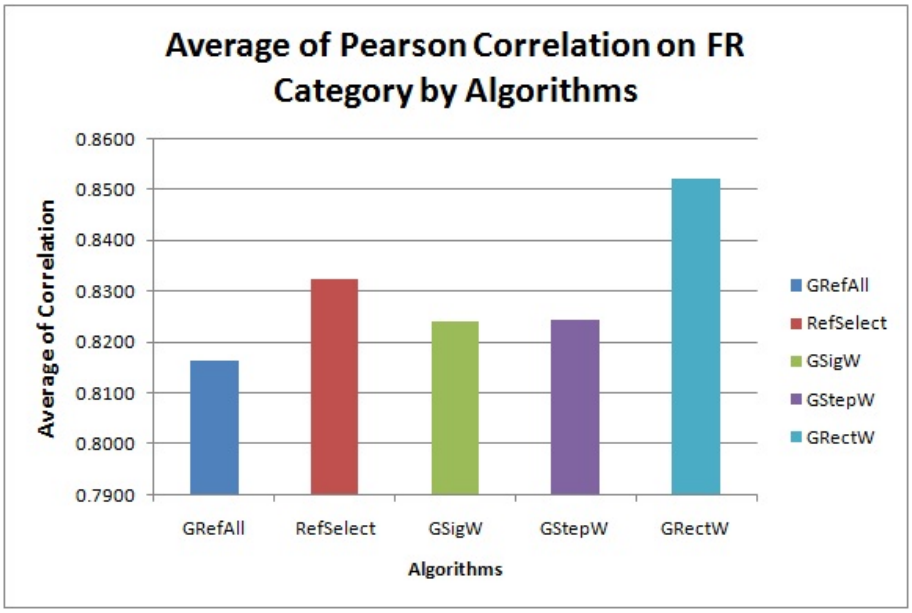

Figure 4.12: Comparison of the average of pearson correlations and algorithms on FR category of train datset

According to Figure 4.12, it shows performances of algorithms where GRectW algorithm begins to shine. In FR category, it shows the largest gap of performace between GRectW and GRefAll which is about 4.39% on test dataset of FR category. RefSelect's performance is better than any other algorithms but GRectW algorithm.

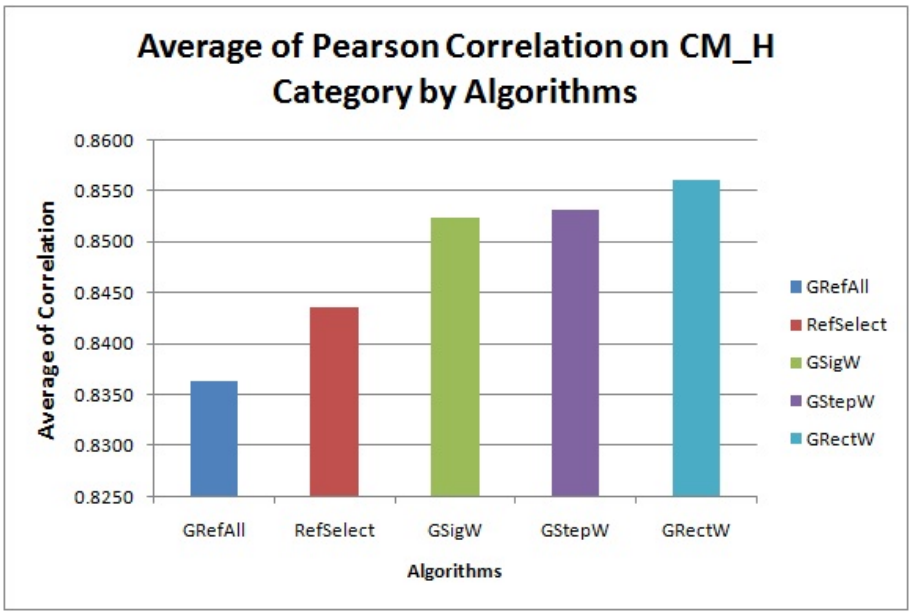

Figure 4.13: Comparison of the average of pearson correlations and algorithms on CM_H category of train datset

As it is shown in Figure 4.13, it is quite clear that GRectW algorithm also outperforms any other algorithms in this category. Especially, GRectW algorithm perform better than GRefAll by 2.36%. GStepW and GSigW algorithms are very close in performance of CM_H category. Even thought RefSelect's performance drop slightly in CM_H group, it is still better than using GRefAll algorithm.

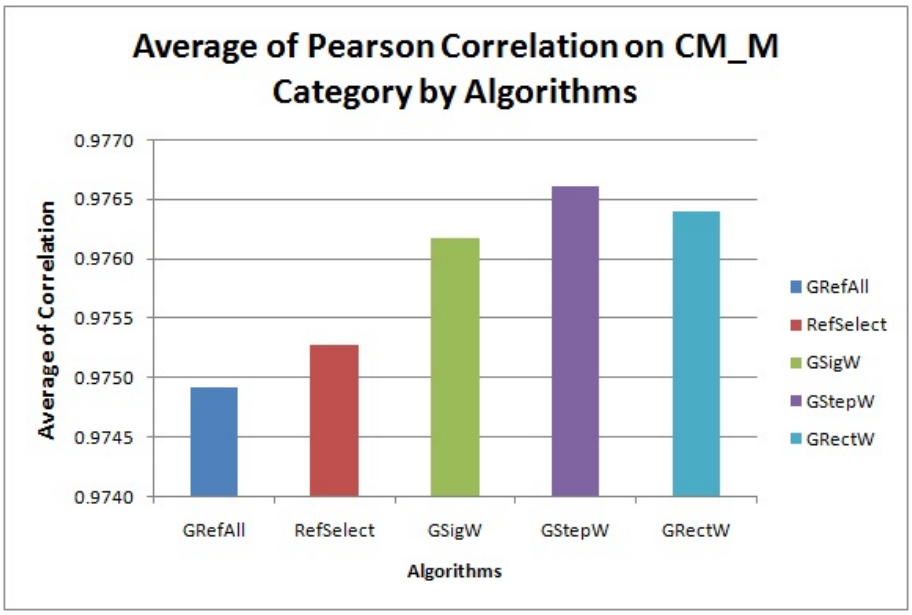

Figure 4.14: Comparison of the average of pearson correlations and algorithms on CM_M category of train datset

Based on Figure 4.14 shows performances in CM_M category. In this category, GStepW algorithm is clearly the best out of tested algorithms. However, the differences between algorithms are very small about 0.2% in performance.

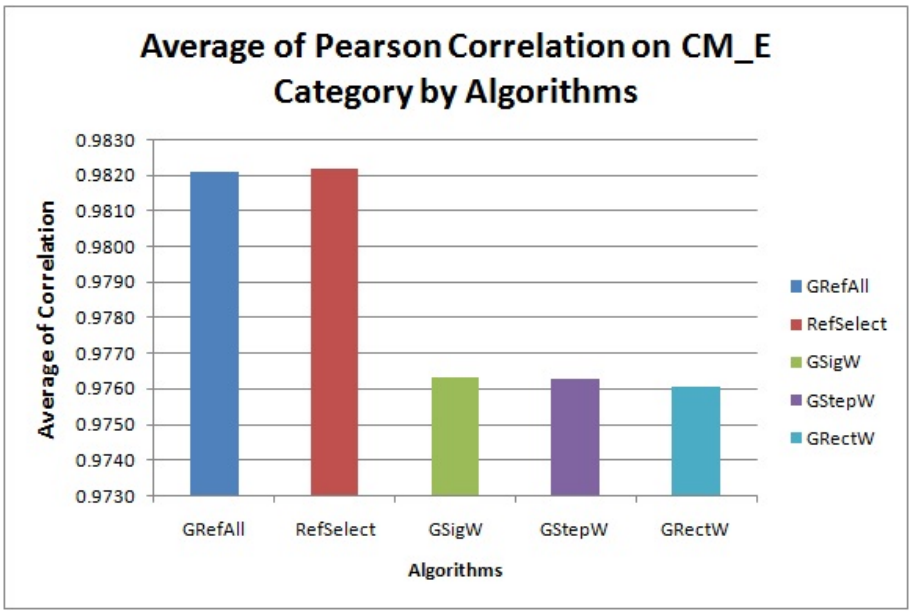

Figure 4.15: Comparison of the average of pearson correlations and algorithms on CM_E category of train datset

In the last category of our experiment presented in Figure 4.15, GRefAll and RefSelect algorithms are on the top. They both show their strengths in the easiest group. RefSelect handles the easiest target group in the similar fashion as GRefAll does. So, it is not very surprising they have relatively similar results in this category. The rest of algorithms are not as good as the top two algorithms.

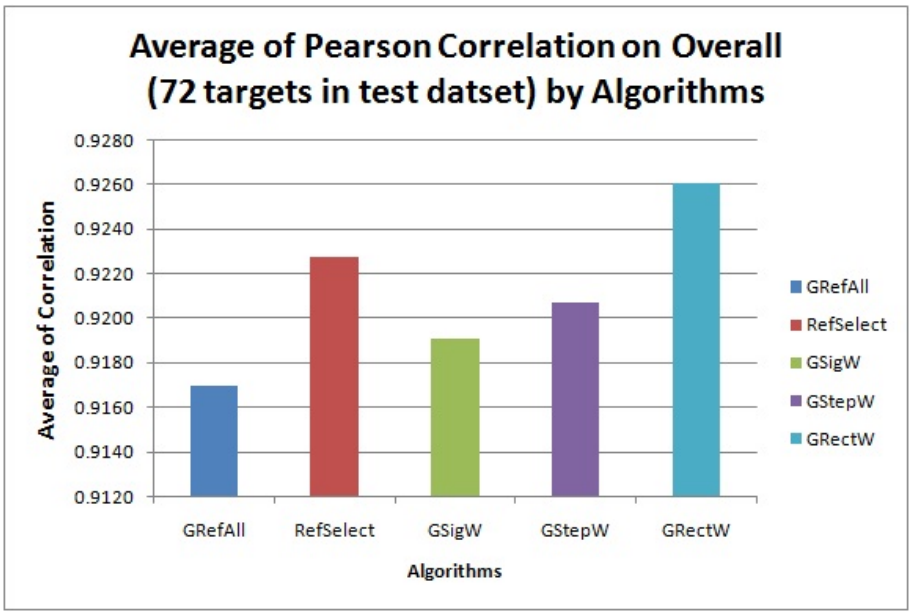

Figure 4.16: Comparison of the average of pearson correlations and algorithms on overall targets of train datset

In overall in Figure 4.16, GRectW algorithm is the best performing algorithm over this datset in 72 targets of CASP8 dataset as known as test dataset. The improvement over GRefAll algorithm slightly dwindles down due to the fact that GRefAll algorithm outperform GRectW algorithm is the easiest group of CM_E category which is the biggest number of targets in CASP8 dataset and our train dataset. Nonetheless, GRectW algorithm is still successful algorithm and clearly outperforms other algorithms because the improvements in other harder groups (FR, FM, CM_H and CM_M category) are big enough.

Table 4.3: Comparison of the Top 5 servers participating in CASP8 competition, RefAll algorithm and RectW algorithm

Group Name	**NO. of targets**	**Avg. result(Pearson corr.)**
GRectW	**122**	**0.9370**
RefSelect [5]	122	0.9335
GRefAll [5]	122	0.9290
Pcons_Pcons [32]	122	0.9241
ModFOLDclust [28]	122	0.9233
SAM-T08-MQAC	121	0.9205
QMEANclust [27]	121	0.9010
MULTICOM [26]	121	0.9021
AVGQRefAll	122	0.8786

Finally, we can draw the conclusions from our experiment that in the harder groups of targets, strategy of removing redundant structures together with removing outliers based on GDT-TS as similarity score is performing relatively well on improving quality of predicted score over GRefAll algorithm. It is shown by graph in Figure 4.11, 4.12 and 4.13 as in FM, FR and CM_H category, respectively. On the other hand, the improvement of GRectW algorithm over GRefAll algorithm starts to be minimal in CM_M category as in Figure 4.14. In the eaisest targets (CM_E) as in Figure 4.15, GRectW algorithm has no more edge. GRefAll is basically the best performing algorithm which emphasizes that using all strcuture as reference is more appropriate strategy for dealing with easier targets than harder ones.

Last but not least, we compare correlation result of GRectW algorithm with the proper setting of parameters and correlation result of other top teams in 122 targets of CASP8 competition. We can see that GRectW can beat other teams

and be on the top with the optimal setting of parameters. Also, we compare the result of using Q score as similarity measure as shown as AVGQRefAll which is the best algorithm from our proposed Q score-based algorithms. It is clear to say that Q score-based algorithms are bad for QA problem because result shows that correlation of predicted scores from AVGQRefAll to the true GDT-TS to native structures is the lowest of the pack of tested algorithms.

4.3.2 Protein Structure Selection Results

Selection results were performed on all 122 targets CASP8 dataset. For those parameter-dependent algorithms, we select optimal parameters based on previous section. Methodology in here is very straightforward which is comparing performances of our algorithms side-by-side with other methods by selecting the best structure according to predicted quality scores on the same dataset.

In order to gauge how good our algorithms are, we also pick some best performing scoring functions on the market to-date. This is to show how well widely-known scoring functions are able to choose a top structure. These scoring functions are, for instance, (1) OPUS-Ca is a knowledge-based potential function requiring only Cα positions [16], (2) DOPE is an atomic distance-dependent statistical potential calculated from a sample of native protein structures [35], (3) DFIRE is a statistical energy function based on the reference state of distance-scaled, finite ideal gases [17], (4) RAPDF is a residue-specic all-atom probability discriminatory function [18].

Table 4.4: The average GDT-TS scores of the top one structures selected by GRefAll, RefSelect and widely-used scoring functions on 122 CASP8 targets

		Scoring functions				**Algorithms [5]**	
Group	**True**	**OPUS-Ca**	**DOPE**	**DFIRE**	**RAPDF**	**GRefAll**	**RefSelect**
FM	0.2830	0.1536	0.1841	0.1724	0.1996	0.1951	0.2147
FR	0.4329	0.2567	0.2912	0.3048	0.2988	0.3551	0.3696
CM_H	0.6524	0.4713	0.4282	0.5423	0.5265	0.5711	0.5788
CM_M	0.7502	0.5807	0.5477	0.6240	0.6649	0.7172	0.7171
CM_E	0.8884	0.6748	0.8025	0.7846	0.8330	0.8558	0.8543
Overall	0.6799	0.4985	0.5196	0.5627	0.5856	0.6267	0.6315

Table 4.5: The average GDT-TS scores of the top one structures selected by our Q score-related algorithms with square-error formulation on 122 CASP8 targets

		QRefAll			**Multiple Q score**	
Group	**True**	Q_{short}^{sq-e}	Q_{long}^{sq-e}	Q_{total}^{sq-e}	**AVGQRefAll**	**IRankQRefAll**
FM	0.2830	0.1912	0.1864	0.1847	0.2234	0.2150
FR	0.4329	0.3387	0.3392	0.3395	0.3498	0.3561
CM_H	0.6524	0.5633	0.5604	0.5375	0.5698	0.5689
CM_M	0.7502	0.7143	0.7056	0.6924	0.7108	0.7107
CM_E	0.8884	0.8591	0.8540	0.8399	0.8527	0.8516
Overall	0.6799	0.6219	0.6170	0.6051	0.6242	0.6245

As shown in Table 4.4 and 4.5, we compare our methods' performances against others. Firstly, the first data column of each table presents GDT-TS scores to the native stuctures of the actual top1 structure which defined the boundary any of tested algorithms and scoring functions could possibly achieve in this experiment. Next 4 columns of Table 4.4 show the performance of how well widely-known scoring functions are able to pick a top structure. Then, 3 columns next to the first column of Table 4.5 show the performance of our algorithms on the same datset. Columns with Q_{short}^{sq-e}, Q_{long}^{sq-e} and Q_{total}^{sq-e} labels denotes result from Algorithm 1 with Q score (QRefAll algorithm) by using Q_{short}^{sq-e}, Q_{long}^{sq-e} and Q_{total}^{sq-e} as a protein structure similarity measures, respectively.

We can see that all Q score consensus-based algorithms can clearly choose the better top 1 structure than the best performing scoring functions based on GDT-TS to the respect native structure. The best performing scoring function based on CASP8 dataset is RAPDF which has overall average GDT-TS of chosen structure as 0.5856. Meanwhile, the IRankQRefAll's performance is better than RAPDF by 6% at 0.6245. However, RefSelect's performance is the best out of

Table 4.6: The average GDT-TS scores of the top one structures selected by our weighted average algorithms on 122 CASP8 targets

		Weighted AVG		
Group	**True**	**GSigW**	**GStepW**	**GRectW**
FM	0.2830	0.1951	0.1951	0.2055
FR	0.4329	0.3566	0.3572	0.3593
CM_H	0.6524	0.5748	0.5745	0.5747
CM_M	0.7502	0.7202	0.7181	0.7281
CM_E	0.8884	0.8575	0.8566	0.8577
Overall	0.6799	0.6291	0.6292	0.6304

tested algorithms in this paper. Even though, top1 structure selected by Q score consensus-based algorithms is not as good as selected by RefSelect or GRefAll algorithms, they have shown that they are good alternatives for solving protein selection problem.

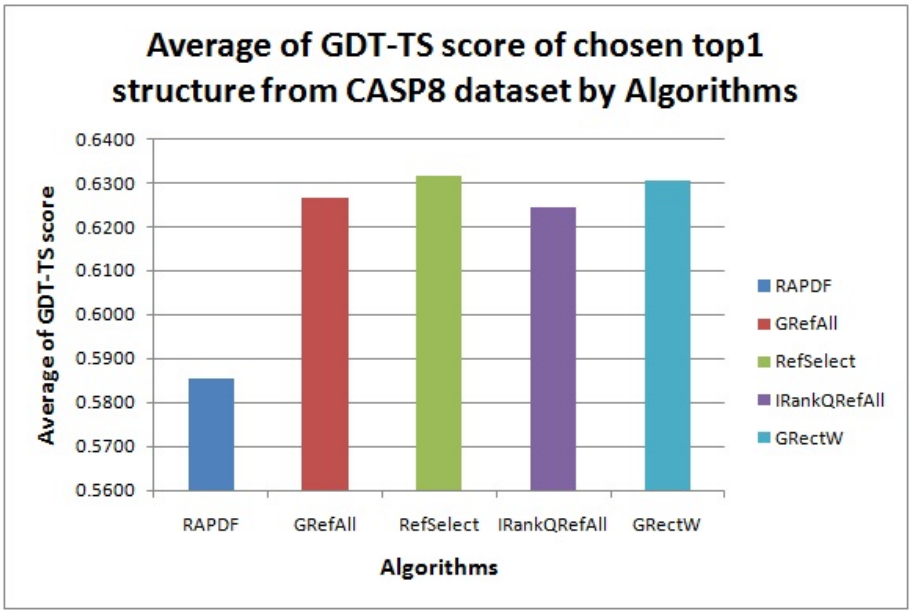

Figure 4.17: Comparison the average of GDT-TS score of chosen top1 structure from all 122 targets of CASP8 dataset and different algorithms

Futhermore, we also compare weighted average methods against other methods. Selection results on according to graph in Figure 4.17 show that using GSigW and GStepW algorithms even more superior than using any of Q score-based algorithms. Again, GRectW is the best among those weighted average algorithms. Though, RefSelect is still the best performing algorithm based on tested algorithms, it clearly shows that removing redundant structure and disimilar structure of GRectW with proper parameter settings improves not only results on protein structure QA problem, but also results on protein structure selection problem over the simple consensus-based algorithm.

Chapter 5

Conclusion

This paper has presented novel algorithms that are designed for QA in protein structure prediction. Our idea is to remove redundant structures to get the better reference set which can improve an overall result (for SigW and Step algorithms) and is to not only romove redundant structures but also remove dissimilar structures (for RectW algorithm). Our proposed algorithms have been thoroughly compare with the best current methods on the bechmark datset suggesting that our algorithm, GRectW, significantly perform better than simple consensus-based algorithm, GRefAll. RectW has advantage over any other algorithms on the harder cases of targets which has low predicted protein structure quality based on GDT-TS score. This fact makes RectW standing out from the rest of algorithms.

Second, algorithms for selecting better protein structures have been proposed. Technique is applying Q score measure as protein structure similarity score. Proven by our experiments on 122 targets of CASP8 dataset, performance of our al-

gorithm overcome the state-of-the-art scoring functions by at least 6.6% on an overall score and at least 10% on hardest case (FM) of 122 targets of CASP8. However, RefSelect algorithm is better in terms of the GDT-TS score of top1 on overall performance. Q score consensus-based methods also show a potential to be an effective way for selecting good stucture in hard targets. Also, GRectW algorithm performs well in selection problem. Compared with top1 structure selected by the best-performing scoring function RAPDF, top1 structure selected by GRectW algorithm has better GDT-TS value by 7.6% on overall score of 122 targets in CASP8 and even better than any of Q score-based algorithm proposed in this study.

In summary, contributions of this thesis are the following;

- **An efficient method of computing Q Score**

  We use idea of random sampling in order to reduce calculation time. With random sampling, we can achieve about 1 or 2 orders of magnitude faster than normal time execution with only 2% loss. Results from experiment shows that QRefAll is consistently better than any other well-known scoring functions in the market to-date.

- **New consensus-based methods for QA in protein structure prediction**

  We give different weights to different structures and compute weighted average as final predicted scores. Assignment of weights based on idea of removing too similar structures, SigW algorithm and Step algorithm and removing both too similar and too dissimilar structures, RectW algorithm. Re-

sults from our best-performing methods are superior than any top teams in CASP8 competition. Also, it has some improvement over basic consensus-based algorithm. Finally, our server implemented by this method ranks in top 3 based on preliminary evaluation of CASP9 competition in 2010 [36].

Future work is to improve the overall results to be more superior both correlation result and selection result. It can be done by several ideas.

- **Improving the accuracy of target hardness indicator**

  We could apply more sophisticated methods. Options can be well-known approaches like Support Vector Machine (SVM), Neural Network (NN), Genetic Algorithm (GA) etc. Since we know that our algorithms will perform differently based on hardness of targets, improving accuracy of target difficulty indicator could make the result better.

- **Finding good ways to combine existing scoring functions**

  Even though, existing scoring functions, like OPUS-Ca, Cheng, RAPDF, DFIRE are not very accurate in terms of indetifying near-native structure, they are moderately useful of differentiating very poor quality structures out of the rest structures. We have tried to use this idea as a strategy of throwing the poor ones and keeping the good ones. However, performance of our implementation on this idea is not quite successful. Nonetheless, it is not by any means to say that scoring functions are useless.

- **Developing some other better protein structure similarity measures**

  Aside from GDT-TS score, Q score is the proven example of how other

similarity measures can be successful in selecting good protein structure. There is no prevent for any other good similarity measures to be successful as well.

- **Designing more reliable single-model assessment algorithm**

  It is due to the fact that the main weakness of consensus-based algorithm is the need of multiple structures in calculation process. If we want to know quality of one structrue at a time, consensus-based algorithm will be unable to perform. Single model assessment algorithm will become very handy in such situation. However, it can be very difficult to achieve because of the complexity of the protein structure.

Bibliography

[1] C. Floudas, "Computational methods in protein structure prediction," *Biotechnology and Bioengineering*, vol. 97, pp. 207–213, 2007.

[2] P. S. P. Center. (1994-2010) 8th community wide experiment on the critical assessment of techniques for protein structure prediction. Protein Structure Prediction Center. [Online]. Available: http://predictioncenter.gc.ucdavis.edu/casp8/index.cgi

[3] D. Cozzetto, A. Kryshtafovych, M. Ceriani, and A. Tramontano, "Assessment of predictions in the model quality assessment category," *Proteins: Structure, Function, and Bioinformatics*, vol. 69, pp. 175–183, 2007.

[4] D. Cozzetto, A. Kryshtafovych, and A. Tramontano, "Evaluation of casp8 model quality predictions," *Proteins: Structure, Function, and Bioinformatics*, vol. 77, pp. 157–166, 2009.

[5] Q. Wang, Y. Shang, and D. Xu, "Protein structure selection based on consensus," 2010.

[6] M. Ben-David, O. Noivirt-Brik, A. Paz, J. Prilusky, J. L. Sussman, and Y. Levy, "Assessment of casp8 structure predictions for template free targets," *Proteins: Structure, Function, and Bioinformatics*, vol. 77, pp. 50–65, 2009.

[7] J. Moult, K. Fidelis, A. Kryshtafovych, B. Rost, T. Hubbard, and A. Tramontano, "Critical assessment of methods of protein structure prediction - round vii," *Proteins: Structure, Function, and Bioinformatics*, vol. 69, pp. 3–9, 2007.

[8] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, "Charmm a program for macromolecular energy, minimization, and dynamics calculations," *Journal of Computational Chemistry*, vol. 4, pp. 187–217, 1982.

[9] W. C. Still, A. Tempczyk, R. C. Hawley, and T. Hendrickson, "Semianalytical treatment of solvation for molecular mechanics and dynamics," *Journal of The American Chemical Society*, vol. 112, pp. 6127–6129, 1990.

[10] H. Lu and J. Skolnick, "A distance-dependent atomic knowledge-based potential for improved protein structure selection," *Proteins: Structure, Function, and Bioinformatics*, vol. 44, pp. 223–232, 2001.

[11] H. Gohlkea, M. Hendlicha, and G. Klebe, "Knowledge-based scoring function to predict protein-ligand interactions," *Journal of Molecular Biology*, vol. 295, pp. 337–356, 2000.

[12] Y. Duan and P. A. Kollman, "Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution," *Science*, vol. 282, pp. 740–744, 1998.

[13] D. T. Jones, W. R. Taytor, and J. M. Thornton, "A new approach to protein fold recognition," *Nature*, vol. 358, pp. 86–89, 1992.

[14] Y. Zhou, H. Zhou, C. Zhang, and S. Liu, "What is a desirable statistical energy functions for proteins and how can it be obtained?" *Cell Biochemistry and Biophysics*, vol. 46, pp. 165–174, 2006.

[15] M. Lu, A. D. Dousis, and J. Ma, "Opus-psp: An orientation-dependent statistical all-atom potential derived from side-chain packing," *Journal of Molecular Biology*, vol. 273, pp. 283–298, 2008.

[16] Y. Wu, M. Lu, M. Chen, J. Li, and J. Ma, "Opus-ca: A knowledge-based potential function requiring only c-alpha positions," *Protein Sci*, vol. 16, pp. 1449–1463, 2007.

[17] 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*, vol. 11, pp. 2714–2726, 2002.

[18] R. Samudrala and J. Moult, "An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction," *Journal of Molecular Biology*, vol. 275, pp. 895–916, 1998.

[19] J. Qiu, W. Sheffler, D. Baker, and W. S. Noble, "Ranking predicted protein structures with support vector regression," *Proteins: Structure, Function, and Bioinformatics*, vol. 71, pp. 1175–1182, 2007.

[20] 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: Structure, Function, and Bioinformatics*, vol. 75, pp. 638–647, 2008.

[21] D. Shortle, K. T. Simons, and D. Baker, "Clustering of low-energy conformations near the native structures of small proteins," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 95, pp. 11 158–11 162, 1998.

[22] M. R. Betancourt and J. Skolnick, "Finding the needle in a haystack: educing native folds from ambiguous ab initio protein structure predictions," *Journal of Computational Chemistry*, vol. 22, pp. 339–353, 2000.

[23] Y. Zhang and J. Skolnick, "Spicker: a clustering approach to identify near-native protein folds," *Journal of Computational Chemistry*, vol. 25, pp. 865–871, 2004.

[24] Q. Wang, Y. Shang, and D. Xu, "A new clustering-based method for protein structure selection," in *IEEE International Joint Conference on Neural Networks*, 2008.

[25] K. Ginalski, A. Elofsson, D. Fischer, and L. Rychlewski, "3d-jury a simple approach to improve protein structure predictions," *Bioinformatics*, vol. 19, pp. 1015–1018, 2003.

[26] J. Cheng, Z. Wang, A. N. Tegge, and J. Eickholt, "Prediction of global and local quality of casp8 models by multicom series," *Proteins: Structure, Function, and Bioinformatics*, vol. 77, pp. 181–184, 2009.

[27] P. Benkert, S. C. E. Tosatto, and T. Schwede, "Global and local model quality estimation at casp8 using the scoring functions qmean and qmeanclust," *Proteins: Structure, Function, and Bioinformatics*, vol. 77, pp. 173–180, 2009.

[28] L. J. McGuffin, "Prediction of global and local model quality in casp8 using the modfold server," *Proteins: Structure, Function, and Bioinformatics*, vol. 77, pp. 185–190, 2009.

[29] A. Zemla, "Lga: a method for finding 3d similarities in protein structures," *Nucleic Acids Research*, vol. 31, pp. 3370–3374, 2003.

[30] Y. Zhang and J. Skolnick, "Scoring function for automated assessment of protein structure template quality," *Proteins: Structure, Function, and Bioinformatics*, vol. 57, pp. 702–710, 2004.

[31] C. Venclovas and M. Margelevicius, "Comparative modeling in casp6 using consensus approach to template selection, sequence-structure alignment, and

structure assessment," *Proteins: Structure, Function, and Bioinformatics*, vol. 7, pp. 99–105, 2005.

[32] P. Larsson, M. J. Skwark, B. Wallner, and A. Elofsson, "Assessment of global and local model quality in casp8 using pcons and proq," *Proteins: Structure, Function, and Bioinformatics*, vol. 77, pp. 167–172, 2009.

[33] S. Shi, J. Pei, R. I. Sadreyev, L. N. Kinch, I. Majumda, J. Tong, H. Cheng, B.-H. Kim, and N. V. Grishin, "Analysis of casp8 targets, predictions, and assessment methods," *Database (Oxford)*, vol. 2009, p. Article ID: bap003, 2009.

[34] M. L. Tress, I. Ezkurdia, and J. S. Richardson, "Target domain definition and classification in casp8," *Proteins: Structure, Function, and Bioinformatics*, vol. 77, pp. 10–17, 2009.

[35] M. yi Shen and A. Sali, "Statistical potential for assessment and prediction of protein structures," *Protein Sci*, vol. 15, pp. 2507–2524, 2006.

[36] J. Cheng. (2010) Automated assessment of casp9 protein structure predictions. University of Missouri-Columbia. [Online]. Available: http://sysbio.rnet.missouri.edu/casp9-assess/qa.php