PROTEIN STRUCTURAL MODELS SELECTION USING 4-mer SEQUENCE AND COMBINED SINGLE AND CONSENSUS SCORES

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

MESHARI SAUD ALAZMI

Dr. Dong Xu, Thesis Supervisor

May 2012

The undersigned, appointed by the dean of the Graduate School,

have examined the thesis entitled

Protein Structural Models Selection Using 4-mer

Sequence and Combined Single and Consensus Scores

Presented by Meshari Alazmi

A candidate for the degree of

Master of Computer Science

and hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Dong Xu

Dr. Yi Shang

Dr. Ioan Kosztin

DEDICATION

First of all, I thank Allah, my God, so much, who is the ultimate creator and the only one worthy of worship because He controls everything from that which is smaller than atom to the most complicated creature.

I also want to thank my family, for their continuous mental encouragement throughout my time here. Special thanks to my mother, who calls me every single day to ask how I am doing.

Many thanks go to my fiancé for her encouragement and patience from far distances with close heart. Though separated by thousands of miles, we are still close in our hearts and full of hope.

ACKNOWLEDGEMENTS

I would like to thank Dr. Dong Xu, my thesis supervisor, for offering me this opportunity to be a part of his research. His instructions and suggestions are very valuable for me. He always pushes me forward to continue studies, try different methods and think from different angles. Also, I would like to thank Zhiquan He, my fellow researcher, who always cooperated with me on this project from the beginning to the end. Both Zhiquan and I advanced our knowledge by discussing many interesting topics on this work. I learned a lot from his extensive experience. I really appreciate his help, and wish him all the best throughout his career. I would like to thank Dr. Jingfen Zhang for her suggestions and help. She always pushes me to work hard and to get things done properly. She has tremendous knowledge. I would like to thank my committee members Dr. Ioan Kosztin, and Dr. Yi Shang, and I would like to thank everyone who made contributions to this protein project.

In addition, I would like to thank my sponsor, Saudi Arabia Cultural Mission, for financial aid. They supported me with tuition and living expenses.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

List of Tables

Table Page

PROTEIN STRUCTURAL MODELS SELECTION USING 4-mer SEQUENCE AND COMBINED SINGLE AND CONSENSUS SCORES

Alazmi, Meshari

Dr. Dong Xu, Thesis Supervisor

ABSTRACT

Quality assessment for protein structure models is an important issue in protein structure prediction. Consensus methods assess each model based on its structural similarity to all the other models in a model set, while single scoring methods, such as Opus-ca and RW, evaluate each model based on its structural properties. In this work, a novel method proposed and developed to effectively combine consensus methods and single scoring methods for better quality assessment. At first, a new method called Single Position Specific Probability (SPSP) Score is proposed based on consensus method using 4-mer sequence. Specifically, every letter in the 4-mer sequence represents a state for a local region consisting of four amino acids. A machine learning method (Neural Network) helped to combine several single scoring methods, RW, DDFire, and OPusCa with consensus methods, SPSP and Consensus Global Distance Test-Total Score (CGDT-TS) to achieve a good combination of all the terms. The method was tested on two benchmark datasets and achieved improvements over the state-of-the-art methods. The first benchmark was on Yang Zhang's data containing 56 targets. The second benchmark was from Rosetta data containing 35 targets. For Zhang's data, the CGDT score is 0.6058, while combined method

achieved 0.6105. For Rosetta data, the CGDT score achieved 0.4255, while combined method achieved 0.4529.

Chapter 1. Introduction

Protein structure prediction is a method to predict the three dimensional structure of a protein from its sequence. Protein structure prediction has been a difficult research topic ever since scientists began developing the cell biology system [55, 56]. The more protein sequences identified, the more work needs to be done to predict their structures which simplifies determining their functions [56]. X-ray crystallography and nuclear magnetic resonance (NMR) are experimental protein structure determination methods. They are very costly and require too much time to determine one structure. Starting with the sequence and using computational methods to predict the structure gives scientists more opportunities to work in this field and solve the problems associated with experimental methods [57]. Some reviews showed the importance of computationally predicted models in applications [58, 59]. For example, high-accuracy models are useful in studying catalytic activity. In addition, predicted models are particularly useful even with medium accuracy up to 6Å RMSD. For example, the function of the protein is predictable using 3-D structures [59].

Quality assessment is one of the most challenging problems in protein structure prediction. It is important to choose the best model within a set of models for further study of attributes. Such study can lead to structure refinement and potential societal benefits in applications such as drug design. In this study, quality assessment of predicted models is based on comparing methods developed with GDT score, chosen standard measurement. This thesis,

therefore, helps solve the problem of how to assess protein structure prediction models.

Quality Assessment methods for protein structure prediction models can be split into four major approaches based on the strategy. The first category is physical-based energies [60, 61]. It mainly calculates atomic-level energy of a model relying on fundamental physics. The disadvantage of this method is that the energy value is sensitive to minor changes in the structure. The second category is knowledge-based scoring functions. Knowledge-based methods give better results than physics-based methods. They rely on statistical information of real proteins' characteristics. For example, Opus-ca [39] is based on molecular interaction energies from C-alpha atoms. Another example is RW [41] scoring method which considers side-chain orientation as a useful input. The third one is the consensus approach [62-64]. The main idea is that all decoys (structural models) will vote to the one that is the closest to them in structural similarity. The best method occurs when the best model is among the major cluster; and it is not recommended to pursue a method if the best model is an outlier in the structural pool. The last approach is machine-learning methods [63, 66]. For example, neural network (NN) and support vector machine (SVM) are based on trained data with known targets. The trained model tested on unseen data. Preprocessing the data to meet classification data needs required formulation to fit the general idea of classification. The features are the method scores. The training process for this method was based on GDT, RMSD, or Z-score

measurements as a target to rank models; then testing was conducted on decoys for unseen targets.

Two methods were developed in this work. The first method is a Single Position Specific Probability (SPSP) Score method using 4-mer sequence. Every letter in the 4-mer sequence represents a state for a local region consisting of four amino acids. The local regions are clustered based on their pseudobond angles between every four C-alpha atoms such as pseudobond bending angles and pseudobond torsion angles. Having all the 4-mer sequences for all the decoys in one target led to comparisons using a consensus-based method. Although this method alone does not have outstanding performance in model selection, when combined with other single score methods such as RW, Opus-ca, and DDFire[73], it outperformed CGDT. The second method is a combination of several scoring methods. In this method, we trained neural network models to effectively capture the underlying correlation among different scoring methods. Specifically, the score differences from different methods for a pair of decoys were used as input features, which were mapped to the difference of actual model quality by neural networks.

The second step consisted of classifying the differences into two classes. If the real GDT difference is positive, then it went to the first class; otherwise, it was in the second class. The training process used leave-one-out method. For training data, 1000 records were randomly picked from N-1 targets. Finally, the model was tested on the remaining target that had not taken anything from for the training purpose.

Assessing models generated by computational methods with known native structure is easily done by measuring the similarity between the model and the real structure. Methods available for that task include GDT, RMSD, and TM-score.

First, GDT (**Global Distance Test**) measures the similarity between two protein structures with identical amino acids sequences but different 3D structures. It ranges between 0 and 1. It scores 1 if the two structures are identical. The GDT score compares two structures under cutoff thresholds which are 1 Å, 2 Å, 4 Å, and 8 Å. Then, the score is the average of the aligned residues with the four thresholds. It is calculated as [51].

$$\mathrm{GDT}(\mathrm{S_i}, \mathrm{S_j}) = (\mathrm{P_1} + \mathrm{P_2} + \mathrm{P_3} + \mathrm{P_4})/4$$

where $\mathrm{P_d}$ is a percent of structurally aligned residues that follow a certain threshold between the $\mathrm{S_i}, \mathrm{S_j}$ proteins structures.

GDT score is used as the major assessment criterion at CASP (Critical Assessment of Structure Prediction), which is a biannual competition for protein structure prediction disciplines. This CASP endorsement led to selection of GDT-TS as the measurement and validation of the methods used in this research [50].

Root-Mean-Square Deviation (RMSD) is the average distance between the residues, while Cα represents the whole residue. RMSD measures the similarity between two structures and translates/rotates them until it minimizes the score. However, it gives a high score for similar structures except in local regions such as a loop if they differ in that region. Thus, the global score was affected by minor changes [50].

Chapter 2. Existing Methods for Protein Structure Prediction Quality Assessment

In this chapter, the protein structure prediction quality assessment (QA) problem is addressed and some terms are defined. Then, some existing methods which fall into four categories: physics-based energies, knowledge-based scoring functions, consensus methods, and machine learning based approaches are explained with more details.

2.1. What is a QA problem?

When protein structure predictions from different servers are used for a target and the native structure of that target is known, quality assessment measures the predicted structure and gives it a score showing how much it is similar to the native structure. Thus, some predictions are near-native; some are close to the native; some share regions with the native, and some are far from the native. The measurement is based on the similarity between the native and predicted structures. Choosing a measurement is a topic subject to debate, but GDT-TS (Global Distance Test-Total) score was chosen as an assessment for the methods because it superimposes the predicted structure until it matches the native thereby minimizing the distances. It works well with good predicted structures. The main reason for using GDT-TS is because of the credibility it has gained in CASP competitions since 2002 [49]. Here, QA gives a score to every

predicted structure. After that, the score can be used to select or rank the models.

2.2. Physics-based Energies:

Any physics-based energy function relies on thermodynamic hypothesis [1]. These functions are based on physical properties in atomic level [1]. Potential energy functions illustrate the relationships between the points (atoms) in the system [70]. Force field functions calculate the potential energy. They are based on internal and external terms, e.g., CHARMM [2] [3], AMBER [4]. Having four sequential atoms, the potential energy calculation is based on the following information:

A) internal terms
   1. Covalent Bond length between two atoms.
   2. The valence angle between 3 adjacent atoms which cares about the atomic orbitals.
   3. The dihedral angle between 1, 2, 3, and 4 atoms which is the angle between 2 planes. The first plane is 1, 2 and 3 atoms. The second plane forms 2, 3, and 4 atoms.
   4. Urey-Bradley distance between 1 and 3 atoms.
   5. The distance between 1 and 4 atoms.
   6. Improper dihedral angle between a line and a plane in 4-point shape.

B) External terms (Van der Waals interactions):
   1. Lennard-Jones (LJ) potential measures the interaction between two neutral atoms.

2. Electrostatic interactions between charged atoms are measured by Coulomb's law.

These are the standard points needed to calculate the potential energy. Other information can be extracted from the structure and it can help in calculating the energy. For example, Y. Harano et. al. [5] considered hydration free energy.

Physics-based approach requires a lot of information which leads to computational processing and is very time-consuming [6]. It is also subject to minor changes in the protein structure prediction. These are the main disadvantages of the physics-based approach.

2.3. Knowledge-based Scoring Functions:

Knowledge-based (statistical) scoring functions are based on relative frequency in a database of a set of proteins which takes the general pattern in the database [7-11]. Statistical scoring functions are also based on optimization method [12-22]. Optimization methods are discussed later. These function inputs record the distances between residues, or the distances between the atoms, and also indicate solvent accessibility, dihedral angles, and packing density. Because they describe the protein, they are called protein descriptors [70]. After organizing this information according to what the statistical scoring functions need, this information is processed based on statistical analysis of the weighted linear sum of pairwise contacts [15, 16, 23-26]. Calculations are based on structural distribution of structural descriptors or optimized the weight:

$$H\big(f(s,a)\big) = H(C) = W.C = \sum_i w_i C_i$$

W.C is the inner product, C_i is the frequency of i_th type of descriptor. W is the weight.

Here, the weight plays an important role. The weight can be calculated from the frequency distributions of native proteins or calculated using optimized methods.

Frequency distribution is based on descriptors which fit to low-energy states (Boltzmann assumption) [23]. It uses solvent terms based on contact potentials between amino acids pairs [24]. Sippl [27] as well as H. Zhou and Y. Zhou [28] considered distance-dependent energy functions for different ranges and their pairwise interactions. Nishikawa and Matsuo added dihedral angles, solvent accessibility and hydrogen-bonding showing how that information can determine the structure of the protein. [29]. Singh and Tropsha considered not just pairwise, but also higher-order interactions [30]. Li and Liang detected three-body interactions [31].

Miyazawa-Jernigan considers potential function and how a pair of residues interacts if their contact is under a particular threshold. The contact potentials are:

- Residue-residue contact potential.
- Residue-Solvent contact potential.
- Solvent-Solvent contact potential.

An advantage of using solvent molecules is that they cannot be detected in X-ray crystal structures [70].

There are 3 types of statistical potentials to calculate the interactions between amino acids:

- Contact potential functions (Miyazawa-Jernigan) [24].
- Distance dependent potential functions [32].
- Geometric potential functions:
    1. Voronoi diagram [33].
    2. Delaunay triangulation [30].
    3. Alpha shape of the protein molecules [34].

Some other analyses are based on optimization. Optimizations are usually maximization of the energy gap between native proteins and decoys with the lowest score. Optimization methods require generation of a large set of decoys. Knowledge-based pairwise potential functions are usually in the form of weighted linear or nonlinear sum of interacting residue pairs. It is the same as the functional form used in statistical potential, where the weight coefficients are derived from database statistics. Several optimization methods have been applied to find the weight vector (w) of linear and nonlinear potential functions [12-14, 17, 35, 36].

Comparing knowledge-based with physics-based energy functions, knowledge-based scoring functions are not time-consuming, cost less and are

more accurate than physics-based scoring functions [37]. However, knowledge-based energy functions have room for error due to noisy data. Opus-psp [38], Opus-ca [39], DFIRE [40], and RW [41] are examples of knowledge-based energy scoring functions.

Knowledge-based energy statistical scoring functions are good in specificity. They may recognize bad decoys because they are trained based on native structures, but the scores are usually not sensitive or accurate.

Nonetheless, knowledge-based scoring functions in some cases contain a lot of noise 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 for calculation. Additionally, due to the fluctuation of protein folding, protein quality prediction using these scoring functions can be unreliable. Because of these limitations, any knowledge-based scoring functions are not dependable. Some of the most popular knowledge-based energy scoring functions are Opus-psp [38], Opus-ca [39], DFIRE [40], and RW [41] which will be described more in detail in the result section.

2.4. Consensus Methods:

Consensus methods are the average of the pair-wise similarity between all the decoys [43] such as 3D-Jury system [42]. Consensus methods use all the information from all the decoys [44]. It compares every decoy with the other decoys [44]. It is simple [42] and does not need computational process. It does not need complicated methods to calculate the energy as in empirical force field

or knowledge-based statistical potentials (see section 2.1). It is actually an optimistic method because it follows the most frequent pattern appearing in the ensemble [45]. Consensus method gives credibility to the majority of the models even though they may be poor [45]. If the model is not included with the majority of the models, that can result in a model choice that is inferior to the best model and excludes it from the best model cluster [45]. The advantages of this method are simple because it doesn't require a lot of information, and powerful because it achieves excellent results in a dataset that contains multiple targets. It performed well in CASP7 and CASP8 [44]. On the other hand, a disadvantage of this method is that it cannot assess single decoys because it is based on voting from sets of decoys. Therefore, using single decoys excludes the consensus-based method [44]. Another disadvantage is that if the best model is not among the most frequent group; then, this method will prefer an inferior model over the best model simply because it shares common features with the others [44]. This is the basis of the consensus method.

2.5. Machine learning based approaches:

Machine-learning methods have performed quite well in CASP8. For example, neural network (NN) and support vector machine (SVM) are based on training and testing data. In some states, some methods scores can be considered as features [46]. Also, Wang, Tegge, and Cheng [47] used 3-D coordinates to extract relative solvent accessibility and secondary structure as features. Machine learning methods are based on trained data with known

targets and tested on unseen data.. Preprocessing the data to meet classification data needs requires some formulation to fit the general idea of classification for model selection data. The training process is based on the GDT, RMSD, or Z-score measurements as a target to rank models. Testing can be then conducted on all decoys for one target. Some methods combine single methods which calculate the score in different perspectives such as MULTICOM-CLUSTER [48] which finds a good reference subset for the ensemble. Then, the other models were compared with a reference subset. QMEANclust [44] uses consensus-based information and the QMEAN method. QMEANclust deletes bad models and then ranks the remaining models based on the consensus-based method.

Chapter 3. Methods and Materials

In this chapter, different kinds of methods are proposed for protein structure prediction quality assessment. The assessment of the methods is the GDT-TS measure. Every method considers or measures the predicted structure from different levels or approaches. Some chosen methods were combined, and the results showed very significant improvement over the existing methods. Combined Method performs better than the state-of-art methods. Good Quality assessment methods correlate with the GDT-TS score over all the predicted structure scores. The inputs for Single Position Specific Probability (SPSP) Score based on 4-mer sequence are sensitive. The inputs were the pseudo-bond angles. These angles are *bending* pseudo-bond angles and *torsion* pseudo-bond angles. Calculation of both types of angles is discussed later. The goal from these inputs is to cluster those angles into 17 clusters as mentioned in the Wei-Moe method [71]. This method is called the 4-mer method because it is based on four residues. It only considers C-alpha positions. It can calculate the angles between every four adjacent residues using sliding window. They are dependent on each other because they share three residues. After having the 4-mer sequence for every predicted structure, model selection methods based on 4-mer sequence can be done. Some of them are based on the consensus approach and others are based on knowledge-based approach. In the next sections, calculating the *bending* pseudo-bond angles, dihedral (*torsion*) pseudo-bond angles, and then extracting the 4-mer sequence, which is based mainly on the states, are addressed. Finally, model selection methods are explained in detail,

which are based on the 4-mer sequence and the combined method between the single scores and the 4-mer sequence based methods.

3.1. Bending pseudo-angles:

3.1.1. Definition:

Bending pseudo-angles are the angles between 3 points in 3-D dimensions. Having three points in an object, A, B, and C, the bending angle will be the angle between the vector AB and the vector BC. Thus, it is called ABC angle.

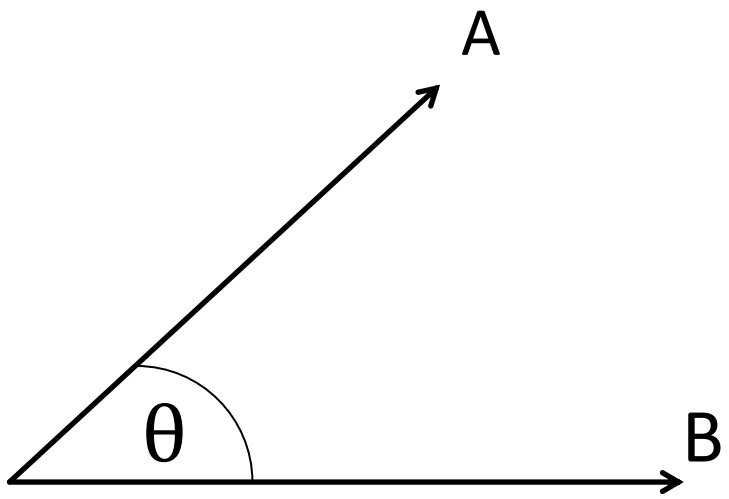

Figure 1. The ABC Angle

3.1.2. Calculating the bending angle:

Calculating the bending angle needs only the coordinates (x, y, z) of three points A, B, and C. First step is to calculate the difference between x1 and x2, y1 and y2, as well as z1 and z2 for the first two points. Then, the same thing must

be done with the second and third point, which calculates the difference between x3 and x2, y3 and y2, and finally, z3 and z2. All these calculations are shown below:

$$dx12 = x1 - x2 \qquad dy12 = y1 - y2 \qquad dz12 = z1 - z2$$

$$dx32 = x3 - x2 \qquad dy32 = y3 - y2 \qquad dz32 = z3 - z2$$

Second, since the differences between the coordinates of A, B and B, C atoms are calculated; it is easy to calculate the magnitudes between the two vectors based on Euclidean distance. As follows:

$$d12 = \sqrt{[(dx12)^2 + (dy12)^2 + (dz12)^2]}$$

$$d32 = \sqrt{[(dx32)^2 + (dy32)^2 + (dz32)^2]}$$

That the third step calculates the angle between the two vectors which is the dot product of the vectors divided by multiplication of the vectors scalars as follows:

$$Cos(\theta) = \frac{dx12 * dx32 + dy12 * dy32 + dz12 * dz32}{d12 * d32}$$

$$\theta = \text{acos}\,(\cos\,(\theta))$$

To convert the angle to radian instead of degree, this formula was used as follows:

$$\text{rad} = \text{deg} \cdot \frac{\pi}{180^\circ}$$

3.2. Dihedral (Torsion) pseudo-angles:

3.2.1. Definition:

Dihedral angle is the angle between two planes. It is the angle between four points. Every three points will represent a plane. Let say K, I, J, L are adjacent points. KIJ represents a plane and IJL represents another plane. The dihedral angle is the angle between KIJ plane and IJL plane. The figure below shows the dihedral angle between the planes.

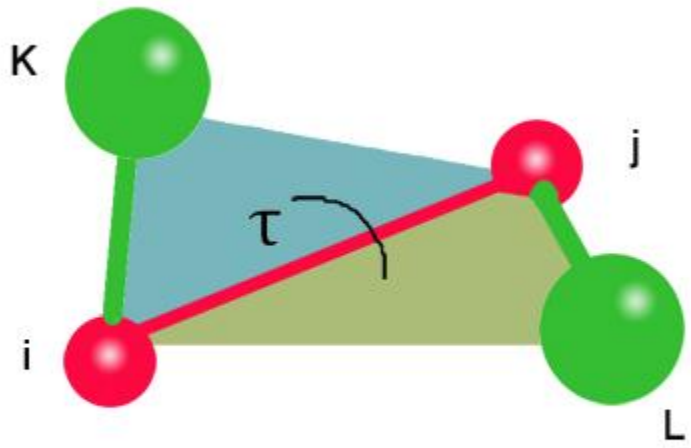

Figure 2. A Dihedral Angle between Two Planes.

3.2.2. Calculating the dihedral pseudo-bond angle:

Calculating a dihedral angle needs the coordinates (x, y, z) for all the four points. The first step was calculating the differences between the coordinates, x2 with x1, x3 with x2, x4 with x3, y2 with y1, y3 with y2, y4 with y3, z2 with z1, z3 with z2, and z4 with z3.

$$dx21 = x2 - x1 \qquad dy21 = y2 - y1 \qquad dz21 = z2 - z1$$

$$dx32 = x3 - x2 \qquad dy32 = y3 - y2 \qquad dz32 = z3 - z2$$

$$dx43 = x4 - x3 \qquad dy43 = y4 - y3 \qquad dz43 = z4 - z3$$

After that, cross products between the three vectors were calculated.

$$dotpx1 = dy21 * dz32 - dy32 * dz21 \qquad dotpy1 = -dx21 * dz32 + dx32 * dz21$$
$$dotpx2 = dy32 * dz43 - dy43 * dz21 \qquad dotpy2 = -dx32 * dz43 + dx43 * dz32$$

$$dotpz1 = dx21 * dy32 - dx32 * dy21$$
$$dotpz2 = dx32 * dy43 - dx43 * dy32$$

Then, the magnitudes were calculated and multiplied together.

$$dp1dp2 = \sqrt{[((dotpx1)^2 + (dotpy1)^2 + (dotpz1)^2) * ((dotpx2)^2 + (dotpy2)^2 + (dotpz2)^2)]}$$

After that, calculating the angle by the dot product method was done as shown in the next formula.

$$dihedral = \text{acos}\left((dotpx1 * dotpx2 + dotpy1 * dotpy2 + dotpz1 * dotpz2)/dp1dp2\right)$$

Dihedral angle range is [Pi, -Pi], so the sign or the direction had to be taken care of. To determine the direction:

$$dotpx3 = dotpy1 * dotpz2 - dotpy2 * dotpz1$$

$$dotpy3 = -dotpx1 * dotpz2 + dotpx2 * dotpz1$$

$$dotpz3 = dotpx1 * dotpy2 - dotpx2 * dotpy1$$

$$if(dotpx3 * dx32 + dotpy3 * dy32 + dotpz3 * dz32) < 0, then\ dihedral = -dihedral$$

The points in the previous definitions for bending angles and dihedral angles are C-alphas. Thus, pseudo-bond angles between the residues are represented by only C-alphas in the backbone

3.3. 4-mer sequence:

3.3.1. Definition:

4-mer means considering only four residues. This method can cluster the pseudo-angles given which are the bending pseudo-angles and dihedral pseudo-angles to 17 clusters. Every cluster will be represented by a letter. These clusters are called states in CLESUM method. CLESUM is based on the mixture model for the angles probability. For every four residues, there are three angles. First is ABC bending angle. Second is ABCD dihedral angle. Third is BCD bending angle. Thus, every C-alpha has 2 angles (θ) and 1 dihedral angle (τ), so $x_k = (\theta_k, \tau_k, \theta_{k+1})$ which is treated as a point x_k. This method plotted this distribution on x, y and z dimensions. They found out that the peaks are 17 peaks which are considered as clusters. Most of these points fall into these 17 clusters.

3.3.2. Extract 4-mer sequence for a given structure

To calculate the probabilities of the angles falling in a state, a mixture model of several normal distributions solved this issue. k is the index of the C-alphas, i is the index of the cluster, μ is the mean, $\sum$ is the symmetric covariance matrix of the normal distribution for the angles of a cluster, and π is a priority parameter for each cluster.

$$\text{for every point } x_k = (\theta_k, \tau_k, \theta_{k+1})$$

$$\text{for every cluster}(C_i)$$

$$\text{caculate}$$

$$P(x_k|C_i) = \pi_i * (2\pi)^{-\frac{3}{2}} * \left|\sum\nolimits_i^{-1}\right|^{-1/2} * \exp[-\frac{1}{2}(x_k - \mu_i)' * \sum\nolimits_i^{-1} * (x_k - \mu_i)]$$

$$\text{end for}$$

$$\text{end for}$$

$$Seq(k) = \arg_i \max(P(x_k|C_i))$$

Some fixed parameters taken directly from their table are as follows:

Table 1. The 17 Structural States Taken from [71]

State	π	$\left\|\sum_i^{-1}\right\|^{-1/2}$	μ			$\sum_i^{-1}$					
			θ	τ	θ'	$\theta\theta$	$\tau\theta$	$\tau\tau$	$\theta'\theta$	$\theta'\tau$	$\theta'\theta'$
I	8.2	1881	1.52	0.83	1.52	275.4	-28.3	84.3	106.9	-46.1	214.4
J	7.3	1797	1.58	1.05	1.55	314.3	-10.3	46	37.8	-70	332.8
H	16.2	10425	1.55	0.88	1.55	706.6	-93.9	245.5	128.9	-171.8	786.1
K	5.9	254	1.48	0.70	1.43	73.8	-13.7	21.5	15.5	-25.3	75.7
F	4.9	105	1.09	-2.72	0.91	24.1	1.9	10.9	-11.2	-8.8	53
E	11.6	109	1.02	-2.98	0.95	34.3	4.2	15.2	-9.3	-22.5	56.8
C	7.5	100	1.01	-1.88	1.14	28	4.1	6.2	2.3	-5.1	69.4
D	5.4	78	0.79	-2.30	1.03	56.2	3.8	4.2	-10.8	-2.1	30.1
A	4.3	203	1.02	-2.00	1.55	30.5	9.1	8.7	6	5.7	228.6
B	3.9	66	1.06	-2.94	1.34	26.9	4.6	4.9	9.5	-5	54.3
G	5.6	133	1.49	2.09	1.05	163.9	0.6	3.8	2	-3.7	32.3
L	5.3	40	1.40	0.75	0.84	43.7	2.5	1.4	-7	-2.9	34.5
M	3.7	144	1.47	1.64	1.44	72.9	2.1	4.8	1.9	-7.9	72.9
N	3.1	74	1.12	0.14	1.49	25.3	3.2	3.1	9.9	0.9	83
O	2.1	247	1.54	-1.89	1.48	170.8	-0.7	3.7	-4.1	3.1	98.7
P	3.2	206	1.24	-2.98	1.49	48	8.2	7.3	-4.9	-6.6	155.6
Q	1.7	25	0.86	-0.37	1.01	28.4	1.5	1.2	3.4	0.1	19.5

3.4 Method

3.4.1. Methods for Protein Structure Predictions Quality Assessment Based on 4-mer Sequence

CLESUM method clusters every four adjacent residues to a state (17 clusters). Given predicted structure information as the bending and torsion angle, CLESUM method gives back a sequence. Every letter in this sequence represents a cluster for four adjacent residues. Thus, a 4-mer sequence for that structure is in hand. From here, model selection methods based on 4-mer sequence were done on this sequence.

3.4.1.1. Single Position Specific Probability Score (SPSP)

This method requires a set of decoys. It is based on a consensus-based approach. The aim from this method is to measure the similarity per position (per column) for all the sequences. For a set of decoys, the first step is calculating the

pseudo-bond angles as dihedral angles, and bending angles as mentioned above. Step 2 is extracting the 4-mer sequences using CLESUM as mentioned earlier. As a result, a set of sequences represents the local regions in the predicted structures. Then Position Specific Probability Matrix (PSPM) was calculated which is a 17 * Sequence length matrix. Step 3 requires calculating the score for every sequence by summing the probability for every column.

Here is calculation of the PSPM score with numbered steps:

1) PSPM (Position Specific Probability Matrix) was calculated based on all the decoys sequences.

$$\mathrm{PSPM}(i,j) = \mathrm{PSPM}(i,j) + 1$$

when

$$\mathrm{Seqs}(k,j) = \mathrm{States}(i)$$

$$\text{where k is the index for the decoys (rows),}$$

$$\text{j is the index of the length of sequence(columns)}$$

$$\text{and i is the index of the states which are 17 states}$$

2) To calculate the probability, it is just PSPM divided by the number of the decoys.

$$\mathrm{PSPM}(i,j) = \frac{\mathrm{PSPM}(i,j)}{\mathrm{DecoysNo.}}$$

3) Calculating the score for every decoy is just summing the rows together.

$$\mathrm{Score}(k) = \sum_{j=1}^{j=\mathrm{length(seq)}} \mathrm{PSPM}(k,j)$$

3.4.1.2. Pair Score:

Calculating the pair scores must have all sequences for decoys for one target. The aim of this score is to calculate the probability for every two adjacent states out from the 17 states. The following step is calculating the pair scores for every 4-mer sequence.

For every target, the following steps must be done:

1- Calculate the PPM (Pairwise Probability Matrix) which is a 17*17 matrix.
2- For every adjacent pair states (letters), add one point in the matrix that matches these two letters. By doing that, the matrix is filled with the frequency.
3- Sum the rows together to calculate the probability.
4- Divide every element in the matrix by the sum of the row to give the probability of every two adjacent 4-mer sequences.
5- Take every two adjacent letters and find out their probability in the matrix; then add it to the score. By the end, this score can be used to rank the models.

A. PPM calculation: for every target, for all the decoys.

$$\mathrm{PPM}(\mathrm{i},\mathrm{j}) = \mathrm{PPM}(\mathrm{i},\mathrm{j}) + 1$$

when:

$$\mathrm{Seq(k,l)} = \mathrm{State(i)}\ \&\&\ Seq(\mathrm{k,l+1}) = \mathrm{State(j)}$$

where k is the index of the decoy, l is the index of the letter in the sequence.

i, j are the indices of the states. They are 17 states.

B. Dividing by the sum of the row gives the probability of a decoy

$$\mathrm{PPM(i,j)} = \mathrm{PPM(i,j)} / \sum_{\mathrm{i=1}}^{\mathrm{i=17}} \mathrm{PPM(i)}$$

C. To calculate the score for every 4-mer sequence (k):

$$\mathrm{Score(k)} = \sum_{\mathrm{l=1}}^{\mathrm{l=length(seq)}} \mathrm{PPM(Seq(k,l), Seq(k,l+1))}$$

3.4.1.3. Sum SPSP and Pair Scores:

For every target, there is a Single Position Specific Probability Score (SPSP) and a pair scores for a decoy. This score must be added together.

$$\mathrm{SumSiPaScore(k)} = \mathrm{SPSP(k)} + \mathrm{PairScore(k)}$$

where k is the index of the decoy

3.4.1.4. P (4-mer Seq Letter|AminoAcidSequence) Scores:

To calculate P (4-mer|AA), the following steps were done to calculate P (4-mer|AA):

First, the pattern was calculated as a pairwise matrix between 17 states and 20 amino acids from more than 10,000 native structures. In other words, the frequency was calculated between the 4-mer sequence and amino acid sequence, resulting in a 20*17 matrix. In this case, only the third amino acid and its pair in the 4-mer sequence were considered because the third amino acid actually represents the dihedral angle and shares between the two bending angles, so it is the most important residue. The aim from this method is to calculate what the probability between the third amino acid and the 4-mer state is.

$$P4GivProSeq(j,i) = P4GivProSeq(j,i) + 1$$

when:

$$4merSeq(k) = State(i)\ \&\&\ ProSeq(k+2) = ProSeqState(j)$$

$$where\ k\ is\ the\ index\ of\ the\ letter\ in\ the\ sequence.$$

$$i\ is\ the\ index\ of\ 4-mer\ state\ which\ is\ up\ to\ 17$$

$$j\ is\ the\ index\ of\ Amino\ Acids\ sequences. We\ have\ 20\ letters.$$

Summing the row (amino acid letters) helps in calculating the probability (4-mer|AA).

$$P\left(4merSeq_{(i)} \middle| AASeq_{(j)}\right) = \frac{P\left(4merSeq_{(i)}\right) \cap P\left(AASeq_{(j)}\right)}{P\left(AASeq_{(j)}\right)}$$

3.4.1.5. P (AminoAcidSequence|4-merSeq):

Summing the columns (4-mer letters) helps to calculate the probability (AAseq|4merSeq)

$$P(AASeq_{(j)}|4merSeq_{(i)}) = \frac{P(AASeq_{(j)}) \cap P(4merSeq_{(i)})}{P(4merSeq_{(i)})}$$

$$where\ i\ is\ the\ index\ of\ states(1{:}\ 17)$$

$$and\ j\ is\ the\ index\ of\ amino\ acids\ (1{:}\ 20)$$

After getting the general matrices from native structures, these matrices applied on the decoys as follows:

$$Score(k) = Score(k) + P(AASeq_{(j)}|4merSeq_{(i)})$$

when

$$AASeq_{(j)} = ProSeqState(j)\ \&\&\ 4merSeq_{(i)} = State(i)$$

$$where\ k\ is\ the\ index\ of\ decoys,$$

$$i\ is\ the\ index\ of\ states,$$

$$and\ j\ is\ the\ index\ of\ the\ amino\ acids$$

3.4.1.6. Refined Single Position Specific Probability Score (SPSP):

After looking over the Single Position Specific Probability (SPSP) Score results, it was determined to be the best method over all the 4-mer methods. Modifying this score by Refined Single Position Specific Probability (RSPSP)

Score which assigns 0 for the states that they have less frequency than 20 per column (position) might give a better performance. The aim from this method was to remove the bad decoys as expected based on 4-mer sequences which represent basically local structure.

$$RSPSP(i,j) = 0$$

when

$$RSPSP(i,j) < 20$$

After ignoring the models that have less than 20 frequencies, the probability and the scores were calculated as explained in the Single Position Specific Probability (SPSP) Score method was done.

3.4.1.7. P (4-merseq|Secondary Sequence):

This method has to have secondary sequence for every decoy. The goal was to get the frequency of what the 4-mer state or letters represents in secondary structure. Because every 4-mer sequence represents four amino acids, four secondary sequence letters represent one state. It is also important to note that a secondary sequence has three main structures: coil(C), helix (H), and beta sheet(S). First step was to extract the pattern from native structures to fill a 17*81 matrix between 4-mer sequence and secondary sequence. 17 represents the states for 4mer sequence, while 81 represents four combinations of the (H, C, S) secondary sequence which are 3^4. Then, applying this matrix on the decoys assesses the decoy based on this method.

To fill out the matrix from the natives the following formula was calculated as follows:

$$4merSS(i,j) = 4merSS(i,j) + 1$$

when:

$$4merSeq(j) \;=\; 4merSS(j)\;\&\&\; SecSeq(k{:}\,k+3) \;=\; 4merSS(i)$$

$$where\ i\ is\ the\ index\ of\ the\ SS\ combinations(81)(rows)$$

$$where\ j\ is\ the\ index\ of\ the\ states\ (17)(Columns)$$

$$where\ k\ is\ the\ index\ in\ the\ secondary\ sequence\ in\ the\ natives$$

Step 2 used this matrix to apply it on the decoys.. The previous condition is also applied to the score calculation.

$$Score(k) = \sum_{l=1}^{l=length(seq)} 4merSS(i,j)$$

$$where\ k\ is\ the\ index\ of\ decoy$$

$$where\ i\ is\ the\ index\ of\ the\ SS\ combinations(81)(rows)$$

$$where\ j\ is\ the\ index\ of\ the\ states\ (17)(Columns)$$

3.4.1.8. Consensus 4-mer Sequence (CombinedMethod for Single and Pair Scores):

Consensus sequence is the sequence that appears most of the time. This method was calculated based on adjacent letters and their frequency in all

decoys, which is a small segment from two letters. Thus, calculating the probability for every letter based on that position and the following position (sliding window method) is the main idea behind this method. The goal was to find the consensus 4-mer sequence and compare it with the decoys to assign a score for every decoy. This method has to have all the 4-mer sequences.

First, for every two adjacent letters, a 17*17 matrix was calculated which represents two adjacent letters in the sequence for all the decoys and shows how diverse they are. After that, the biggest number was found in the matrix which represents the frequency of two adjacent pairs. The two adjacent letters were the ones with the highest frequency found which represents a local 2-letter consensus region. The same thing was done for the other segments consisting of two letters. For example, first letter is with the second; the second is with third, and so on.

For every two adjacent letters, calculating the PPM (Pair Probability Matrix) splits the sequence into segments. Every time, the most frequent two adjacent letters are found and take only the first letter to represent the position of a common or consensus letter. After extracting the consensus sequence for the whole length, comparing it to all the decoy sequences gives an idea of how close the decoys are to the consensus.

$$CPM(i,j) = CPM(i,j) + 1$$

when

$$AllSeqs(k,b) = States(i) \&\& \; AllSeqs(k,b+1) = State(j)$$

$$where\ k\ is\ the\ index\ of\ the\ decoys;$$

$$i, j\ are\ the\ indices\ of\ the\ states\ which\ are\ 17\ states$$

$$and\ b\ is\ the\ index\ of\ the\ length\ of\ decoy\ 4mer\ sequence$$

To find the corresponding letters for two-letter regions, the maximum value must be found. Those letters will be the local consensus segments in particular positions. This step was done for all sequences to extract the consensus 4mer sequence overall.

Comparing the consensus sequence with every decoy sequence gave a score to every decoy

$$Score(k) = Score(k) + 1$$

when

$$Seq(j) = con.Seq(j)$$

$$where\ k\ is\ the\ index\ of\ the\ decoys$$

$$j\ is\ the\ index\ of\ letter\ in\ the\ sequence$$

3.4.2. Combining Method:

Looking at the results of all the previous methods led to combining all the scores together with a resultant performance which was better than previous

state-of-the-art methods. The computational process for the combination method gave a hint of the value of combining some methods not previously combined as possible ways to improve the process. All combinations of the scores have been tried to find the best combination methods. The best combination (shown in the following sections) costs less and shows a better performance. The following flowchart explains the general idea of CombinedMethod.

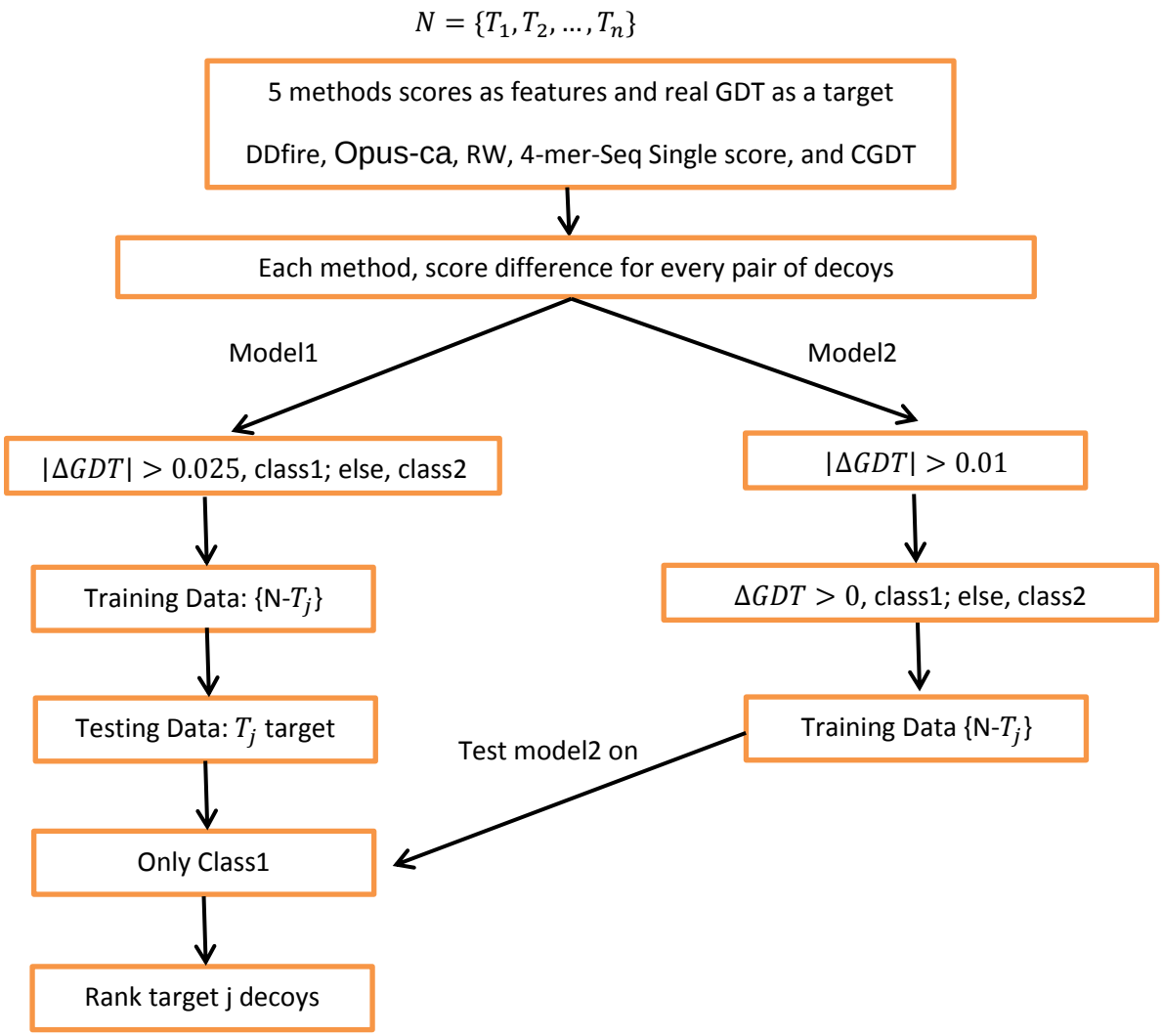

Figure 3 A flowchart of CombinedMethod. It shows the process for target T_j. This process was repeated for all the targets using the Leave-one-out method. In both training Datasets, 1000 records were randomly sampled from every target T_i except T_j because it is the testing target.

3.4.2.1. Preprocessing the Data:

For a better trained model, the observations that have GDT difference less than 0.01 were removed before randomly generating the training data–a step that addressed a previous problem with redundancy. Since the real GDT score was not used in the testing data, a model was trained to distinguish the observations (records) that have a value difference between their real GDT score of less than 0.025. Records showing less than 0.025 were put in the second class; otherwise, they are in the first class. Thus, for every target, training and testing data were generated using the leave-one-out method. A total of 1000 records (the differences of the scores for all the methods) were randomly picked from N-1 targets to generate the training data. Then, the model was tested on the last target. From the results of the testing data, the records that were expected to have a GDT difference higher than 0.025 were considered for the next model as a testing data. This new testing data was used for the testing process mentioned in the next step.

3.4.2.2. Combined Score:

Combination method combined single scores (Opus-ca, RW, DDfire, and CGDT) with consensus score (Single Position Specific Probability (SPSP) Score) using neural network method in Weka tool [72]. For generating the data, the differences for every decoy against the others were calculated for every method as follows

$$S^{ij}{}_{(k)} = S^{i}{}_{(k)} - S^{j}{}_{(k)}$$

where k is the index of the methods,

i is the index of a decoy, j is the index of the other decoys.

As features, the decoy difference scores were used in every method. The real GDT differences for the decoys were used as a target for the trained model. Then, the records were classified into two classes. If the real GDT difference is positive, then it goes to the first class; otherwise, it is in the second class. For training process, the leave-one-out method was used. For training data, 1000 records were picked randomly from N-1 targets. Neural network parameters were set, i.e., rotations = 1000, learning rate = 0.05 and momentum = 0.1. After training the model, the model was tested on the remaining target that had not taken anything from for the training purpose. This test data is the new test data that was mentioned in the preprocessing step.

After testing the model on testing data, the prediction results were processed. If it is 2, then the second decoy is better than the first one; otherwise, the first one is better since every observation consists of a pair of two decoys. Based on the prediction, one point was added to the decoy that is better than the other and. Then, the decoys were ranked based on the new combined score.

CombinedMethod using linear regression was used to improve the performance over the classification methods. Unlike a neural network, classifying the points or the records into classes was not needed. The score on an as is basis was enough to run the regression methods. For the target in neural

network, it is only 1 or 2, but in the linear regression, just the GDT difference as it is was taken without classifications. Other steps are the same.

3.5. Datasets

The methods were applied on two benchmarks produced by different model generation methods. Benchmark 1 was from Zhang's lab, generated by the I-TASSER ab initio modeling tool [69] containing 56 proteins. The other one, benchmark 2, was generated by the Robetta server or Rosetta [67, 68], containing 35 CASP8 proteins. Each protein in both benchmarks had hundreds of decoys. Figure 2 shows the maximum, average and minimum GDT-TS score for models of each protein using both benchmarks. The best of model for each protein had a GDT-TS score greater than 0.4, which ensured that the pool contained some reasonably good models.

3.5.1. Yang Zhang's Data

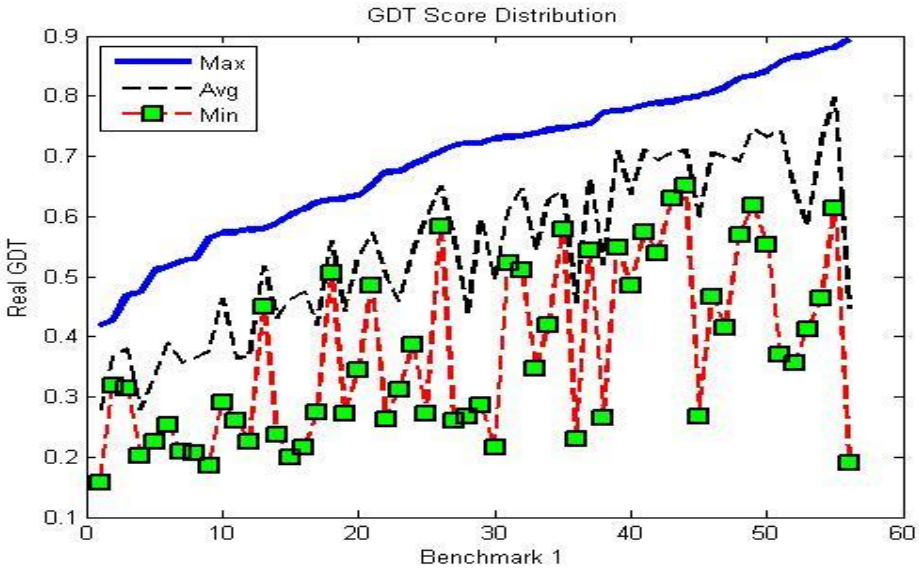

Fig. 4. Y-axis Shown Here as the Real GDT-TS Score to the Native Structure. X-axis is the proteins of Yang Zhang's data sorted by maximum of the GDT-TS score.

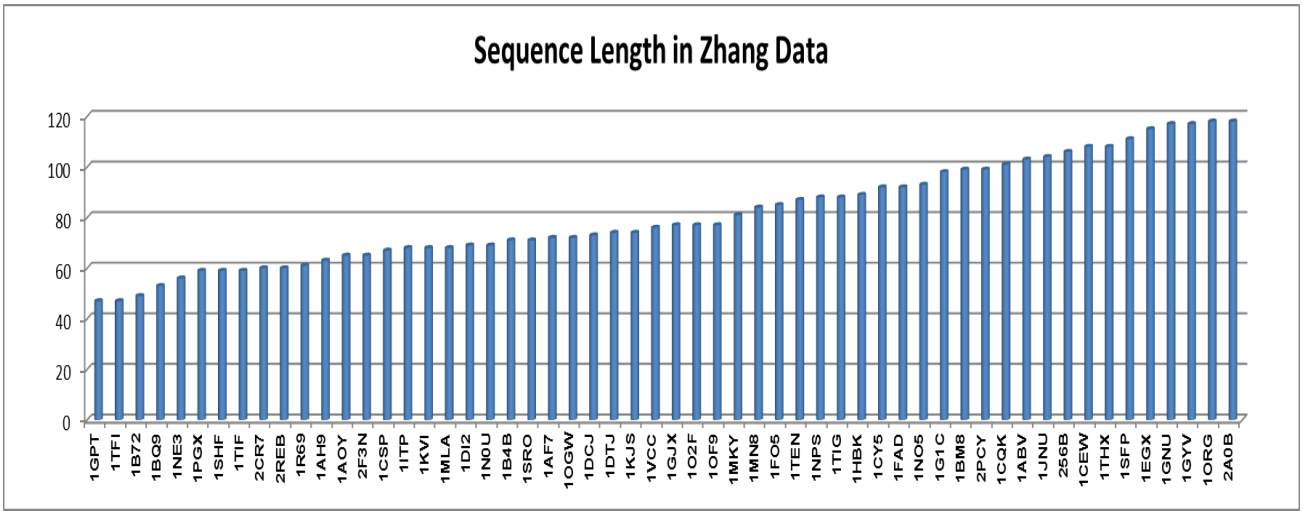

Figure 5. Comparison between Sequence Lengths in the 56 Targets in Yang Zhang's Data.

3.5.2. Rosetta Data

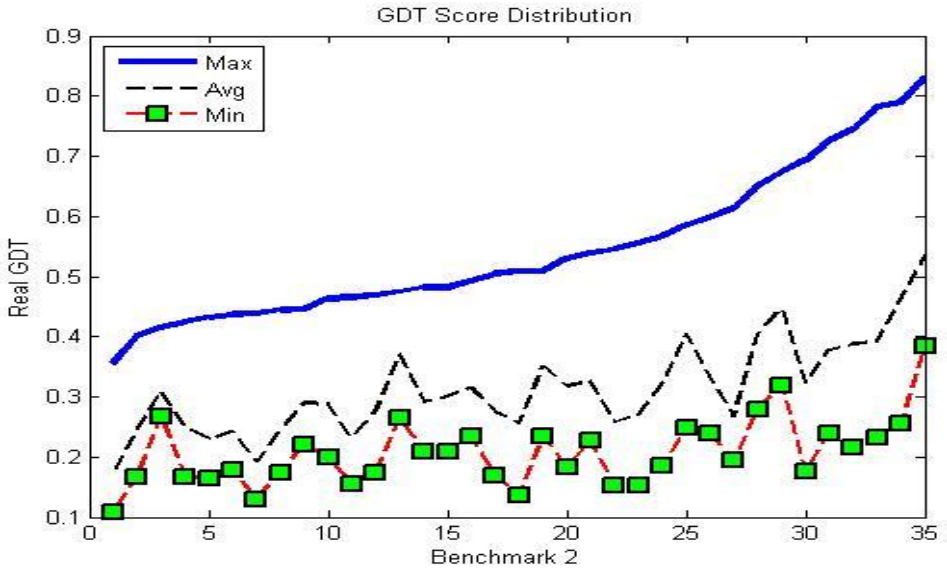

Fig. 6. Y-axis is the Real GDT-TS Score to the Native Structure. X-axis is the Proteins of Rosetta Data Sorted by Maximum of the GDT-TS Score.

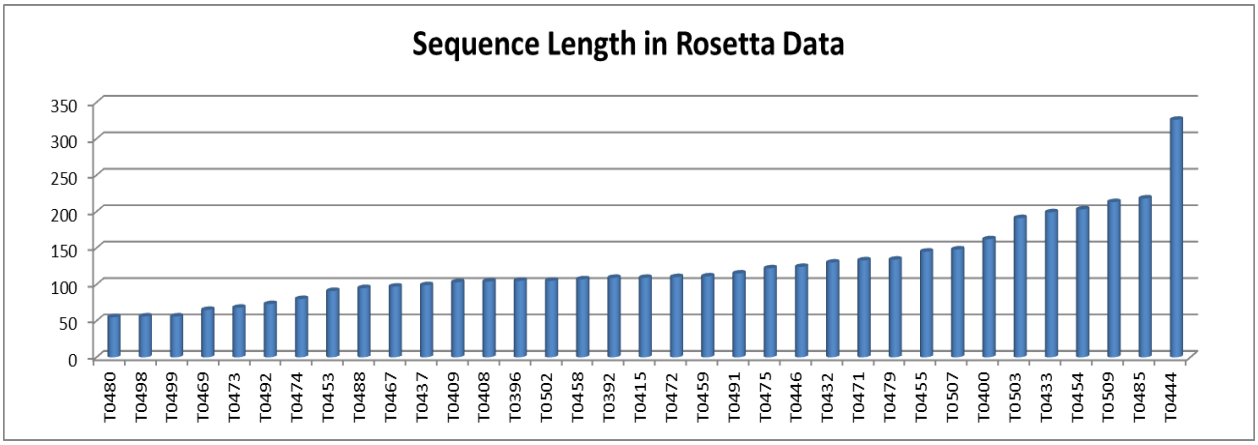

Figure 7. Comparison between Sequence Lengths in the 35 Targets in Rosetta Data.

Yang Zhang's data had a small range between 47 to 118 residues. It had more targets than Rosetta. On the other hand, Rosetta data had a wider range between 55 to 326 residues in just 35 targets. Rosetta Data is harder than

Zhang's data in terms of sequence length. Also, the average GDT-TS score in Rosetta data is less than the average in Zhang's data. There are other factors besides sequence length and the average GDT-TS score.

3.5.3. Target Difficulty

The targets were classified into three categories. First, hard targets have GDT-TS scores less than 0.4. Second, medium targets have a GDT-TS score between 0.4 and 0.6. Third, easy targets have a GDT-TS score more than 0.6. Those cutoffs were used on Zhang's and Rosetta data.

Table 2: Distribution of 56 Targets in Zhang's Data Based on Cutoff Thresholds

Category	GDT-TS	No. of Targets
Hard	[0, 0.4]	11
Medium	[0.4, 0.6]	23
Easy	[0.6, 1]	22

Table 3: Distribution of 35 Targets in Rosetta Data Based on Cutoff Thresholds

Category	GDT-TS	No. of Targets
Hard	[0, 0.4]	30
Medium	[0.4, 0.6]	5
Easy	[0.6, 1]	0

Chapter 4. Results and Performance

In the test, each score was used to rank the models of a given protein. Four different methods were used to compare the performance of each scoring method. Table 1 compared the methods, 4-mer sequence Single Position Specific Probability (SPSP) Score and CombinedMethod, with the single scores, RW, DDfire, and Opus-ca, and also, the CGDT, and the real GDT. The comparison between the methods was in terms of the first selection of the model "GDT1", average Top5 selections of the models, the Top5 selection, and the Pearson and Spearman correlations between the real GDT and the score of a method. The methods were done on datasets, 56 targets in Zhang's data and 35 targets in the Rosetta data. Table 4 shows how the CombinedMethod can give a better performance on all the state-of-the-art methods including CGDT. On the other hand, 4-mer sequence Single Position Specific Probability (SPSP) Score does not really give good results by itself, but it helps significantly the combination method performance. Taking this method out of the CombinedMethod will not make CombinedMethod better than CGDT. From here, Single Position Specific Probability (SPSP) Score of 4-mer sequence is a complementary of single scores in the model selection problem.

Table 4. Comparison of Methods and their Performance Overall on Yang Zhang's Data

Score	Top1	bestTop5	AvgTop5
GDT	0.6946	0.6946	0.6767
CGDT	0.6058	0.6280	0.6039
Dfire	0.6010	0.6398	0.5904
DDFire	0.6006	0.6387	0.5906
Opus-ca	0.5959	0.6367	0.5925
RW	0.5954	0.6381	0.5879
SPSP	0.5847	0.6161	0.5734
Given4merSeq	0.5572	0.6191	0.5561
GivenProSeq	0.5634	0.6143	0.5587
RefinedSPSP	0.5764	0.6191	0.5736
SecondarySeq	0.5322	0.5997	0.5426
Cons.Seq	0.5795	0.6177	0.5707
CombinedMethod	0.6105	0.6309	0.6056
Pair Score	0.5531	0.6123	0.5575
SumOfPaSPSPscore	0.5750	0.6203	0.5712

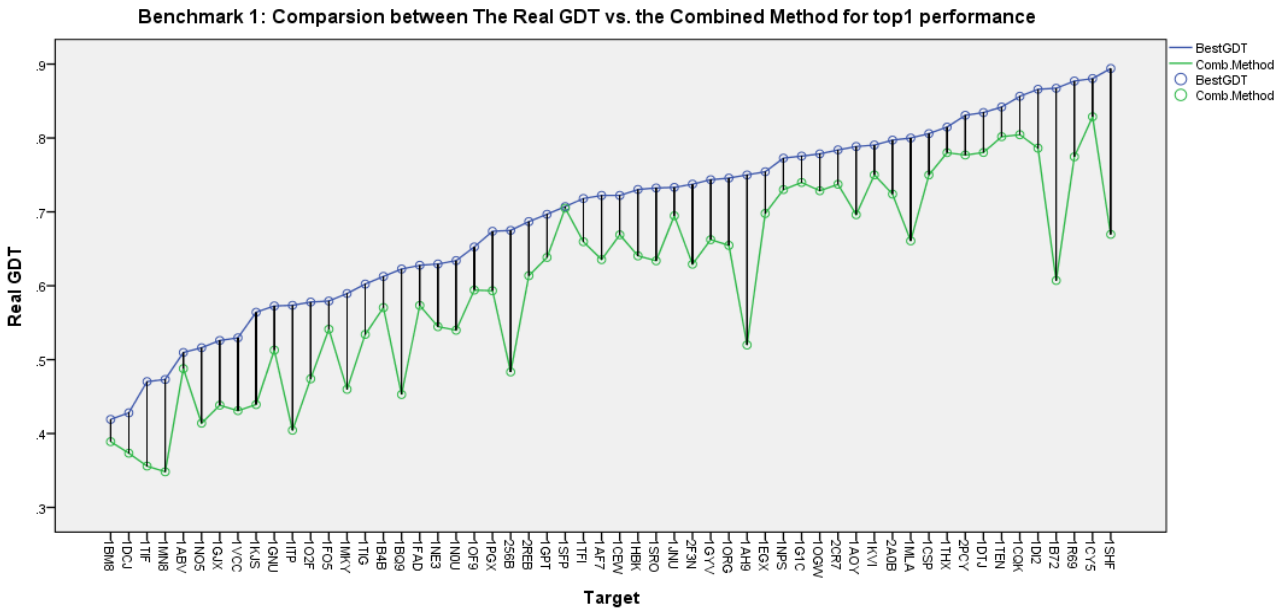

Figure 8. Performance in Yang Zhang's Data of Combined Score (Green) vs. Best GDT (Blue) on y-axis and 56 Targets on x-axis for Top1 Selection.

Table 5: Comparison of methods for the average Z-score on Yang Zhang's Data

Score	Top1	Best Top5	Avg Top5
GDT	2.4372	2.4372	2.1133
CGDT	0.8951	1.2804	0.8501
Dfire	0.8055	1.4693	0.6414
DDFire	0.7902	1.4367	0.6381
Opus-ca	0.7348	1.3505	0.6618
RW	0.6831	1.4305	0.6084
SPSP	0.5015	1.0157	0.3860
Given4merSeq	0.3849	1.1138	0.2468
GivenProSeq	0.2026	1.1471	0.1524
RefinedSPSP	0.4617	1.0669	0.3929
SecondarySeq	-0.0855	0.9633	0.0420
Cons.Seq	0.4660	1.0807	0.3437
CombinedMethod	0.9660	1.3224	0.8672
Pair Score	0.2323	1.1206	0.2666
SumOfPaSPSPscore	0.4007	1.1479	0.3657

Table 6: Comparison of methods and their performance overall on Rosetta Data

Score	Top1	Best Top5	Avg Top5
GDT	0.5449	0.5449	0.5219
CGDT	0.4255	0.4622	0.4060
DDFire	0.3901	0.4666	0.3788
Opus-ca	0.3763	0.4551	0.3663
RW	0.3662	0.4567	0.3696
SPSP	0.3435	0.4173	0.3534
CombinedAllAboveScores(NN)	0.4529	0.4796	0.4309
CombinedMethod(Lin.Reg.)	0.4477	0.4763	0.4212
Pair Score	0.3214	0.3796	0.3294
SumPaSiScores	0.3449	0.3928	0.3407
GivenProSeq	0.3242	0.3917	0.3215
Given4-merSeq	0.3127	0.3636	0.3116
ConsSeq	0.3489	0.4023	0.3482
RefinedSPSP	0.3426	0.4173	0.3534

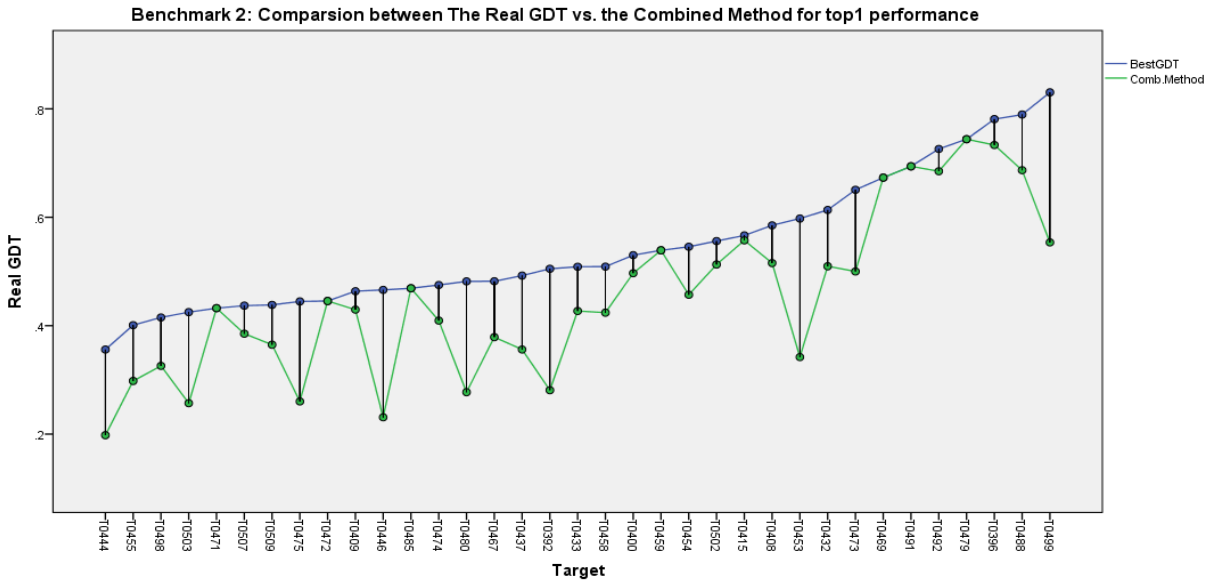

Figure 9. Performance in Rosetta Data of Combined Score (Green) vs. Best GDT (Blue) on y-axis and 35 Targets on x-axis for Top1 Selection.

Table 7: Comparison of methods for the average Z-score on Rosetta Data

Score	Top1	Best Top5	Avg Top5
GDT	4.3545	4.3545	3.8657
CGDT	1.7480	2.5113	1.4647
DDFire	1.2535	2.8060	1.0614
Opus-ca	1.2360	2.5751	0.8730
RW	0.8735	2.7475	1.0064
SPSP	0.3866	1.7615	0.5441
CombinedAllAboveScores(NN)	2.3241	2.9434	1.9968
CombinedMethod(Lin.Reg.)	2.3427	2.8545	1.7890
Pair Score	0.0190	1.1338	0.2102
SumPaSiScores	0.4285	1.3777	0.3711
GivenProSeq	0.0156	0.9296	-0.0179
Given4-merSeq	0.0934	1.3384	0.1148
ConsSeq	0.4128	1.3670	0.4187
RefinedSPSP	0.3649	1.7615	0.5436

Table 8: Comparison of methods and their performance overall on Zhang's Data

	Benchmark 1			
	GDT1	avgGDT5	Pearson	Spearman
GDT_TS	0.6946	0.6767	1	1
Opus-ca	0.5959	0.5925	0.5105	0.4156
CGDT	0.6058	0.6039	0.6969	0.5845
Ddfire	0.6006	0.5906	0.5266	0.4403
RW	0.5954	0.5879	0.4899	0.4172
SPSP	0.5847	0.5734	0.4312	0.3299
CmbindMthd	0.6105	0.6056	0.7112	0.6011

"GDT1" is the average GDT_TS score of Top1 model; "avgGDT5" is the average of 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 the real GDT_TS score.

Table 9: Comparison of methods and their performance overall on Rosetta Data

	Benchmark 2			
	GDT1	avgGDT5	Pearson	Spearman
GDT_TS	0.5449	0.5219	1	1
Opus-ca	0.3763	0.3663	0.2961	0.2739
CGDT	0.4255	0.4060	0.5274	0.5584
Ddfire	0.3901	0.3788	0.3108	0.2722
RW	0.3662	0.3696	0.2983	0.2766
SPSP	0.3435	0.3534	0.2304	0.2462
CmbindMthd	0.4529	0.4309	0.5601	0.5615

"GDT1" is the average GDT_TS score of Top1 model; "avgGDT5" is the average of the mean GDT_TS score of Top5 models. "Pearson" indicates the Pearson

correlation to real GDT_TS and "Spearman" is the Spearman correlation to real GDT_TS score.

After designing an adequate number of methods and calculating the results on two datasets, it was time to analyze them and compare the overall results. In the next graphs, comparisons between couples of methods were viewed and the real GDT score in every graph. It would be difficult to understand the performance for every method if they were all plotted in one graph. For this reason, the graphs have a variety of methods of performance. The graphs considered model selections for Top1 and the best one of the Top5 and the average 5 for both benchmarks.

4.1. Performance on Zhang's Data

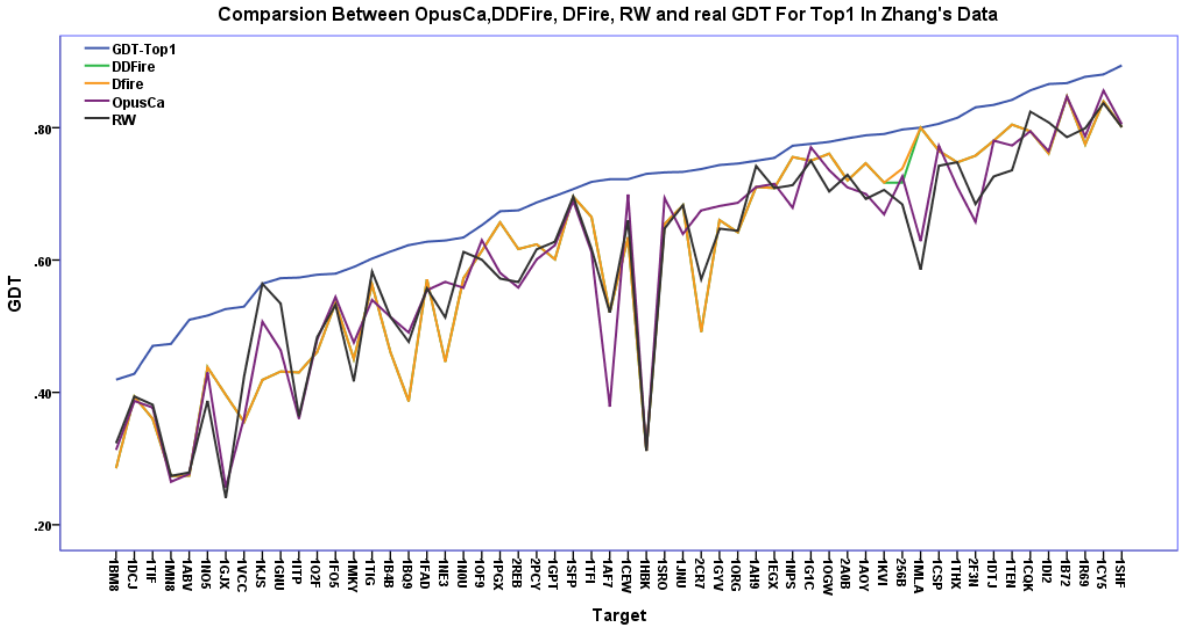

Figure 10. Performance in Zhang's Data of DDFire, DFire, RW, Opus-ca, and Best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Top1 Selection.

In the above graph, DDfire and Dfire have almost the same curve except in few a targets such as 256B which explains the reason for not seeing the green curve in DDfire results. Even in the overall results, Dfire and DDFire performed the best among the plotted methods such as 1PGX target, but they (Dfire and DDFire) showed the worst results in some targets such as 1B09, 2CR7, 1NE3, and 1KJS. DDfire and Dfire predicted the best decoy in 1MLA target. Meanwhile, Opus-ca and RW did not do a good prediction in this target. Even though Opus-ca performed the third best in overall results among all four methods, it ranked the best in some targets such as 1G1C, 1CEW, and 1SRO. However, Opus-ca also performed the worst in some targets such as 1AF7. The last method, RW, had interesting results performing the worst in some cases such as 1TEN and

1MLA but predicting the best decoy in the 1KJS target. Some methods shared the same score in some targets such as the 1HBK target.

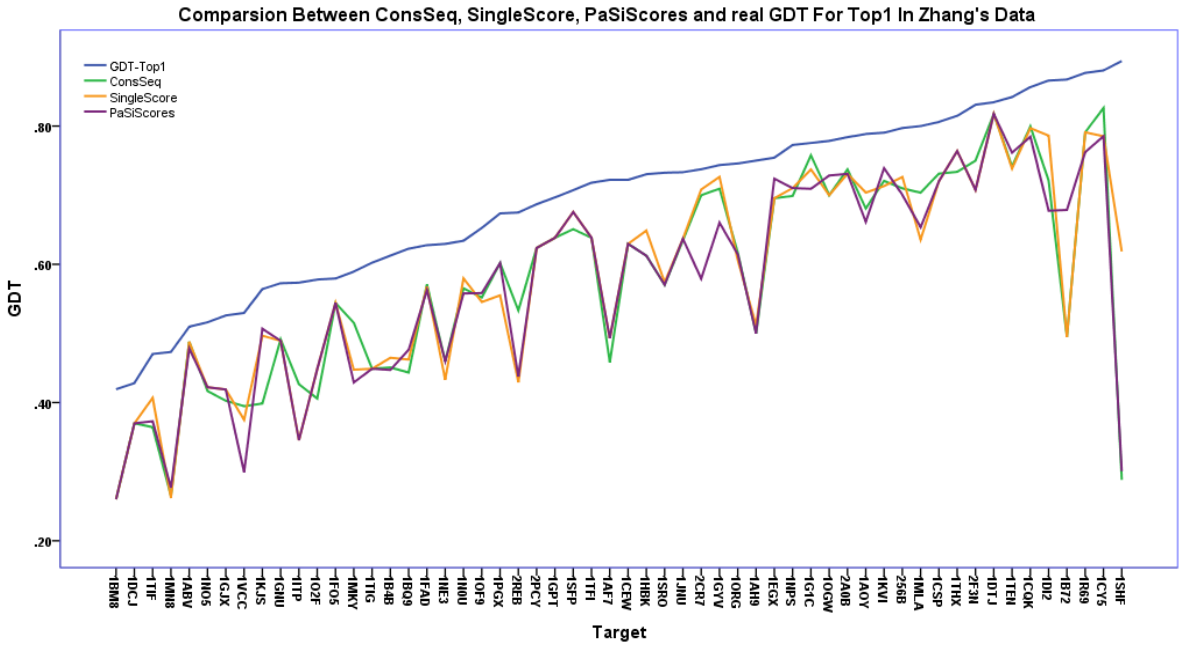

Figure 11 Performance in Yang Zhang's Data of ConsSeq, SPSP, PaSiScores, and Best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Top1 Selection.

In the above graph, the comparison is between the 4-mer methods. The overall results show that Single Position Specific Probability (SPSP) Score was the best, then ConsSeq, and finally PaSiScore. Single Position Specific Probability (SPSP) Score performed the best in target 1TIF, 1DI2, and 1SHF. On the other hand, it performed the worst in some targets such as 1PGX. ConsSeq method performed the best in 1MKY, 2REB, 1CY5, and 1MLA. However, it performed the worst in some targets such as 1AF7. PaSiScore performed the best in 1SFP, 1EGX, and 1B72 even though it showed the lowest results overall. PaSiScore performed the worst in some targets such as 1VCC, 2CR7, and

1GYV. The plot also shows where some methods shared the same score for some targets such as 1AH9, and 1CEW.

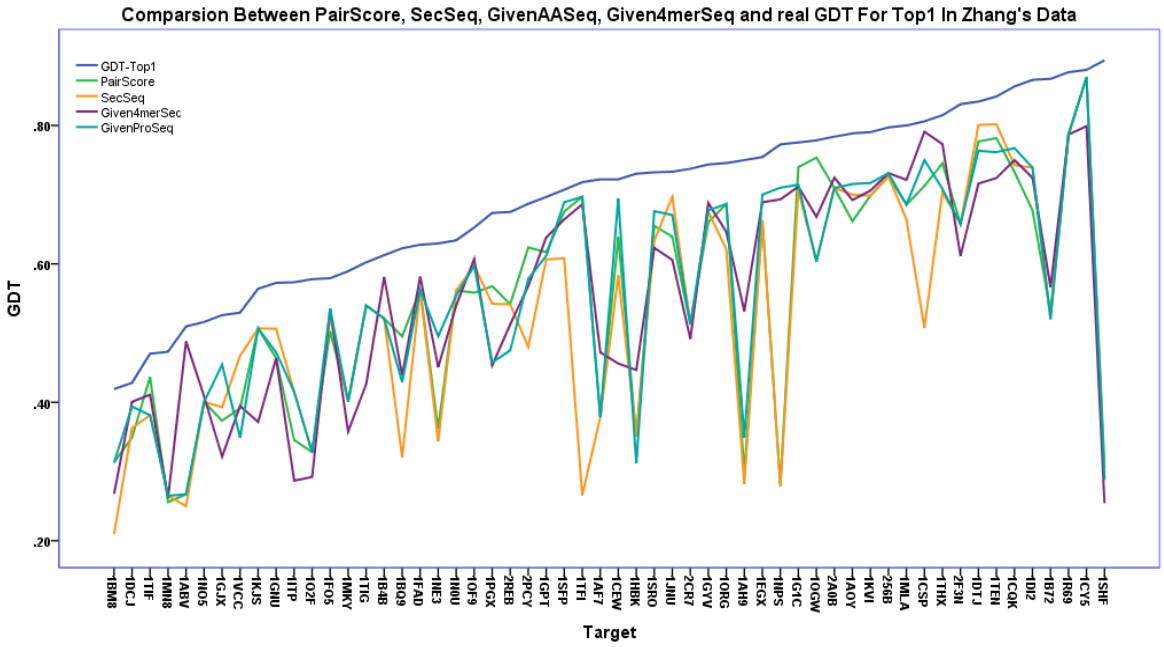

Figure 12. Performance in Yang Zhang's Data of PairScore, SecSeq, GivenAASeq, Given4merSeq, and Best GDT (blue) on y-axis and the 56 Targets on the x-axis for Top1 Selection.

All the methods in the above graph are 4mer methods. In the overall results, GivenProSeq performed the best, then Given4mer, Pair Score, and the last is Secondary Sequence method. GivenProSeq performed the best in some targets such as 1CEW. However, it performed the worst in some targets such as 1HBK and 2REB. Given4merSeq method performed the best in some targets such as 1ABV, and 1CSP targets. It also performed the worst in some targets such as 1ITP, 2F3N, and 1GJX targets. Pair score performed the best in some targets such as 1OGW and 2PCY targets even though it did not perform well in some cases such as 1DI2 and 1AOY. Secondary sequence method performed

the best in some targets such as 1VCC and 1GNU targets. However, it performed the worst in a lot of targets such as 1TFI, 1B09, and 1CSP targets.

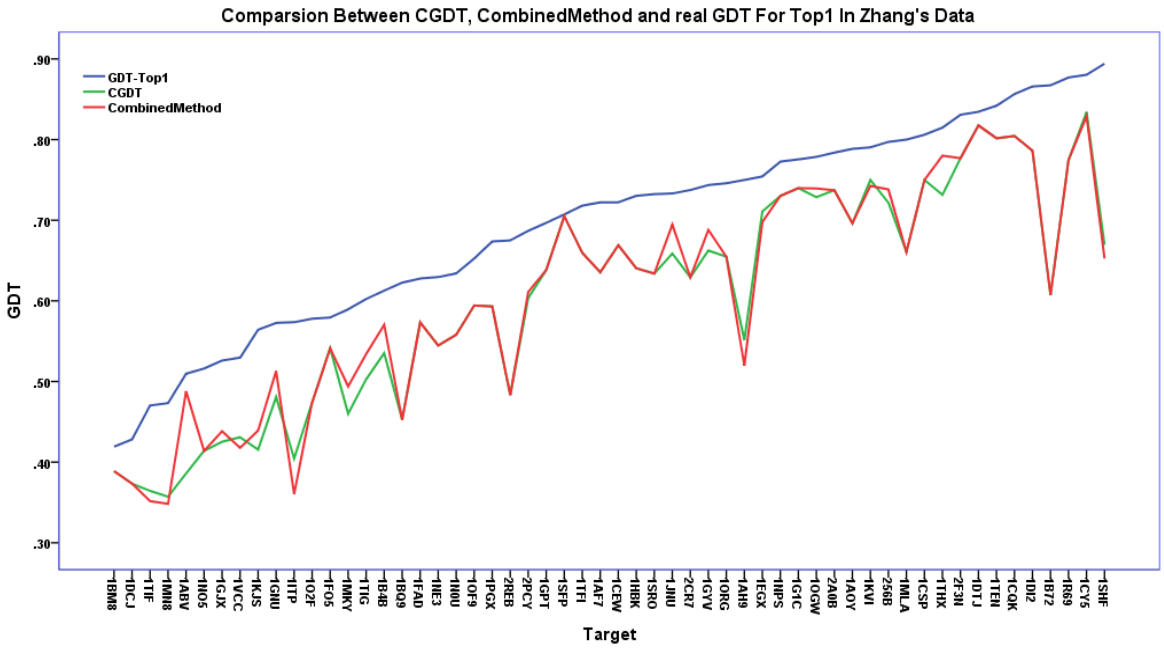

Figure 13. Performance in Yang Zhang's Data of CGDT, CombinedMethod and Best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Top1 Selection.

CombinedMethod performed better than CGDT in the overall results. In the plot, CombinedMethod performed better in some targets such as 1ABV and 1THX targets. However, both methods shared the same score in many targets because the model was trying to maximize the score among a set of methods, and because the CGDT performed the best after CombinedMethod ranking on all methods. Now, CombinedMethod learned most likely from CGDT. From the plot, CGDT performed a little bit better than CombinedMethod in few targets such as 1TIF and 1ITP because the model had some error.

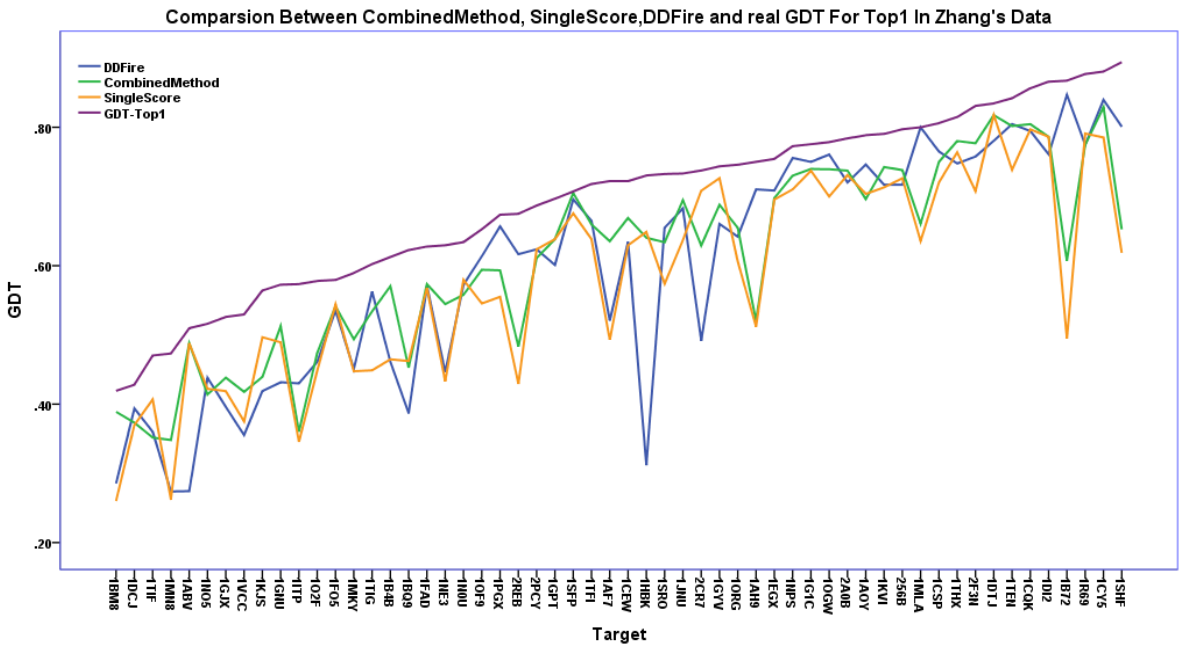

Figure 14. Performance in Yang Zhang's Data of DDFire, CombinedMethod, SPSP, and Best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Top1 Selection.

The above graph has three different methods from different aspects. In terms of the overall results, CombinedMethod was better than DDFire and Single Position Specific Probability (SPSP) Score. CombinedMethod performed the best in some targets such as 1B4B, 1NE3, 1BM8 and 1AF7. It did not perform the worst case in any target which was expected. DDFire performed the best in some cases such as 1PGX, 2REB, and 1MLA. However, it performed the worst in some other cases such as 1HBK, 1ABV, and 2CR7. Single Position Specific Probability (SPSP) Score performed the best in some targets such as 1KJS, 2CR7, and 1GYV. However, it performed the worst in some cases such as 2REB, 1TIG, and 1B72.

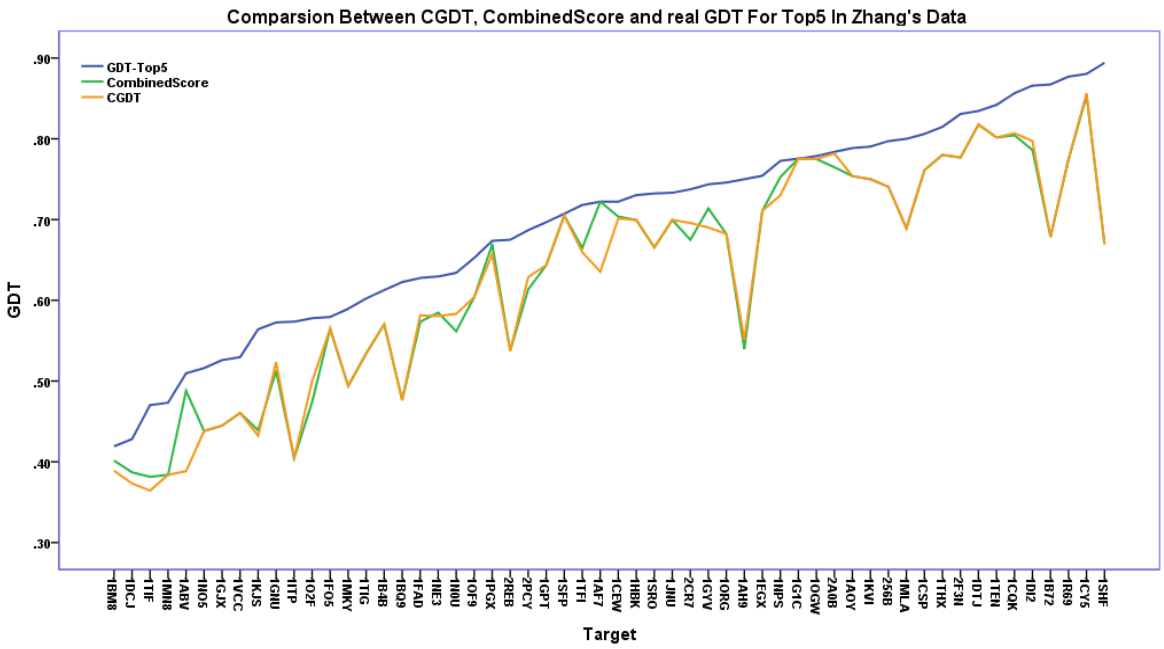

Figure 15.. Performance in Yang Zhang's Data of CGDT, Combined Score, and best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Top5 Selection.

In the overall results, combined score performs better than CGDT. In the above figure, CombinedMethod performed the best in some targets such as 1ABV, 1AF7. CGDT performed better in some targets such as 2A0B. However, both methods have similar scores in most targets which explain the reason for not seeing the green curve. It was expected to have similar results for these methods especially in the Top5 selection due to the fact that CGDT performs better than other methods except for the CombinedMethod, and the CombinedMethod performed better than the state-of-art methods.

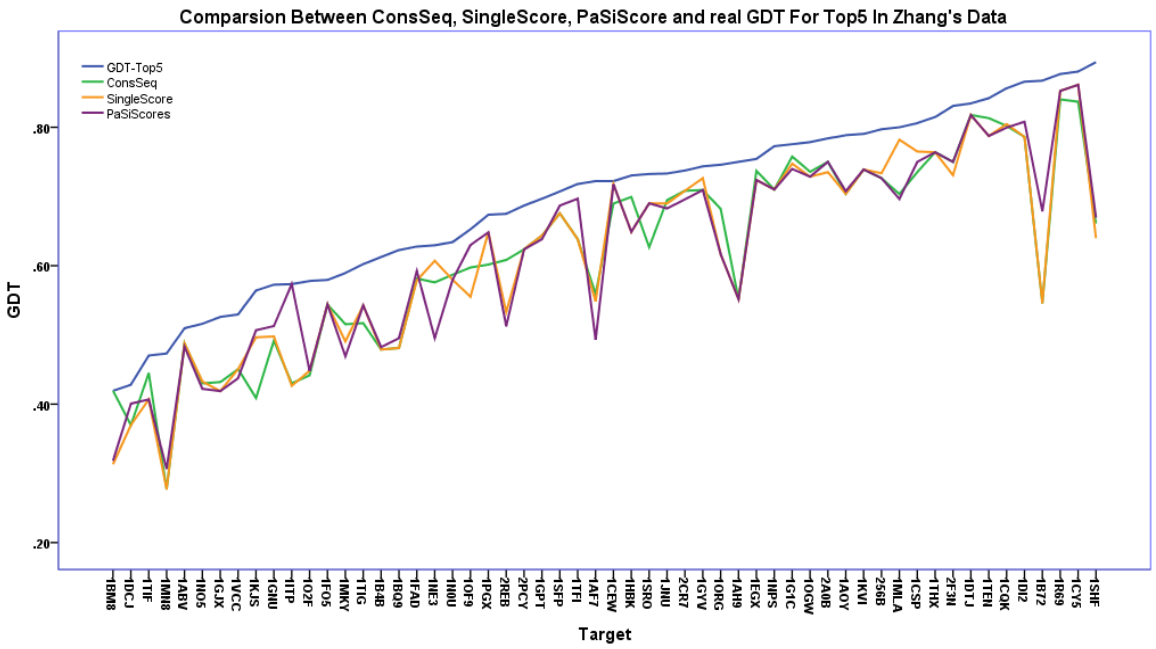

Figure 16. Performance in Yang Zhang's Data of ConsSeq, SPSP, PaSiScore, and Best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Top5 Selection.

In the Top5, PaSiScore performs the best; then ConsSeq; and the last is Single Position Specific Probability Score (SPSP) in the overall results. PaSiScore performed the best in 1ITP, 1BM8, and 1OF9 targets. However, it performed the lowest in some targets such as 1NE3 and 1AF7. ConsSeq performed the best in some targets such as 1BM8, 1TIF and 2REB. However, it was the worst in some cases such as 1KJS and 1SRO. Single Position Specific Probability (SPSP) Score performed the best in some targets such as 1NE3 and 1MLA. However, it performed the worst in some cases such as 1MN8, and 1OF9.

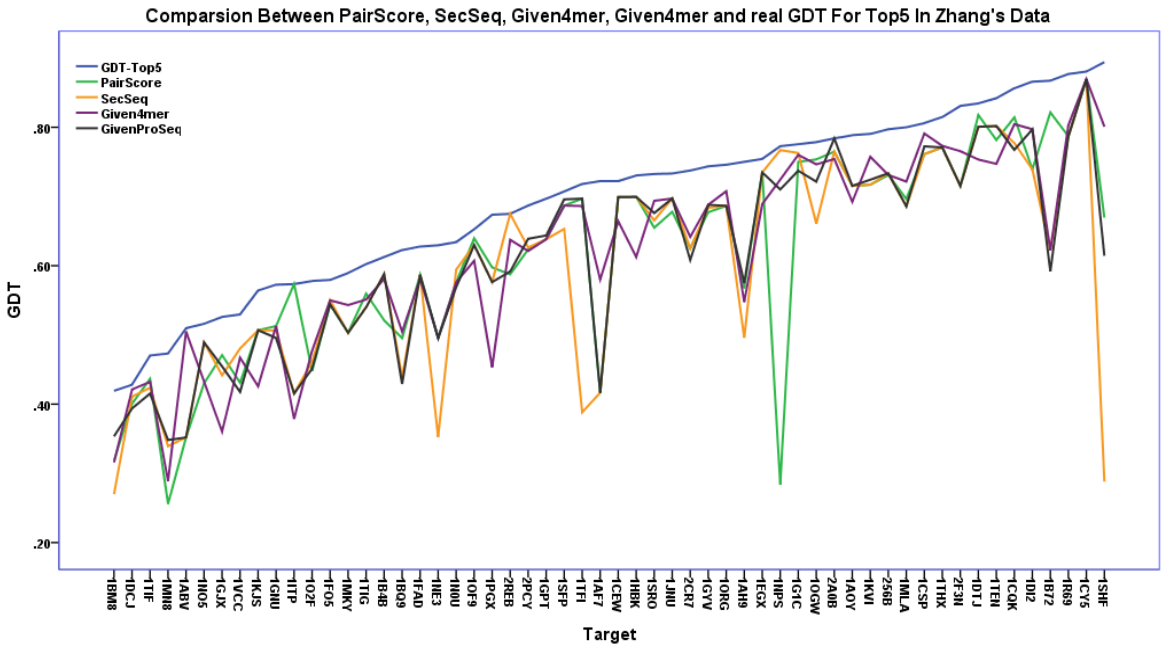

Figure 17. Performance in Yang Zhang's Data of PairScore, SecSeq, Given4mer, GivenProSeq, and Best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Top5 Selection.

In the above graph, the comparison is between 4-mer methods. In the overall results, Given4mer performed the best followed by GivenProSeq, Pair score, and SecSeq respectively. Given4mer method performed the best in some targets such as 1ABV, 1AF7, and 1SHF. However, it performed the worst in some targets such as 1GJX and 1PGX. GivenProSeq method performed the best in some targets such as 2A0B. However, it also performed the worst in some targets such as 1B72. PairScore performed the best in some targets such as 1B72 and 1ITP. In this target, it predicted the best decoy. On the other hand, it performed the worst in some targets such as 1NPS. Secondary sequence performed the best in some targets such as 1NPS. However, it performed the worst in some targets such as 1BM8, 1NE3, 1TFI, and 1SHF.

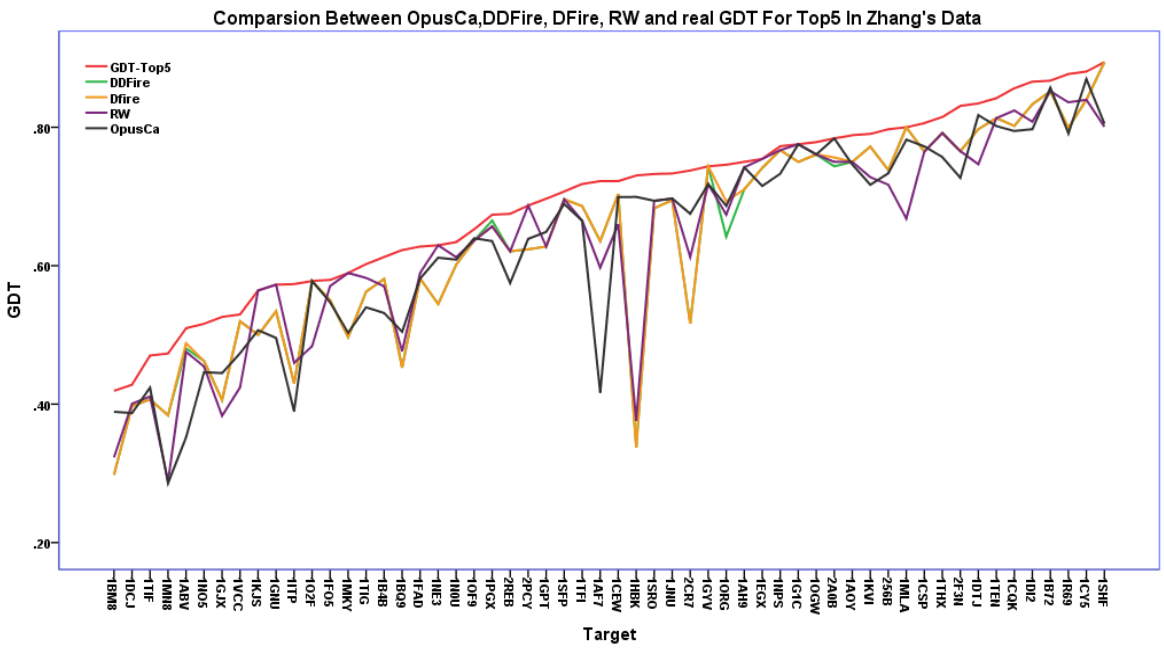

Figure 18. Performance in Yang Zhang's Data of DDfire, DFire, RW, and Best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Top5 Selection.

In the above graph, the methods show only single scores. In the overall results, Dfire performed the best, then DDfire, RW, and Opus-ca respectively. Dfire and DDFire methods had similar results. Dfire performed better in 1ORG target. DDfire performed better in 1PGX target. They both performed high in 1SHF. However, they both performed the worst in 2CR7. RW performed the best in some targets such as 1KJS, 1GNU, and 1MKY. However, it performed the worst for some targets such as 1MLA. Opus-ca performed the best in some targets such as 1HBK and 2A0B. However, it performed the worst in some targets such as 1AF7 and 2REB. Overall all in the plot, all the methods performed very well because of the capability to choose the best one from the Top5 models.

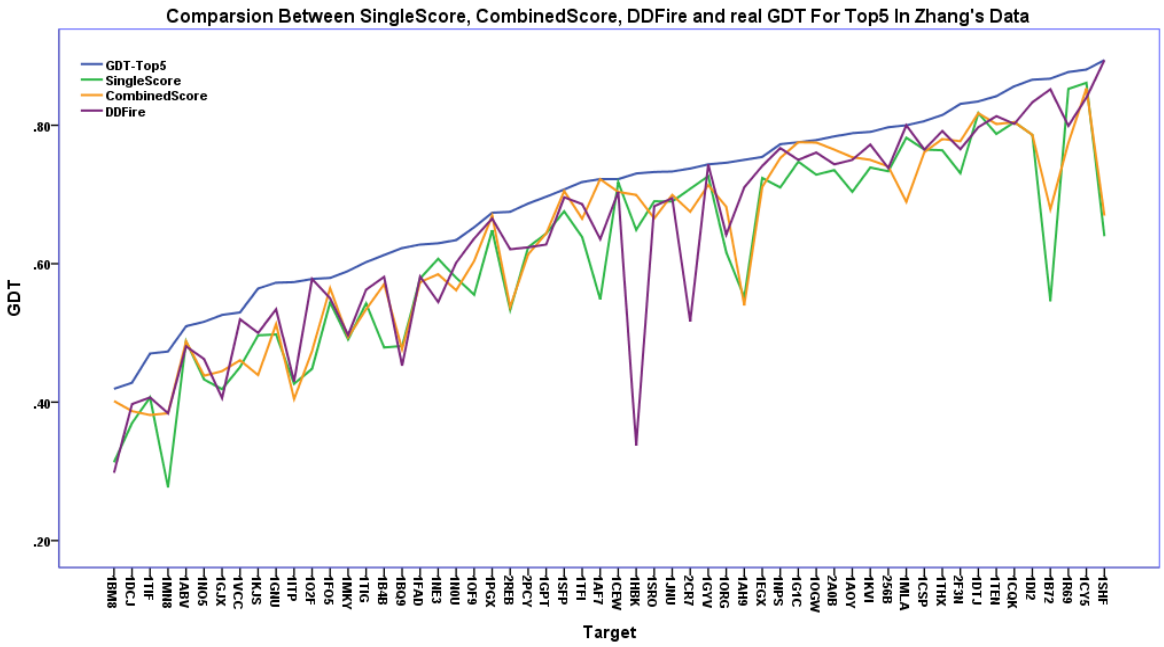

Figure 19 Performance in Yang Zhang's Data of Combined Score, DDFire, SPSP, and best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Top5 Selection.

In the above graph, the comparison is between mixed methods from different aspects. In the overall results, DDFire had the best performance, then CombinedMethod, followed by Single Position Specific Probability (SPSP) Score. CombinedMethod performed the best in some targets such as 1AF7 and 1BM8. However, it performed the worst in some cases such as 1KJS and 1MLA. DDFire performed the best in some targets such as 1VCC, 1AH9 as well as 1O2F and 1B72. However, it performed the worst in some cases such as 1HBK and 2CR7. Single Position Specific Probability (SPSP) Score performed the best in a few targets such as 1R69. However, it performed the worst in some targets such as 1MN8 and 1B72.

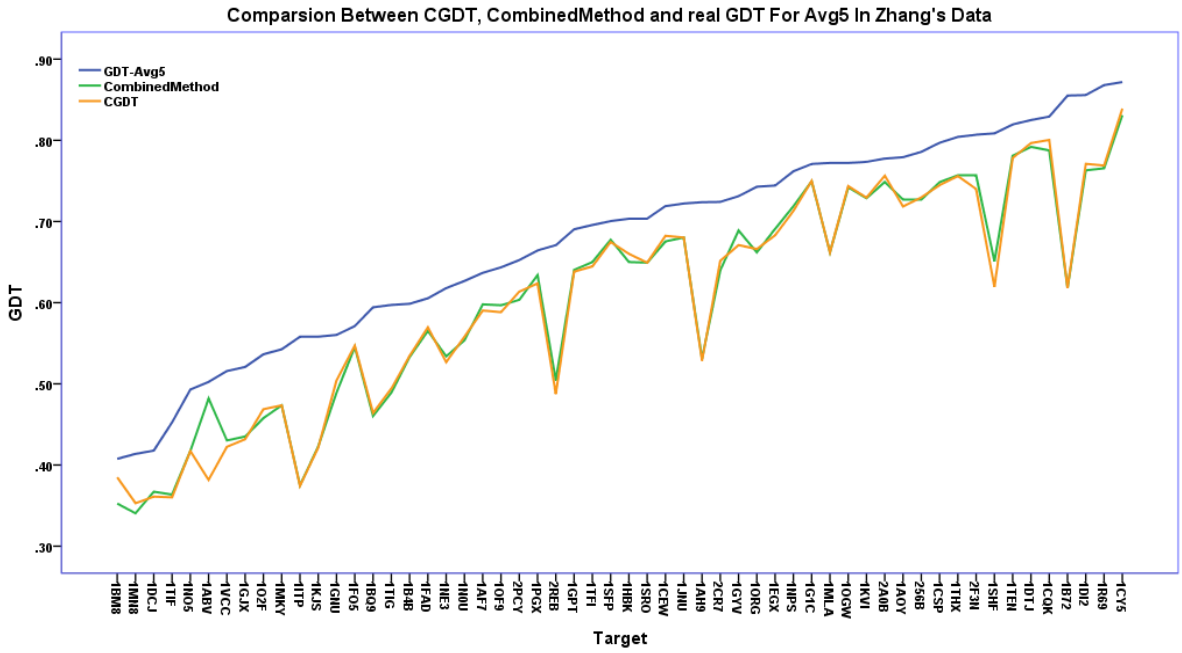

Figure 20 Performance in Yang Zhang's Data of CGDT, CombinedMethod and best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Avg5 Selections.

In the above graph, CombinedMethod still performed better than CGDT. It also performed better in the overall results. CombinedMethod performed the best in some targets such as 1ABV. However, it performed the worst in some targets such as 1BM8. In most cases, CombinedMethod and CGDT have similar scores.

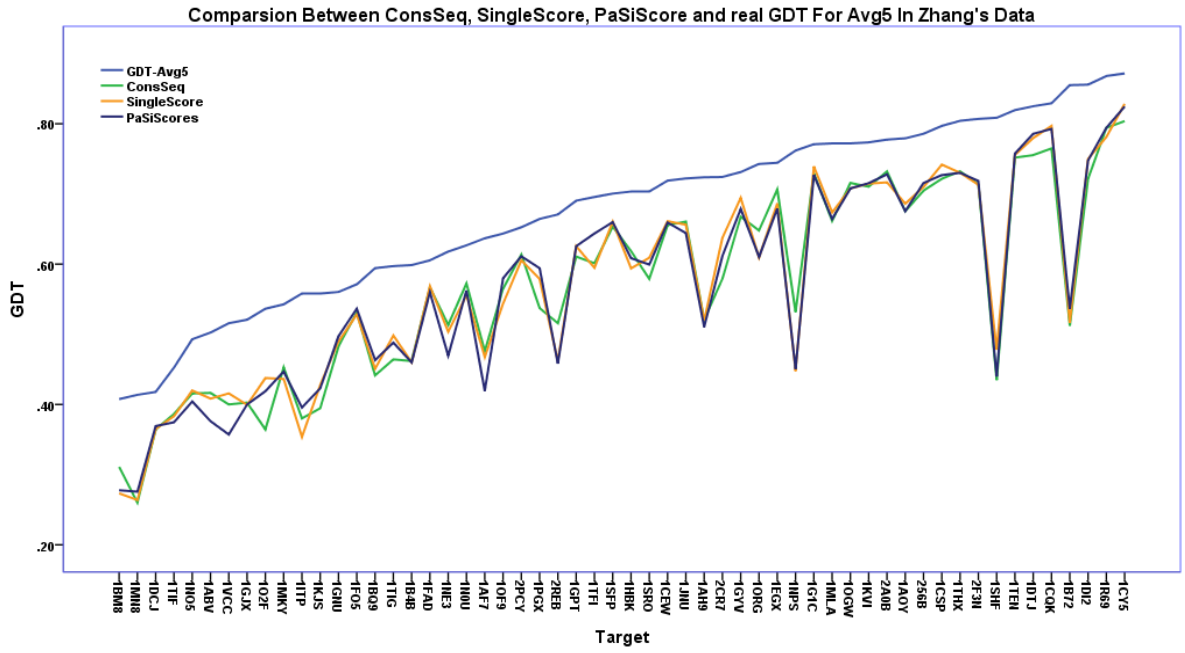

Figure 21. Performance in Yang Zhang's Data of ConsSeq, SPSP, PaSiScore and Best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Average Top5 selections.

In the above graph, all the methods are 4-mer methods. In the overall results, Single Position Specific Probability (SPSP) Score performed the best, then PaSiScore followed by ConsSeq. Since it is the average of the Top5 models, there were no big differences between the scores in every target. For example, SPSP performance was very similar to the others. It achieved a good score in some targets such as 1CSP and 1O2F. In this kind of a plot, if there is a big difference in some targets, there should be a big match or mismatch for that target with a method. The goal was to analyze the methods that match some targets as well as the methods that do not match other targets. In this way, understanding how a given target fits such a method helps in selecting fitted methods for particular targets. For the Top5 average, the scores tell whether it is a good fit method for a target. ConsSeq performed better than the other methods

in some targets such as 1BM8 and 1NPS. However, it performed the worst in some targets such as 1O2F. PaSiScore performed the best in some targets such as 1TFI. However, PaSiScore performed the worst in some targets such as 1VCC and 1AF7.

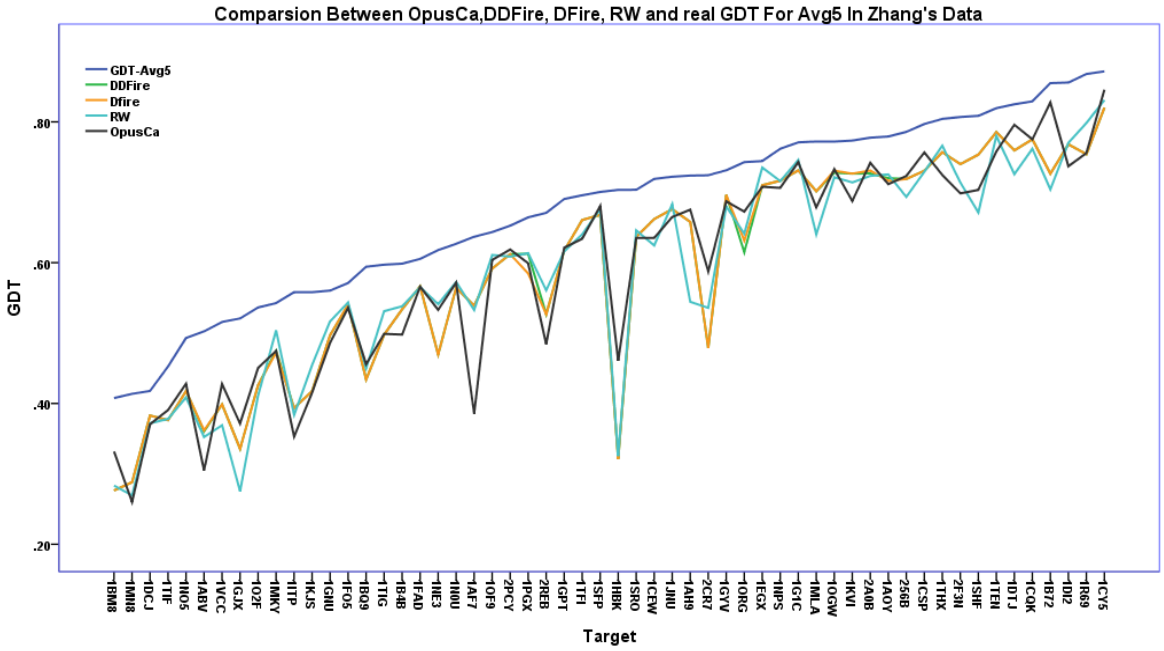

Figure 22. Performance in Yang Zhang's Data of Opus-ca, DDFire, DFire, RW, and Best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Average Top5 Selections.

In the above graph, the methods are all single scores. In the overall results, they performed better as follows: Opus-ca, DDFire, DFire, and then RW. Opus-ca performed the best in few targets such as 1BM8, 1HBK and 1B72. However, it performed the worst in few other targets such as 1AF7. DDFire and Dfire methods gave almost similar results even for the average of the Top5 models. They did not perform the best or the worst significantly. RW performed a little better than the other methods in a few targets such as 1TIG and 1EGX. However, RW performed the best in targets such as 1GJX and 1AH9.

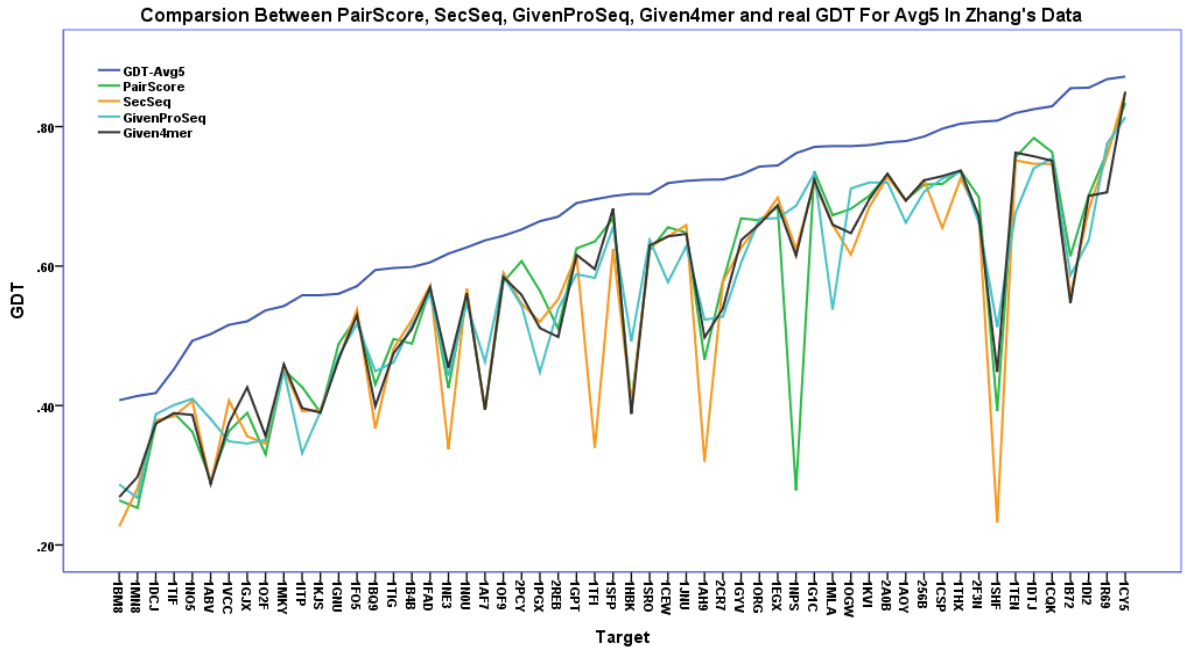

Figure 23. Performance in Zhang's Data of PairScore, SecSeq, GivenProSeq, Given4mer and Best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Average Top5 Selections.

In the above graph, all the methods are 4-mer methods. In the overall results, the ranking for these methods are as follows: GivenProSeq, Given4mer, PairScore, and SecSeq. GivenProSeq performed better in some targets such as 1ABV, 1HBK and 1NPS. However, it performed the worst in some targets such as 1ITP, 1PGX, 1CEW, and 1MLA. Given4mer method performed better than the other targets such as 1GJX. However, it performed the worst in some targets such as 1R69. Pair score performed the best among the methods in some targets such as 2PCY, 1GYV, 1PGX, 1TFI, and 1DTJ. However, it performed the worst in some cases such as 1NPS. Secondary sequence method performed the best over the other methods in some targets such as 1VCC. However, it performed the worst in some other targets such as 1TFI, 1AH9, and 1SH

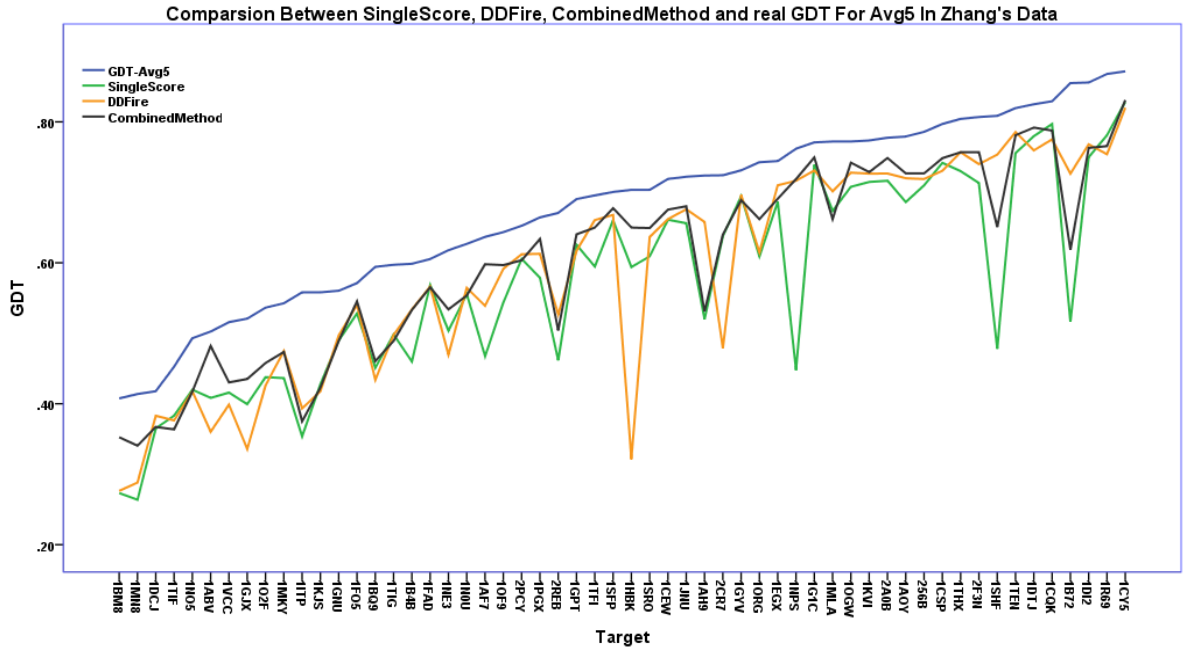

Figure 24. Performance in Yang Zhang's Data of SPSP, DDFire, CombinedMethod, and Best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Average Top5 Selections.

In the above figure, the methods are from different aspects. Overall CombinedMethod performed the best; DDfire came in second; the last was Single Position Specific Probability (SPSP) Score. CombinedMethod performed the best in some targets such as 1ABV, 1AF7, and 1HBK. CombinedMethod did not show any significant worst cases. DDfire performed the best in some cases such as 1SHF and 1B72. However, DDfire performed the worst in some cases such as 1HBK and 2CR7. Single Position Specific Probability (SPSP) Score performed the worst in some cases such as 1NPS, 1SHF, and 1B72.

4.2. Performance on Rosetta Data

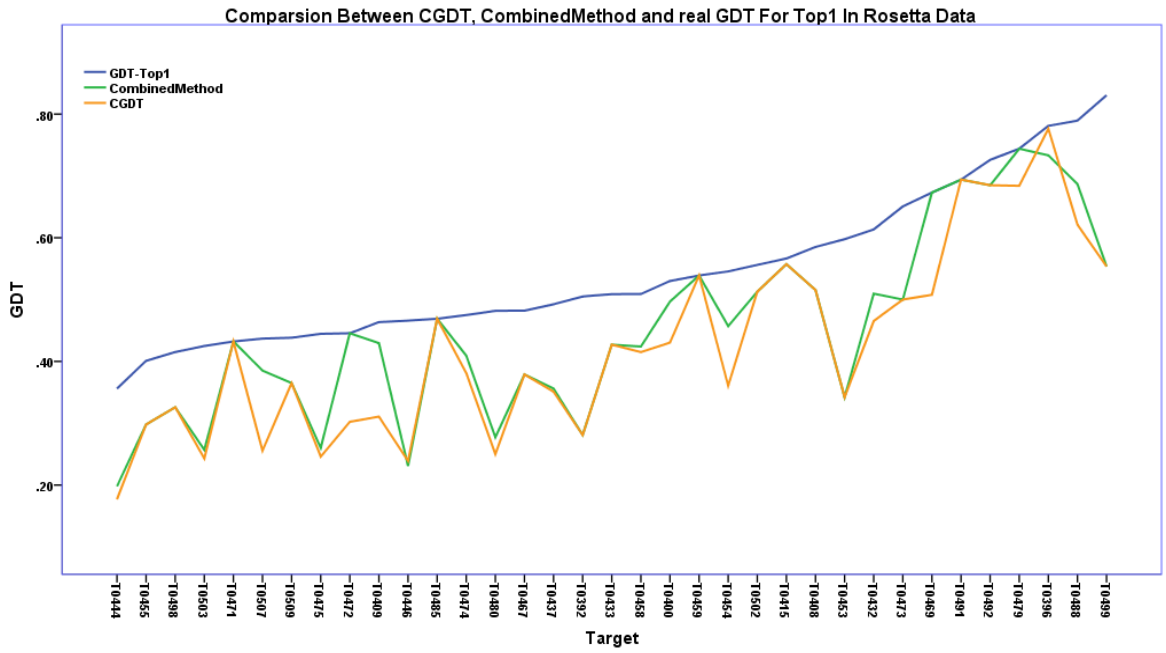

Figure 25. Performance in Rosetta Data of CGDT, CombinedMethod and Best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Top1 Selection.

Overall CombinedMethod performed better than CGDT. CombinedMethod performed the best in some targets such as T0472, T0409, and T0469. In those targets, CGDT performed very low compared to the CombinedMethod. CGDT performed better in T0396 target. However, CGDT performed low in some targets such as T0507 and T0472. CombinedMethod showed significant improvement in performance over the state-of-the-art methods.

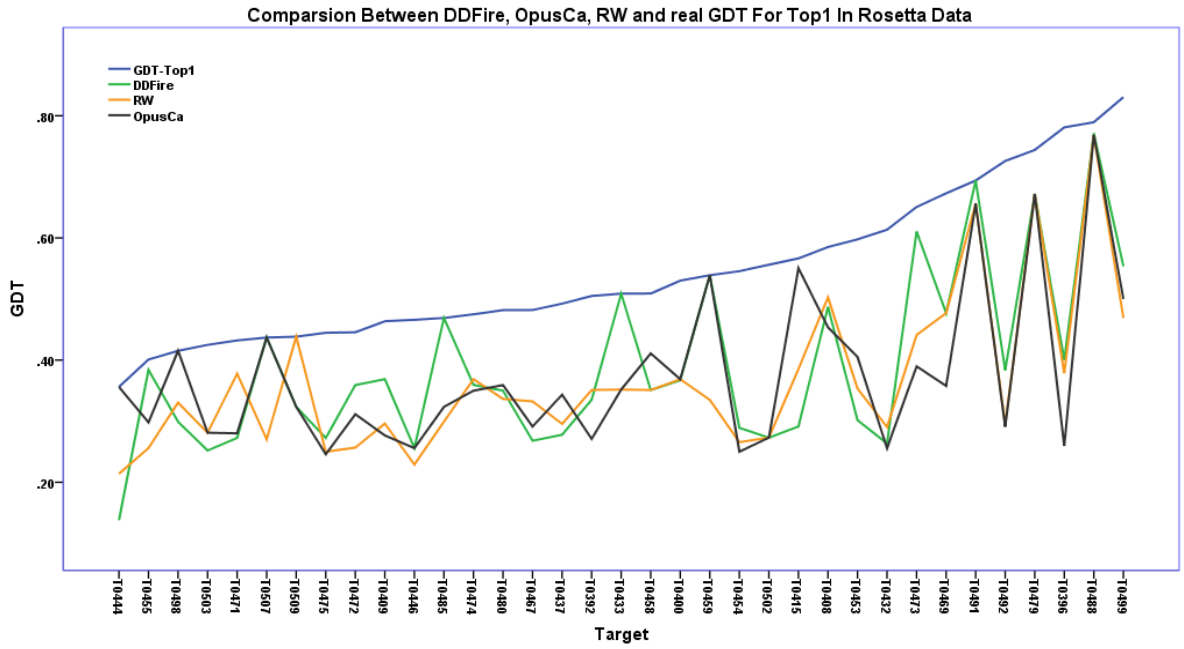

Figure 26. Performance in Rosetta Data of DDFire, Opus-ca, RW, and Best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Top1 Selection.

In the above graph, all the methods are single scores. Overall DDFire performed the best; Opus-ca came in second; and the last was RW. DDFire performed better than the other methods in some targets such as T0485, T0433, and T0473. However, it performed worse than the other methods in some cases such as T0444 and T0415. Opus-ca performed better in some targets such as T0444, T0498, and T0415. However, it performed worse in some targets such as T0469 and T0396. RW performed better in some targets such as T0471 and T0509. However, it performed worse in some other targets such as T0459 and T0472.

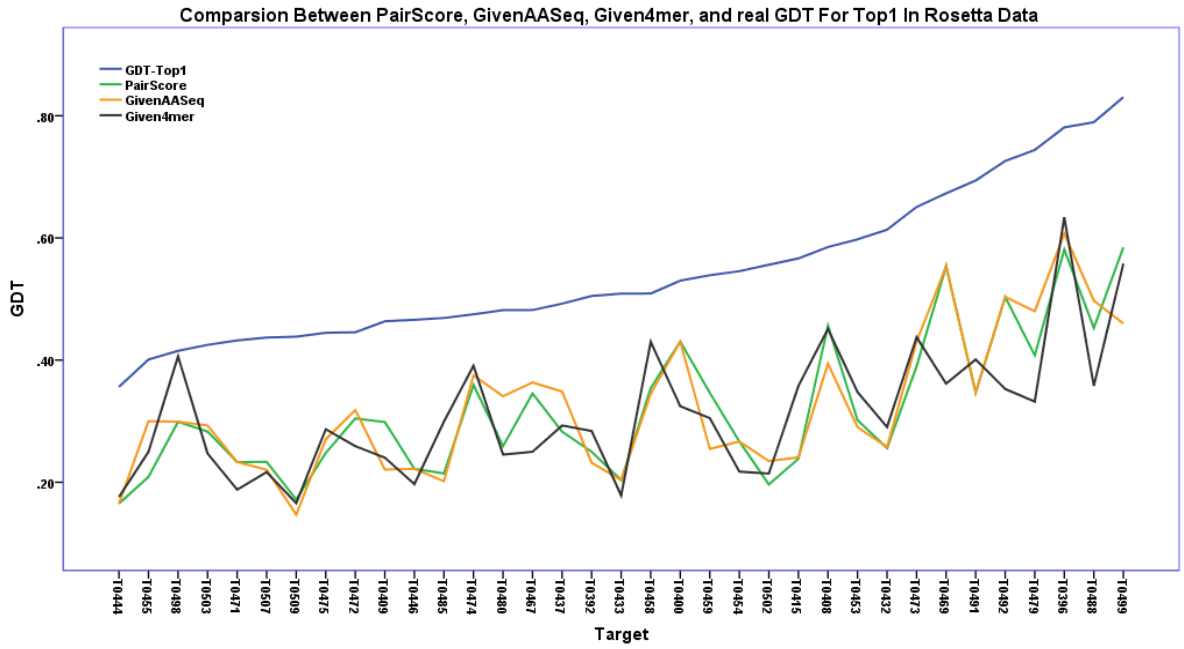

Figure 27. Performance in Rosetta Data of PairScore, GivenAASeq, Given4mer, and Best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Top1 Selection.

In the above graph, all the methods are based on 4-mer sequence. In the overall results, GivenAASeq performs the best; Pair score comes second, and Given4mer ranked last. GivenAASeq performed the best in some targets such as T0455, T0480, and T0437. However, it performed the worst in some cases such as T0408, T0459 and T0499. Given4mer performed the best in some targets such as T0498, T0485 and T0415. However, it performed the worst in some targets such as T0469, T0492, T0479, and T0488. Pair score performed the best in some targets such as T0409 and T0459. However, it performed the worst in some other targets such as T0455.

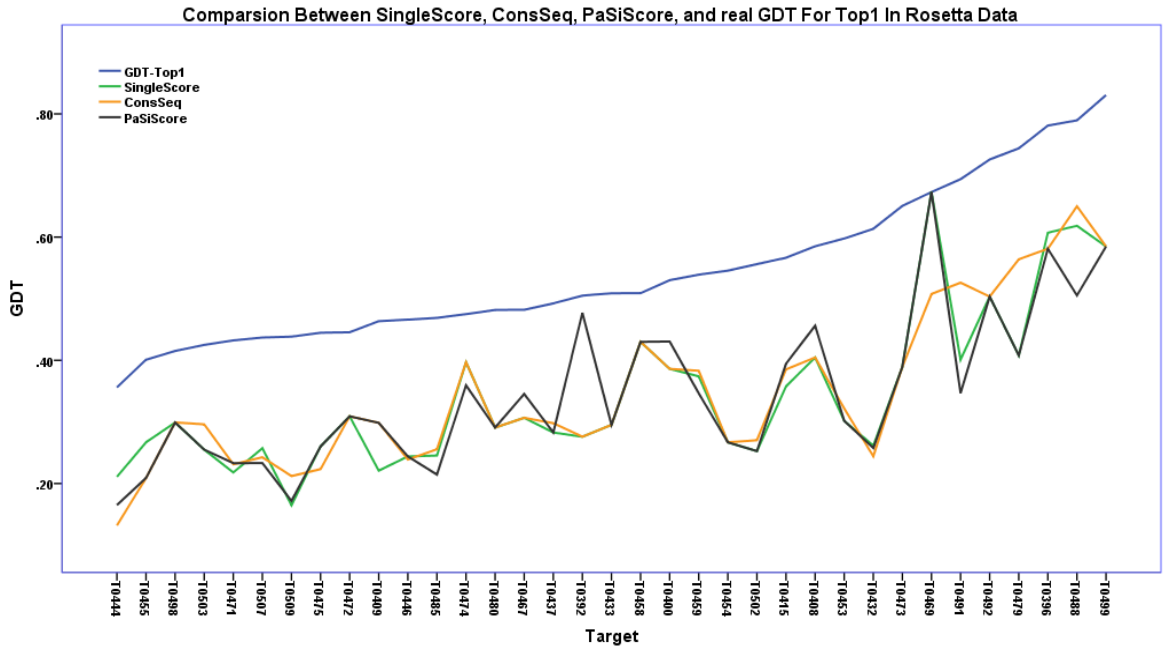

Figure 28. Performance in Rosetta Data of SPSP, ConsSeq, PaSiScore, and Best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Top1 Selection.

In the above graph, all the methods are based on 4-mer sequence. Overall ConsSeq performed the best followed by PaSiScore and then Single Position Specific Probability (SPSP) Score. ConsSeq performed the best in some targets such as T0491 and T0479. However, it performed worse than the other methods in some cases such as T0444 and T0469. PaSiScore performed better in some targets such as T0392 and T0408. However, it performed worse than T0491 and T0488. Single Position Specific Probability (SPSP) Score performed better in some targets such as T0444 and T0455. However, it performed the worst in T0409.

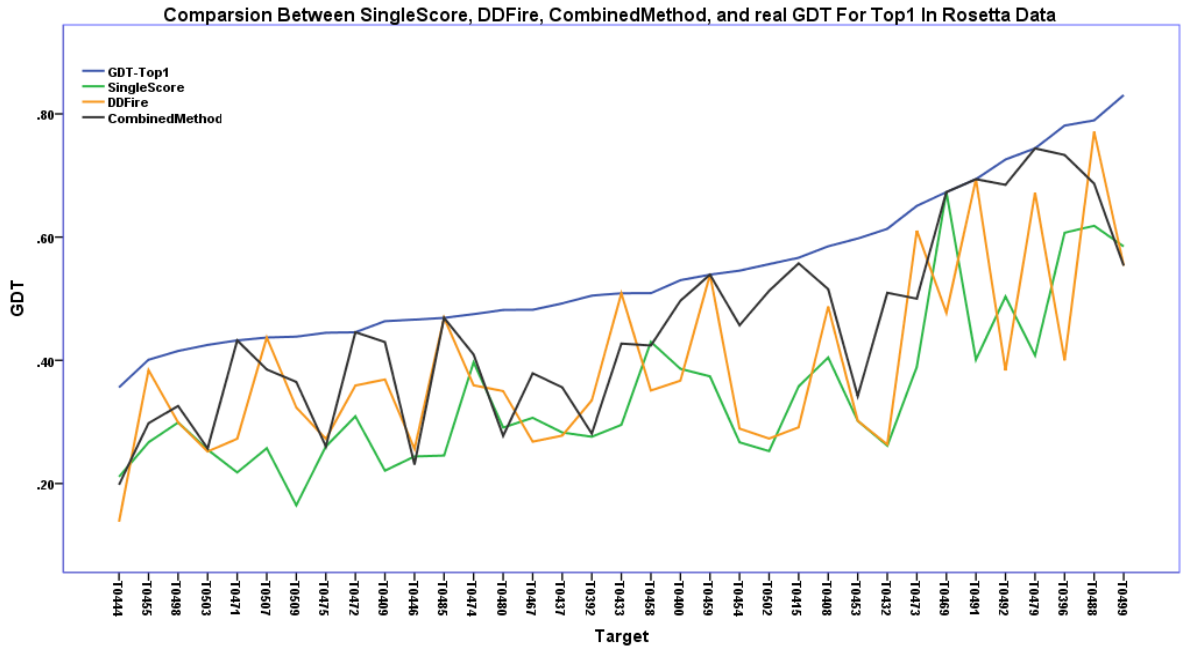

Figure 29. Performance in Rosetta Data of DDFire, SPSP, Combined Score, and Best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Top1 Selection.

In Figure 28, the compared methods are from different aspects. Overall CombinedMethod ranked first, then DDFire, and Single Position Specific Probability (SPSP) Score was last. CombinedMethod performed better in most cases such as T0471, T0472, T0415, and T0396. As expected, it did not give significantly lower results than the other methods. DDFire performed better in some targets such as T0455, T0433, and T0488. However, it performed worse than the other methods in some cases such as T0492 and T0396. Single Position Specific Probability (SPSP) Score was inferior to all other methods. Single Position Specific Probability (SPSP) Score performed less than the other methods, notably in targets T0507, T0485, and T0509.

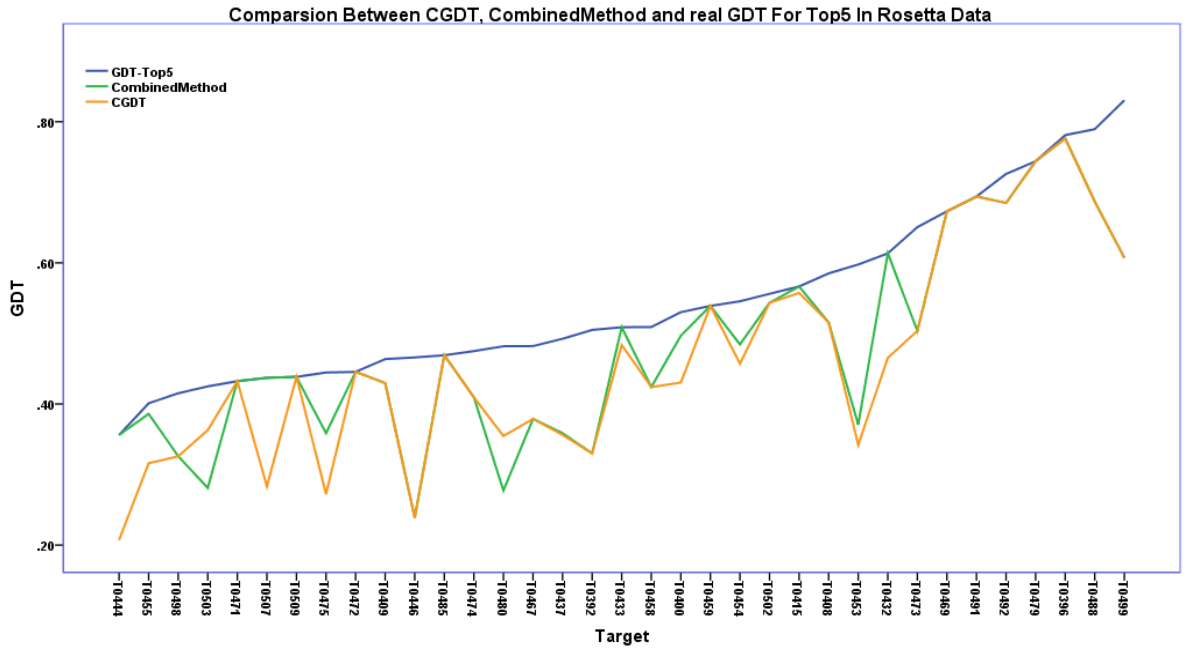

Figure 30. Performance in Rosetta Data of CGDT, CombinedMethod and Best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Top5 Selection.

In the above graph, CombinedMethod performed better than CGDT. CombinedMethod performed better in some cases while CGDT had low scores, notably in T0444, T0455, T0475, T0507, and T0432. However, in many cases CGDT had similar scores. CGDT performed better than CombinedMethod in few cases such as T0503 and T0480. This was due to some error in the model for the CombinedMethod.

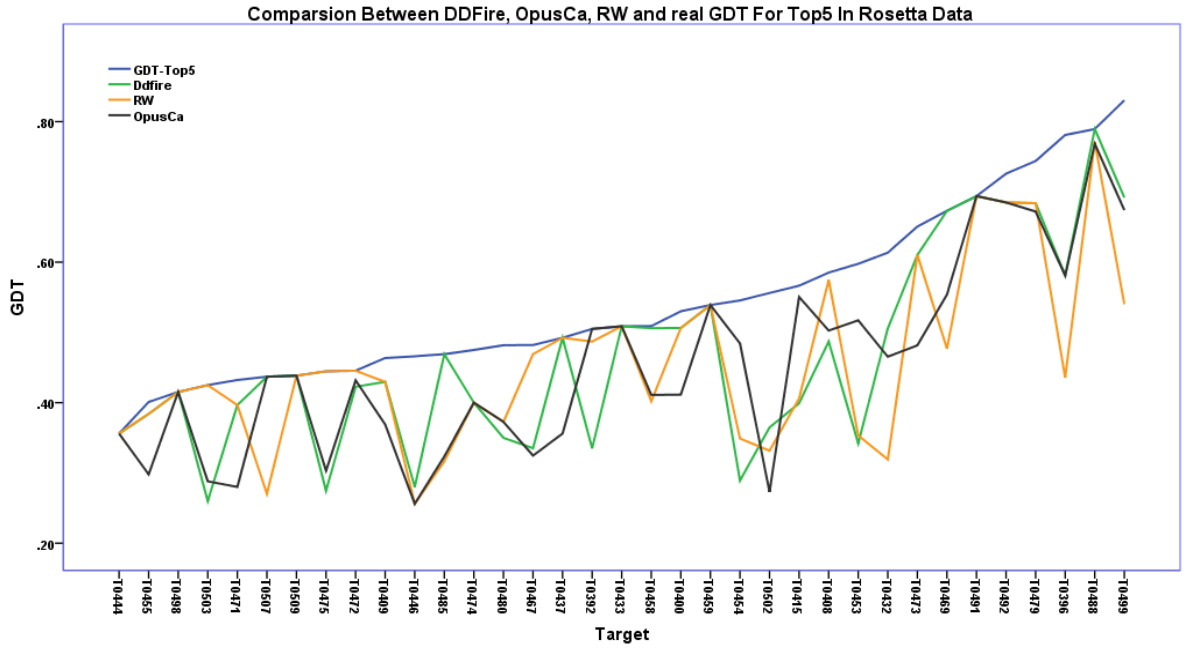

Figure 31. Performance in Rosetta Data of DDFire, Opus-ca, RW, and Best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Top5 Selection.

In the above graph, the methods are single scores only. In the overall results, DDFire performed the best, then RW, and finally Opus-ca. DDFire performed the best in some cases such as T0469, T0458, and T0485. However, it performed the worst in some cases such as T0392. Opus-ca performed the best in some cases such as T0415, T0454, and T0453. However, it performed the worst in some cases such as T0471, and T0437. RW performed the best in some cases such as T0503, T0475, and T0467. However, RW performed the worst in some cases such as T0507, T0432, T0396, and T0499.

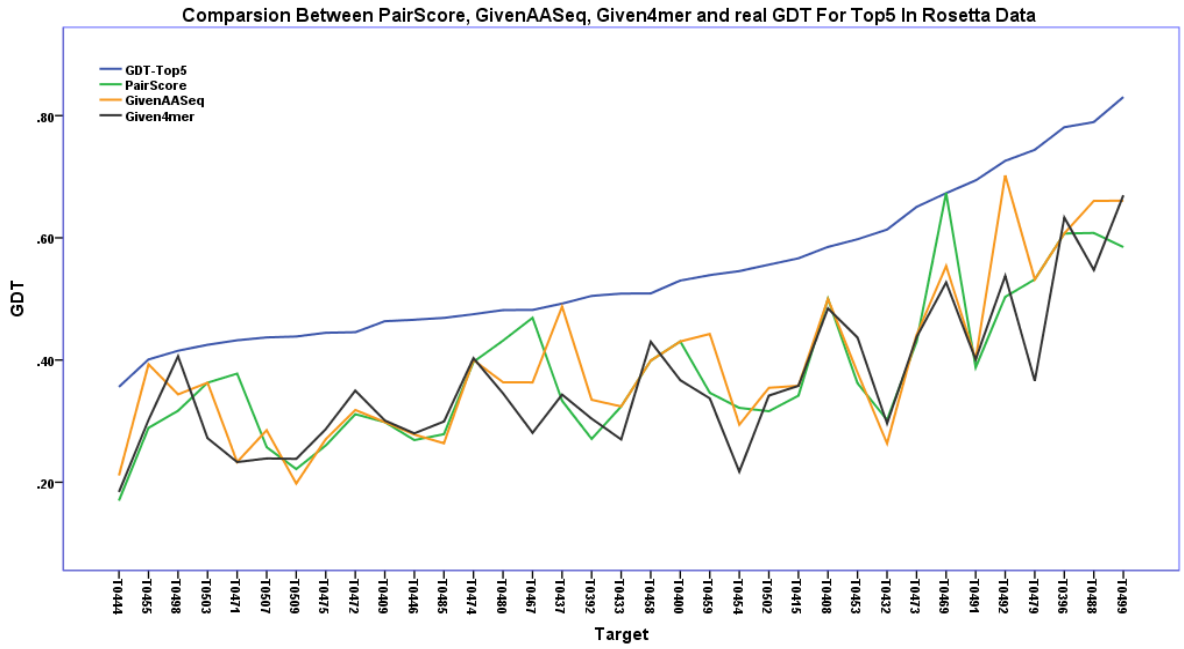

Figure 32. Performance in Rosetta Data of PairScore, GivenAASeq, Given4mer, and Best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Top5 Selection.

In the above graph, all the methods are based on 4-mer sequence. In the overall results, GivenAASeq performed the best, while PairScore came in second, and the last was Given4mer. GivenAASeq performed the best in some targets such as T0455, T0437, and T0492. However, it performed the worst in some cases such as T0432. PairScore performed the best in some targets such as T0471, T0467, and T0469. However, it performed the worst in some cases such as T0392 and T0499. Given4mer performed the best in some targets such as T0498 and T0472. However, it performed the worst in some cases such as T0467, T0454, and T0479.

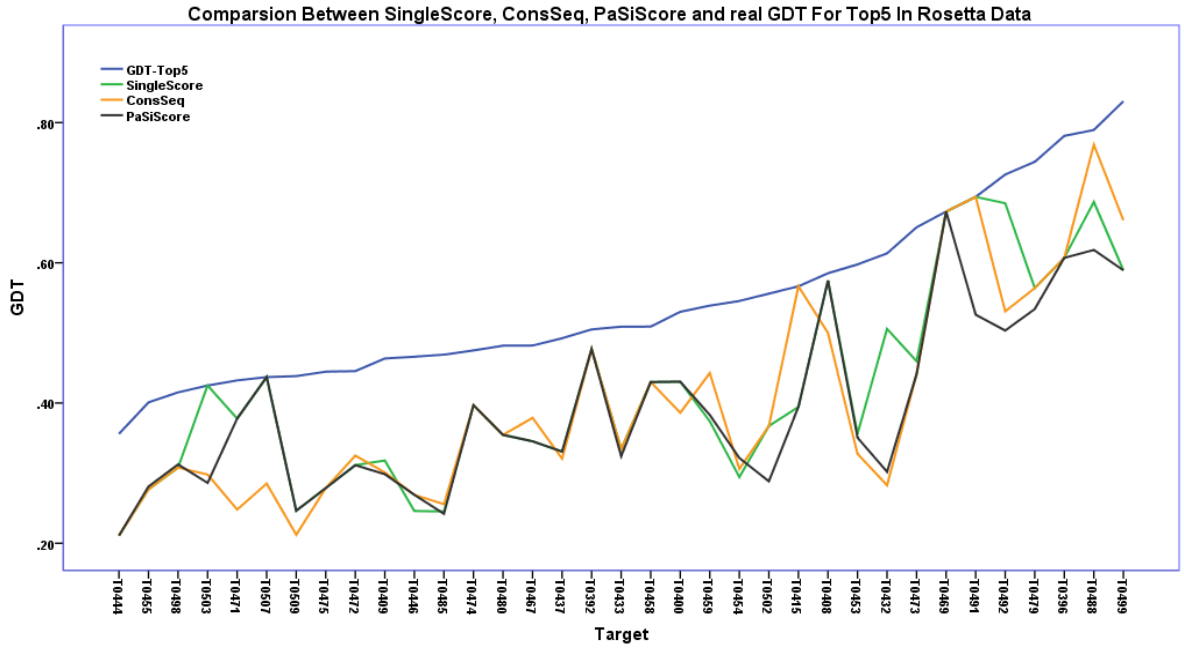

Figure 33. Performance in Rosetta Data of SPSP, ConsSeq, Sum of Pair-SPSP score, and Best GDT (blue) on y-axis and the 56 Targets on the x-axis for Top5 Selection.

In the above graph, all the methods are based on 4-mer sequence. In the overall results, Single Position Specific Probability (SPSP) Score performed the best over all the methods, with ConsSeq ranking second, and PaSiScore ranking last. Single Position Specific Probability (SPSP) Score performed the best on some cases such as T0503, T0432, and T0492. However, it performed the worst in T0446. ConsSeq performed the best in some cases such as T0415 and T0488. However, it performed the worst in some cases such as T0507, T0471 and T0509. PaSiScore had the worst performance in all the cases, most notably in T0491 and T0502.

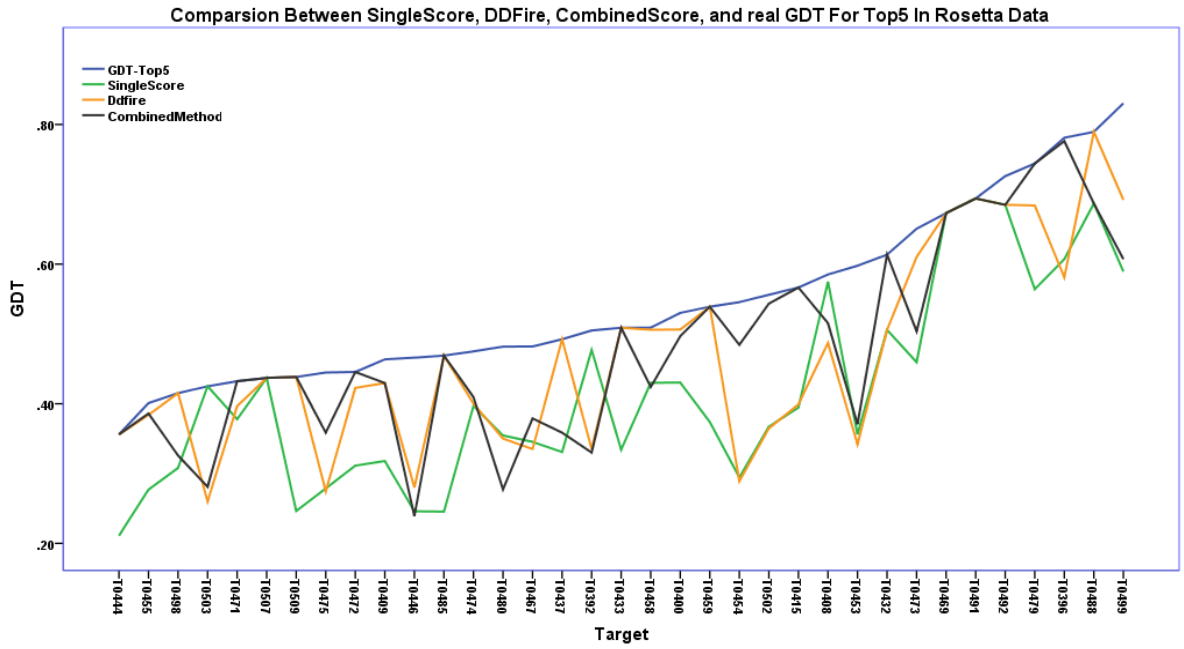

Figure 34. Performance in Rosetta Data of DDFire, CombinedMethod, SPSP, and Best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Top5 Selection.

In the above graph, the methods are from different aspects. In the overall results, CombinedMethod performed the best. DDFire ranked second followed by Single Position Specific Probability (SPSP) Score. CombinedMethod performed the best in some cases such as T0454, T0502, T0396, and T0415. However, it performed the worst in some other cases such as T0480. DDFire performed the best in some cases such as T0498, T0437, and T0488. However, it performed the worst in some cases such as T0503. Single Position Specific Probability (SPSP) Score performed the best in some cases such as T0408, T0503, and T0392. However, it performed the worst in some other cases such as T0444, T0509, and T0433.

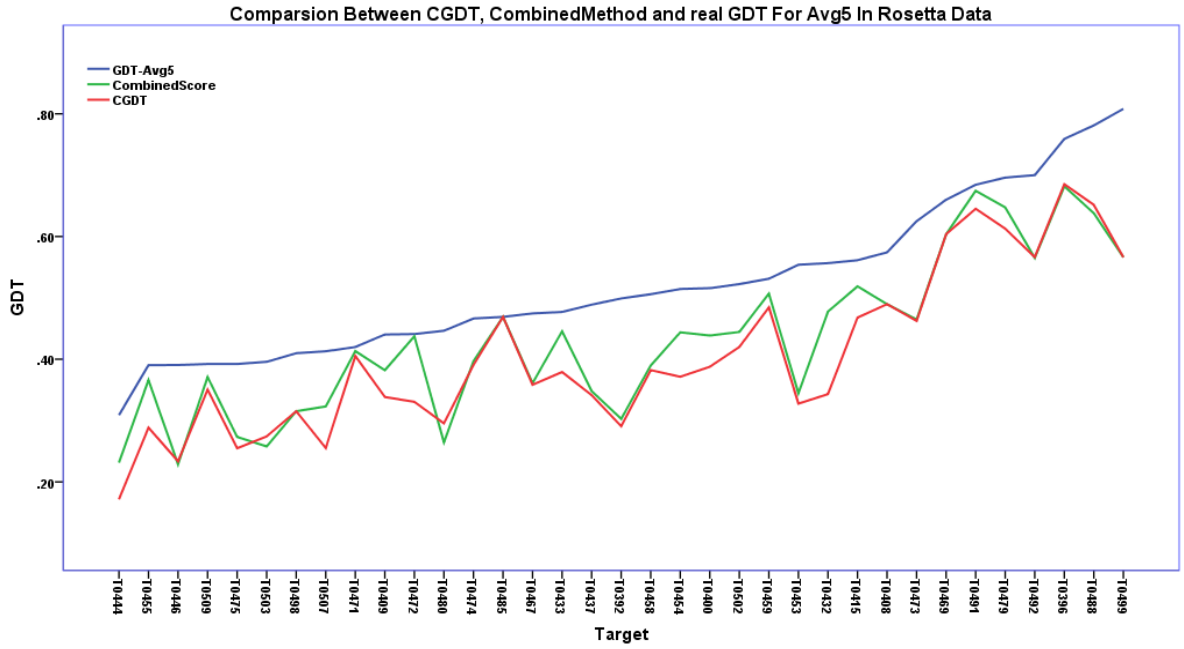

Figure 35. Performance in Rosetta Data of CGDT, CombinedMethod and Best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Avg5 Selection.

In the above figure, CombinedMethod performed better than CGDT in the overall results. CombinedMethod performed better than CGDT in some targets such as T0472, T0432 and T0433. However, CombinedMethod performed worse than CGDT in some targets due to training error such as T0480.

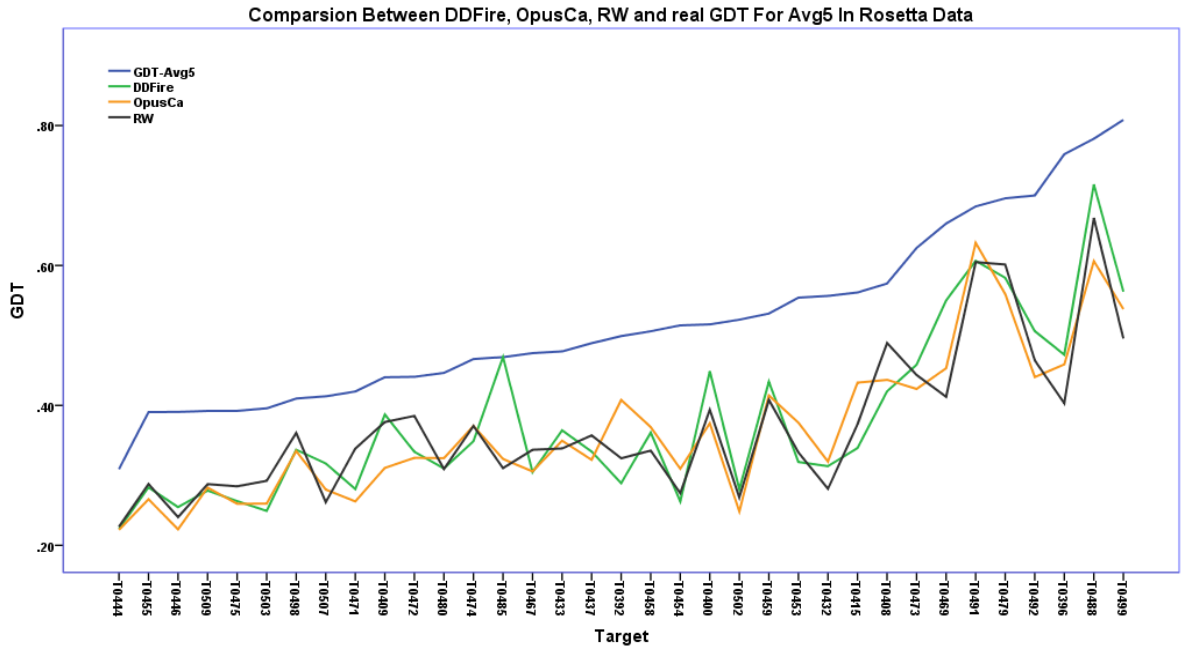

Figure 36. Performance in Rosetta Data of DDFire, Opus-ca, RW, and Best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Avg5 Selection.

In the above graph, all the methods are single scores. In the overall results, DDFire performed the best followed by RW and then Opus-ca. DDFire performed the best in some targets such as T0485 and T0469. However, DDFire performed the worst in some targets such as T0392. Opus-ca performed the best in some targets such as T0392. However, it performed the worst in some other targets such as T0409. RW performed the best in some targets such as T0471 and T0503. However, it performed the worst in some other targets such as T0396.

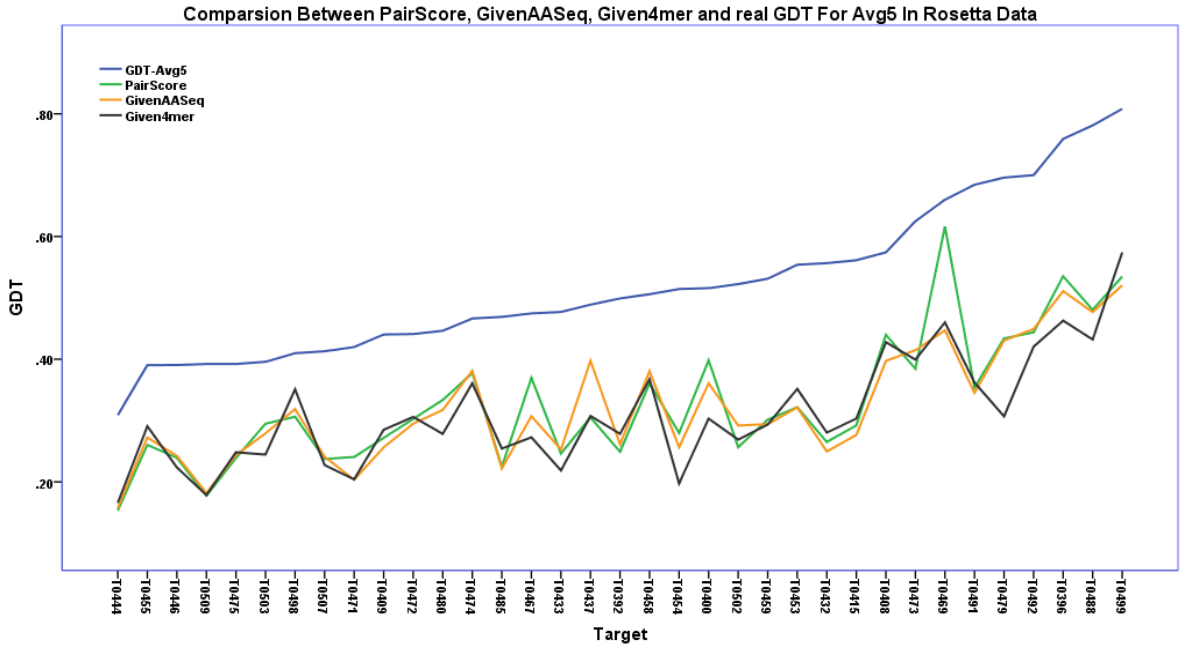

Figure 37. Performance in Rosetta Data of PairScore, GivenAASeq, Given4mer, and Best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Avg5 Selection.

In the above graph, all the methods are based on 4-mer sequence. Pair score performed the best, GivenAASeq, and then Given4mer. Pair score performed the best in some targets such as T0469 and T0467. However, it did not perform a distinct low score. GivenAASeq performed the best in some targets such as T0437. However, it did not receive a really significant bad performance score. Given4mer performed the best in some targets such as T0498, T0453, and T0499. However, it performed the worst in some other cases such as T0545 and T0479.

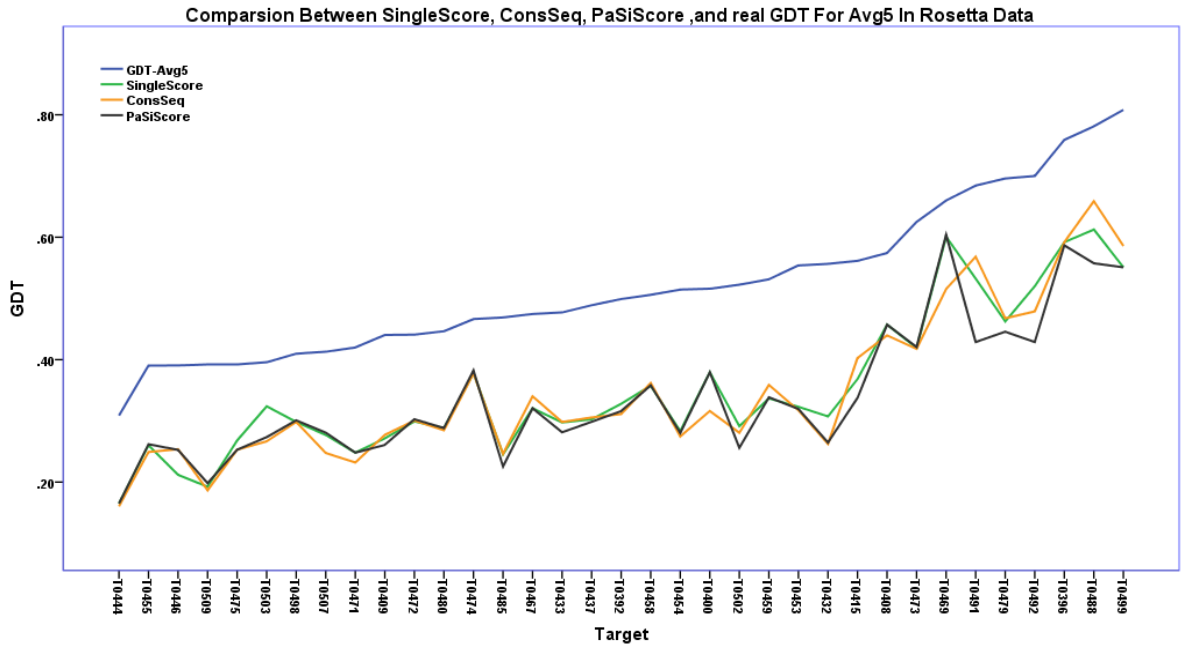

Figure 38. Performance in Rosetta Data of SPSP, ConsSeq, Sum of Pair–SPSP score, and Best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Avg5 Selection.

In the above graph, all the methods are based on 4-mer sequence. Overall Single Position Specific Probability (SPSP) Score performed the best followed by ConsSeq and then PaSiScore. SPSP performed the best in some cases such as T0503. It did not perform badly in all cases. ConsSeq method performed the best in some cases such as T0488. However, it performed the worst in some cases such as T0400. PaSiScore did not perform better than any of the other methods, performing the worst in T0491 and T0492. In general, the curves go close to each other in most cases. That means they pick the Top5 models in a very close way.

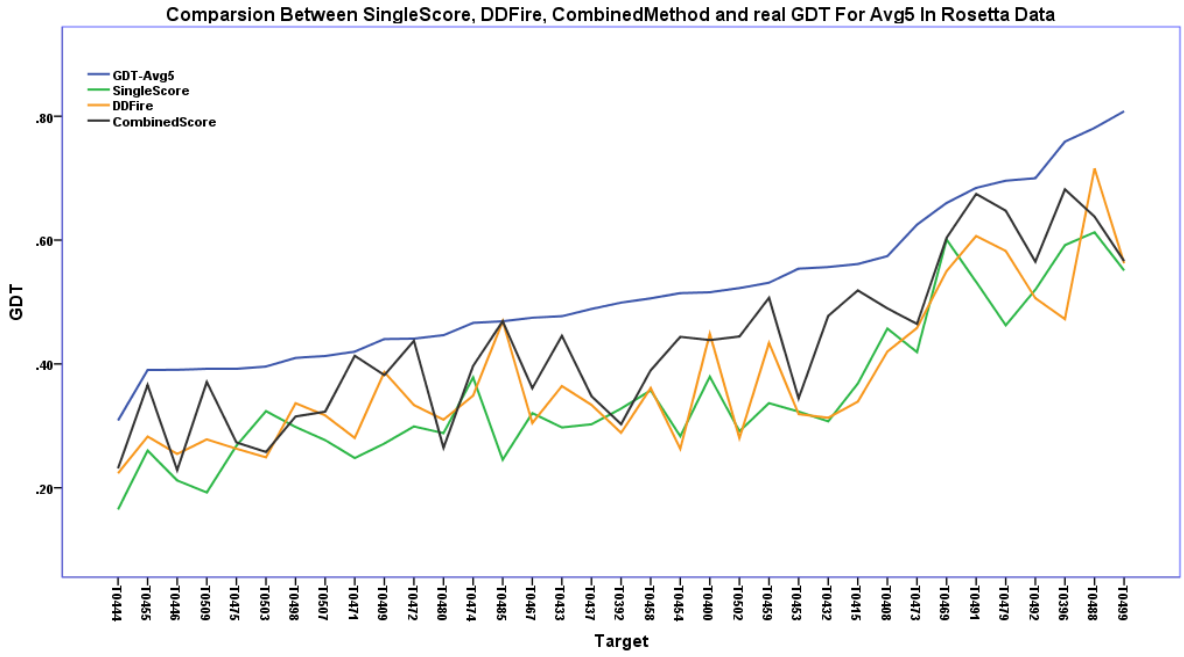

Figure 39. Performance in Rosetta Data of SPSP, DDFire, Combined Score, and Best GDT (Blue) on y-axis and the 56 Targets on the x-axis for Avg5 Selection.

In the above graph, the methods are from different aspects. Overall, CombinedMethod performed the best followed by DDFire and then Single Position Specific Probability (SPSP) Score. CombinedMethod performed the best in some targets such as T0509, T0502, and T0432. DDFire performed the best in some targets such as T0488. However, DDFire performed the worst in some other targets such as T0396. SPSP performed the best in some targets such as T0503. However, SPSP performed the worst in some targets such as T0479 and T0509.

Table 10. Best and Worst performance of the Methods over the Targets in Both Benchmarks for Top1 Model Selection.

Top1 in both Benchmarks				
	Yang Zhang's Data		Rosetta Data	
Method	Best performance	Worst performance	Best performance	Worst performance
CGDT	1TIF,1ITP	1ABV	T0396	T0507,T0472
DDFire	1MLA	1B09,2CR7	T0485,T0433	T0444,T0415
Opus-ca	1CEW,1SRO	1AF7	T0415,T0444	T0469,T0396
RW	1KJS	1TEN	T0471,T0509	T0459,T0472
SPSP	1SHF,1DI2	1PGX	T0444,T0455	T0409
Given4merSeq	1ABV,1CSP	1ITP,1GJX	T0498,T0485	T0492,T0479
GivenProSeq	1CEW	2REB	T0480,T0437	T0408,T0499
Cons.Seq	1MKY,2REB	1AF7	T0491,T0479	T0444,T0469
CombinedMethod	1ABV,1THX	1ITP	T0469,T0472	T0396
Pair Score	1OGW,2PCY	1DI2,1AOY	T0409,T0459	T0455
SumOfPaSPSPscore	1B72	1VCC,2CR7	T0392,T0408	T0491,T0488

Table 11. Best and Worst Performance of Methods Over Targets in Both Benchmarks for the Best Model Chosen from the Top5 Models Selection.

Top5 in both Benchmarks				
	Yang Zhang's Data		Rosetta Data	
Method	Best performance	Worst performance	Best performance	Worst performance
CGDT	2A0B	1ABV	T0503,T0480	T0444
DDFire	1SHF	2CR7	T0458,T0485	T0392
Opus-ca	1HBK,2A0B	1AF7,2REB	T0454,T0453	T0471,T0437
RW	1KJS,1MKY	1MLA	T0503,T0475	T0507,T0432
SPSP	1NE3,1MLA	1OF9	T0432,T0492	T0446
Given4merSeq	1ABV,1SHF	1GJX,1PGX	T0498,T0472	T0454,T0479
GivenProSeq	2A0B	1B72	T0455,T0437	T0432
Cons.Seq	1BM8,2REB	1KJS,1SRO	T0415,T0488	T0507,T0471
CombinedMethod	1ABV,1AF7	2A0B	T0444	T0503,T0480
Pair Score	1ITP,1B72	1NPS	T0471,T0469	T0392,T0499
SumOfPaSPSPscore	1ITP,1OF9	1NE3,1AF7	T0454	T0491,T0502

Table 12. Best and Worst Performance of Methods Over Targets in Both Benchmarks for the Average Top5 Models Selection.

Avg5 in both Benchmarks				
	Yang Zhang's Data		Rosetta Data	
Method	Best performance	Worst performance	Best performance	Worst performance
CGDT	1BM8	1ABV	T0480	T0472,T0432
DDFire	1SHF,2F3N	1NE3	T0485,T0469	T0392
Opus-ca	1BM8,1B72	1AF7	T0392	T0409
RW	1TIG,1EGX	1GJX, 1AH9	T0471,T0503	T0396
SPSP	1CSP	1ITP	T0503	T0446
Given4merSeq	1GJX	1R69	T0498,T0453	T0545,T0479
GivenProSeq	1ABV,1HBK	1ITP, 1MLA	T0437	T0432
Cons.Seq	1BM8,1NPS	1O2F	T0488	T0400
CombinedMethod	1ABV	1BM8	T0472,T0432	T0480
Pair Score	2PCY,1PGX	1NPS	T0469,T0467	T0473
SumofPaSiScores	1TFI	1VCC, 1AF7	T0509	T0491

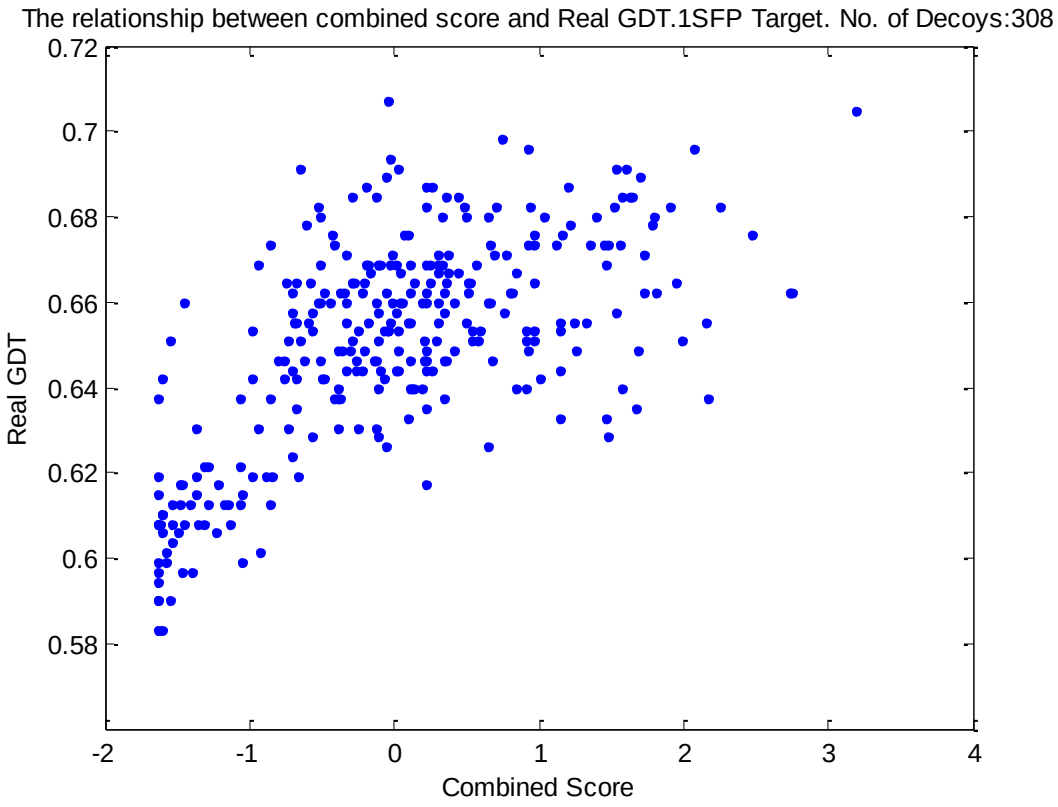

Figure 40. The Relationship between Combined Score and Real GDT on 1SFP Target in Yang Zhang Data

In the above graph, figure 40, 1SFP target decoys scores are plotted. It is considered to be an easy target in Zhang's Data. On the y-axis the real GDT was considered as a standard measurement. The combined score is on the x-axis the combined score. It partially correlates with real GDT. There are 308 decoys. Some of them correlate with GDT. However, the others are above or below the expectations. Combined score can predict one of the best decoys and ranks it the first. Some decoys share the same score from both measurements.

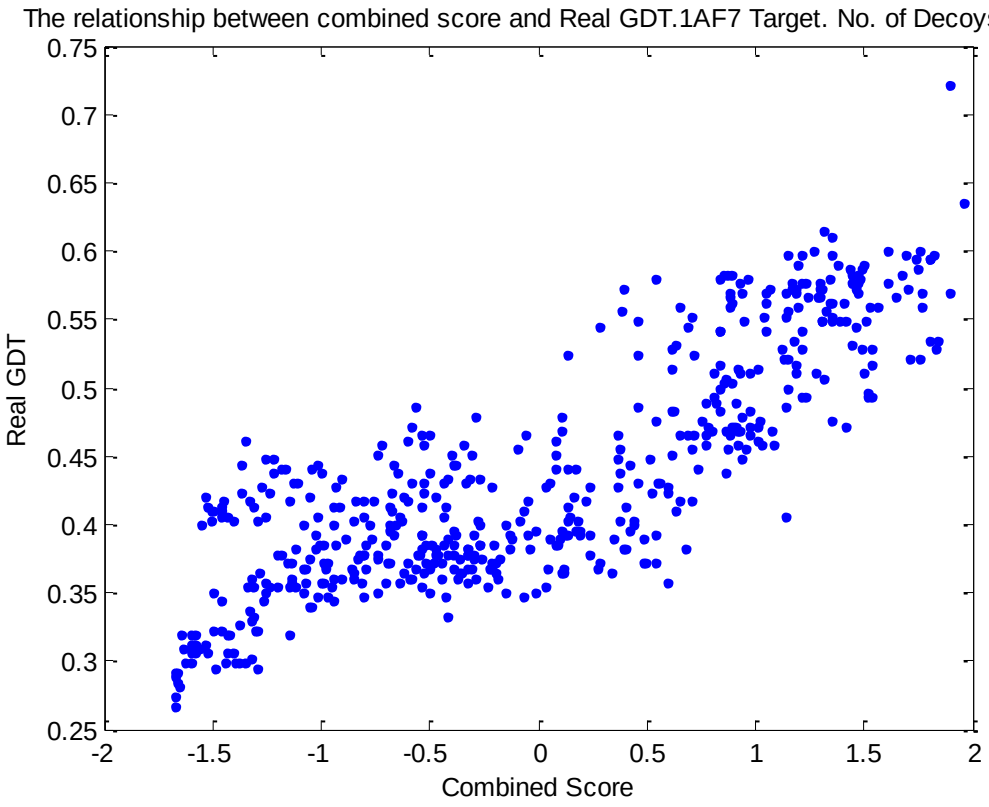

Figure 41. The Relationship between Combined Score and Real GDT on 1AF7 target in Yang Zhang Data

In the above graph, figure 41, 1AF7 target decoys scores are plotted. It is considered to be a medium target in Zhang's Data. On the x-axis is the real GDT. On the y-axis is the combined score. It partially correlates with real GDT. There are 527 decoys. Some of them correlate with GDT. However, the others are above or below the expectations. Combined score can predict one of the best decoys and rank it as first. Some decoys share the same score from both measurements.

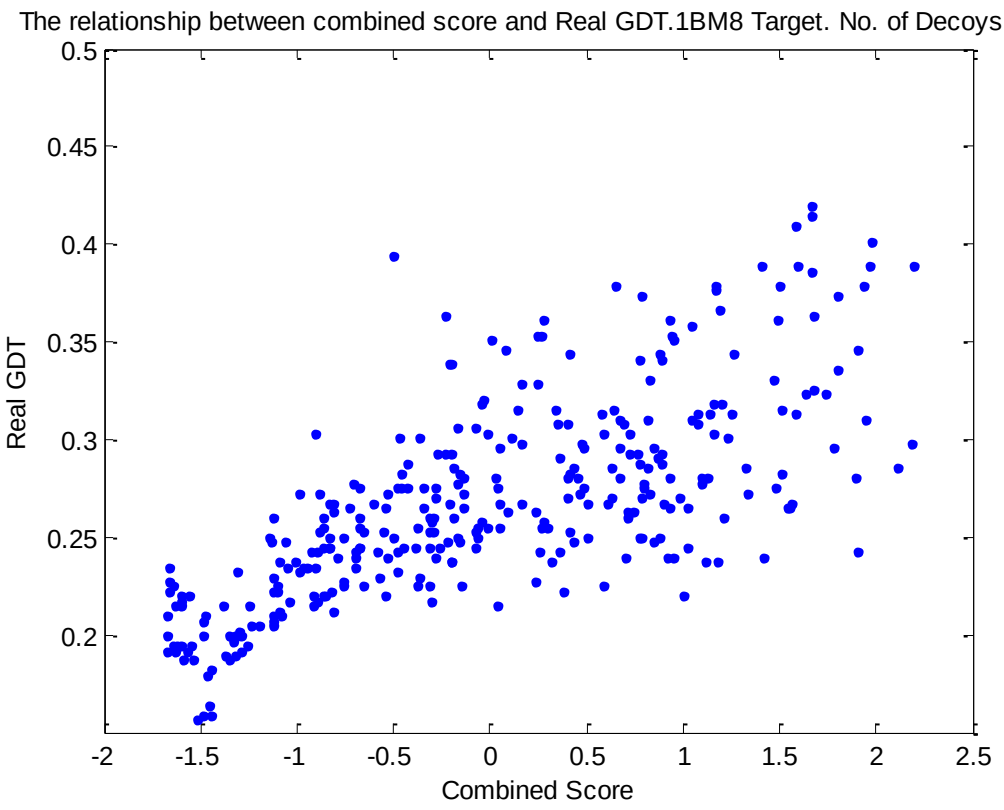

Figure 42. The Relationship between Combined Score and Real GDT on 1BM8 Target in Yang Zhang Data

In the above graph, figure 42, 1BM8 target decoys scores are plotted. It is considered to be a hard target in Zhang's Data. On the x-axis is the real GDT. On the y-axis is the combined score. It partially correlates with real GDT. There are 329 decoys. Few of them correlate with GDT. However, most of them are above or below the expectations. Combined score is not able of rank the best decoy in this target. Some decoys share the same score from both measurements.

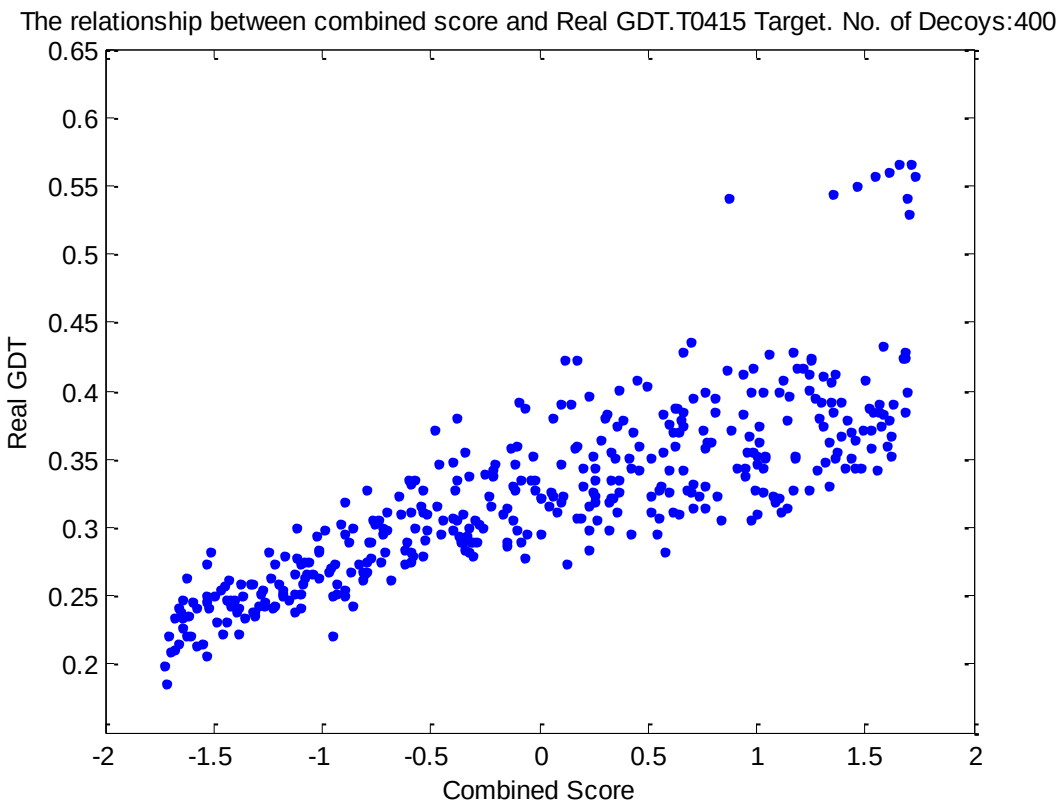

Figure 43. The Relationship between Combined Score and Real GDT on T0415 Target in Rosetta Data

In the above graph, figure 43, T0415 target decoys scores are plotted. It is considered to be a hard target in Rosetta Data. On the x-axis is the real GDT. On the y-axis is the combined score. It partially correlates with real GDT. There are 400 decoys. Some of them correlate with GDT. However, the other some are above or below the expectations. Combined score ranks the best decoy in this target among the Top5 models. Some decoys share the same score from both measurements.

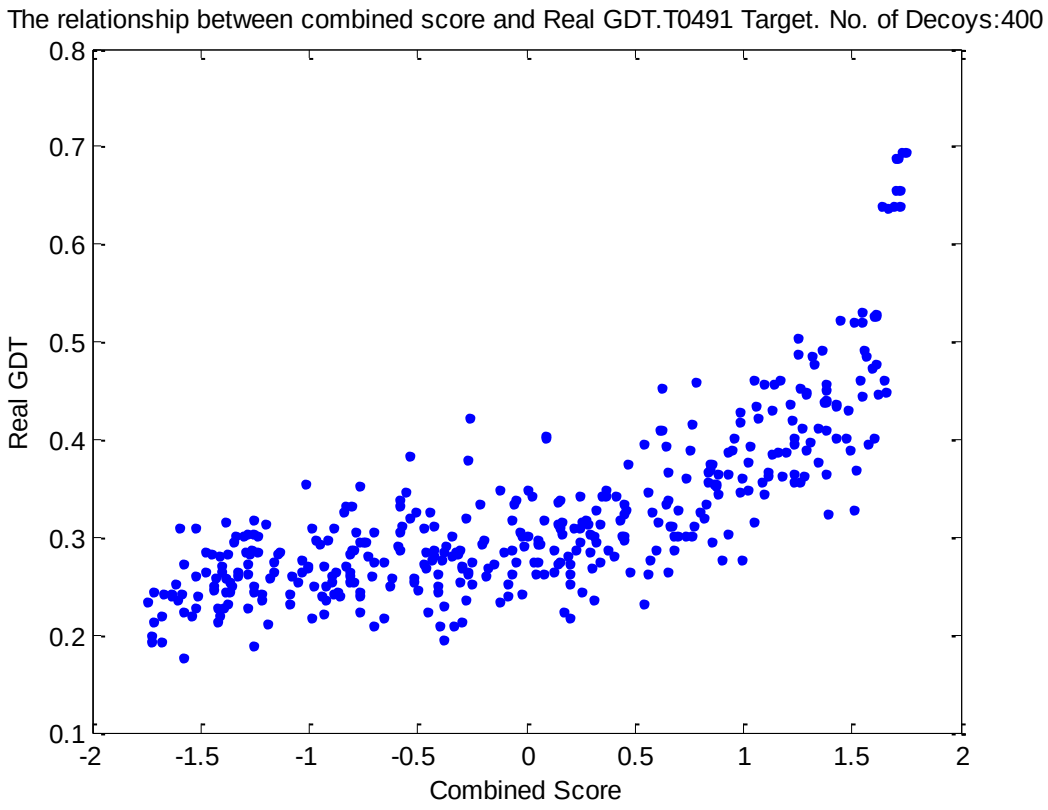

Figure 44. The Relationship between Combined Score and Real GDT on T0491 Target in Rosetta Data

In the above graph, figure 44, T0491 target decoys scores are plotted. It is considered to be a hard target in Rosetta Data. On the x-axis is the real GDT. On the y-axis is the combined score. It partially correlates with real GDT. There are 400 decoys. Some of them correlate with GDT. However, the other some are above or below the expectations. Combined score ranks the best decoy in this target to the first decoy. Some decoys share the same score from both measurements.

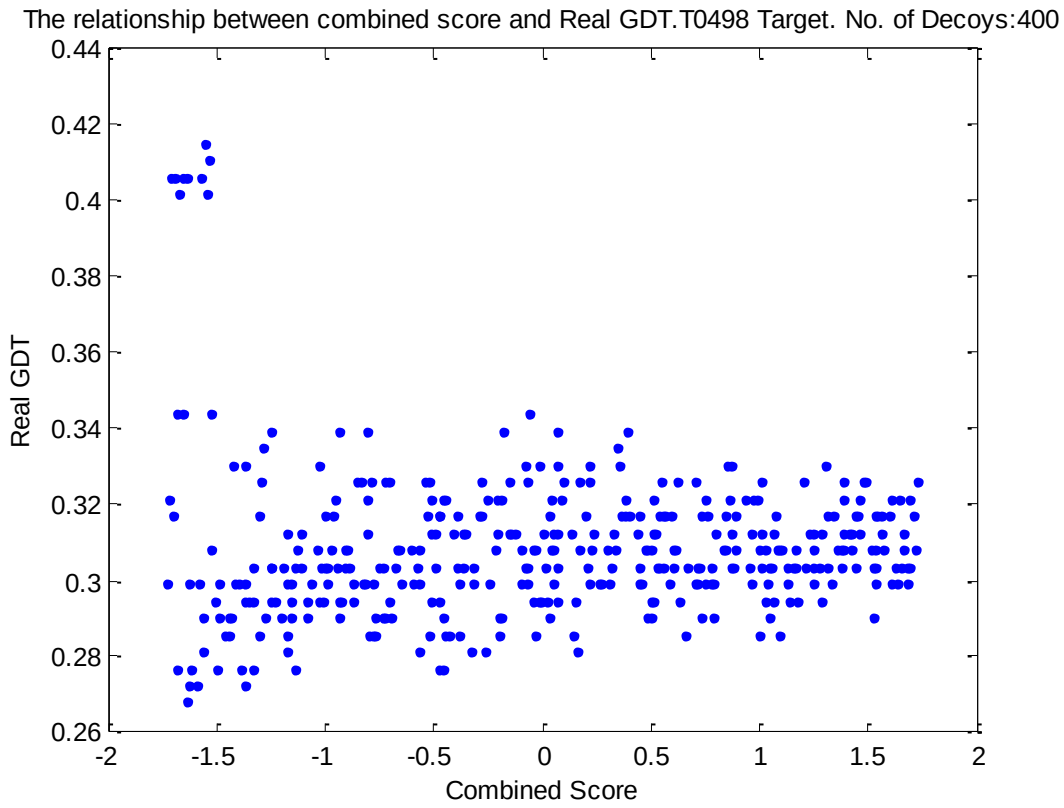

Figure 45. The Relationship between Combined Score and Real GDT on T0498 Target in Rosetta Data

In the above graph, figure 45, T0498 target decoys scores are plotted. It is considered to be a hard target in Rosetta Data. On the x-axis is the real GDT. On the y-axis is the combined score. It is scattered with real GDT. There are 400 decoys. Most of the decoys share the GDT score if we look at it from the side. Combined score fails to rank the best decoy in this target.

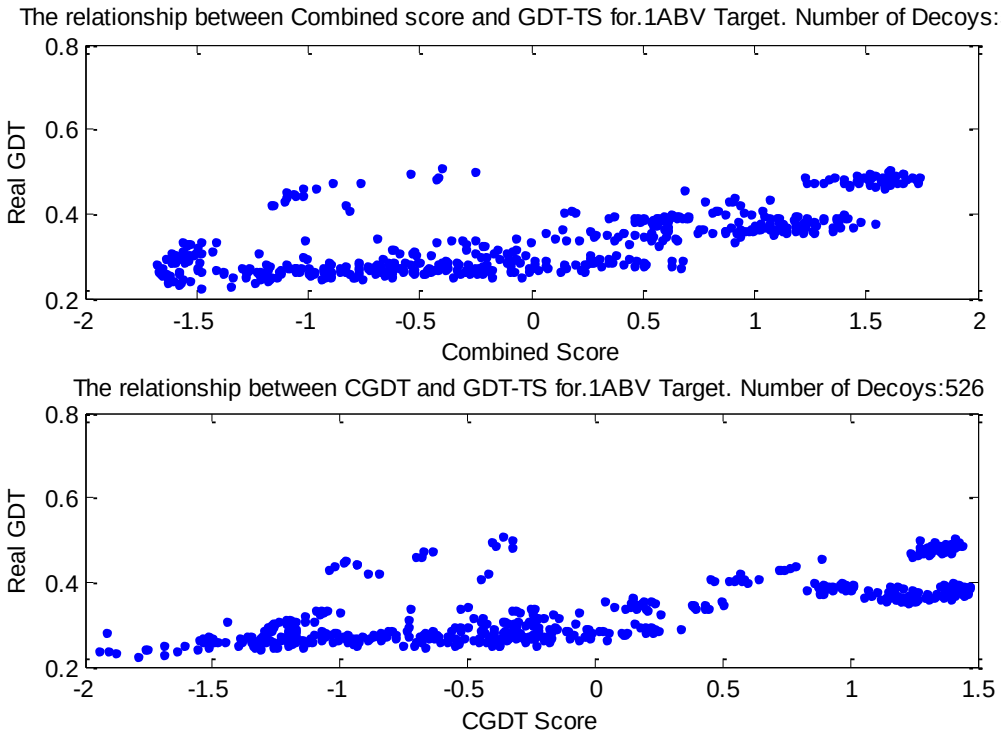

Figure 46. Comparison between combined score and CGDT in 1ABV from Zhang's Data.

In the previous graph, this is an example to show that combined score performed better than CGDT. Combined score chose a decoy that has GDT score better than the first decoy picked by CGDT. In the upper graph, combined score picked a decoy had 0.48. However, CGDT chose the first decoy that had 0.39. CGDT cannot differentiate between different clusters whether they are similar to each other or not. For example, looking from y-axis, there are two clusters at the end of the graph which are clustered based on the real GDT However, looking from x-axis, there is only one cluster at the end of the graph.

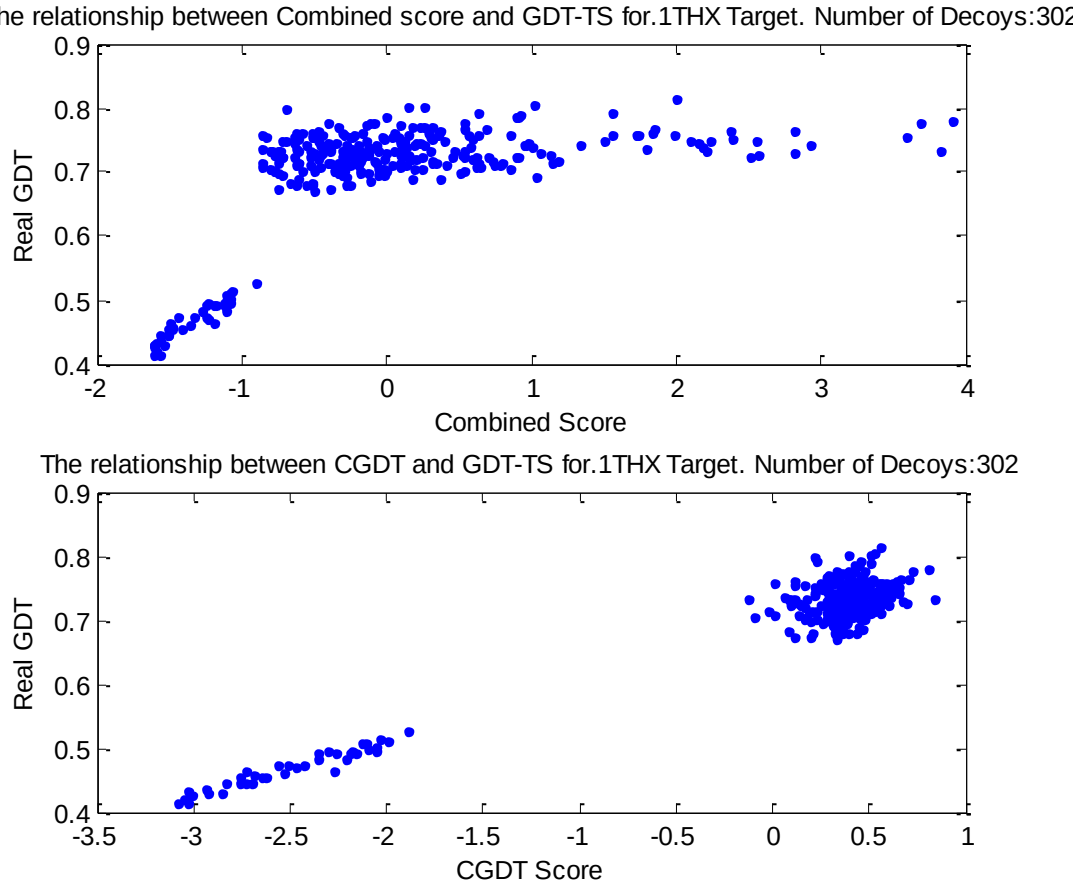

Figure 47. Comparison between combined score and CGDT in 1THX from Zhang's Data.

In the previous graph, combined score also performed better than CGDT. The first decoy was chosen by combined score had 0.78. However, the first decoy was chosen by CGDT had 0.73. The interesting note is that CGDT had really dense cluster which seems that this target is an easy target to be predicted.

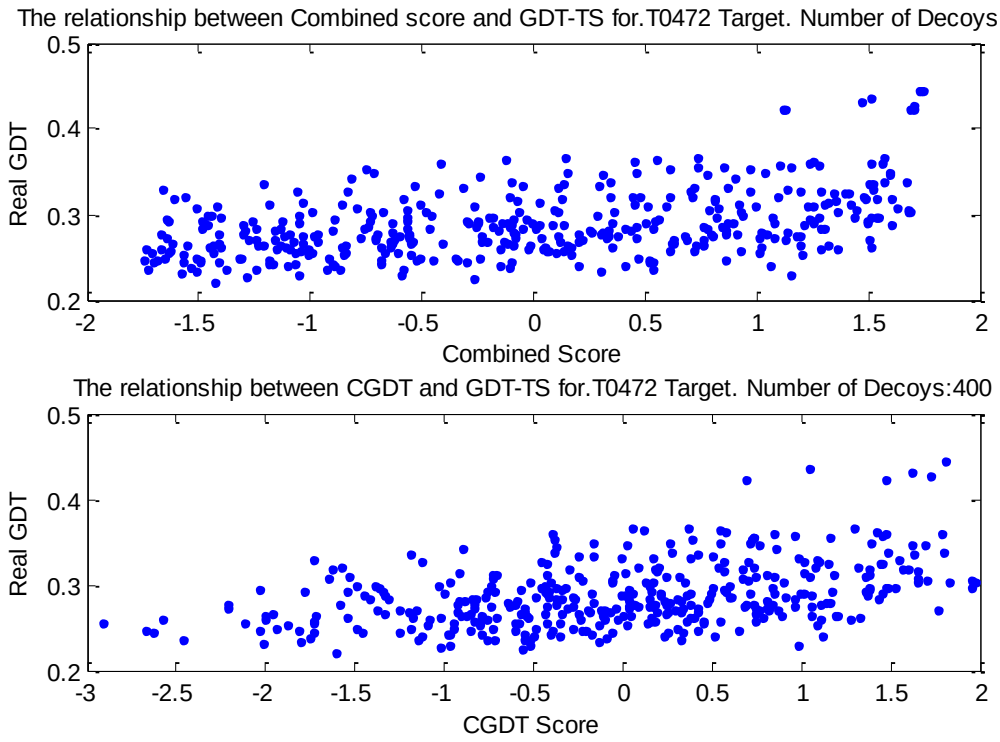

Figure 48. Comparison between combined score and CGDT in T0472 from Rosetta Data.

In the previous graph, the first decoy chosen by combined score had 0.44. However, the first decoy chosen by CGDT had 0.3. The ranges are different. That tells that CGDT could give a low score for decoys far from where the density is. Another notice is that CGDT seems to be denser than combined score.

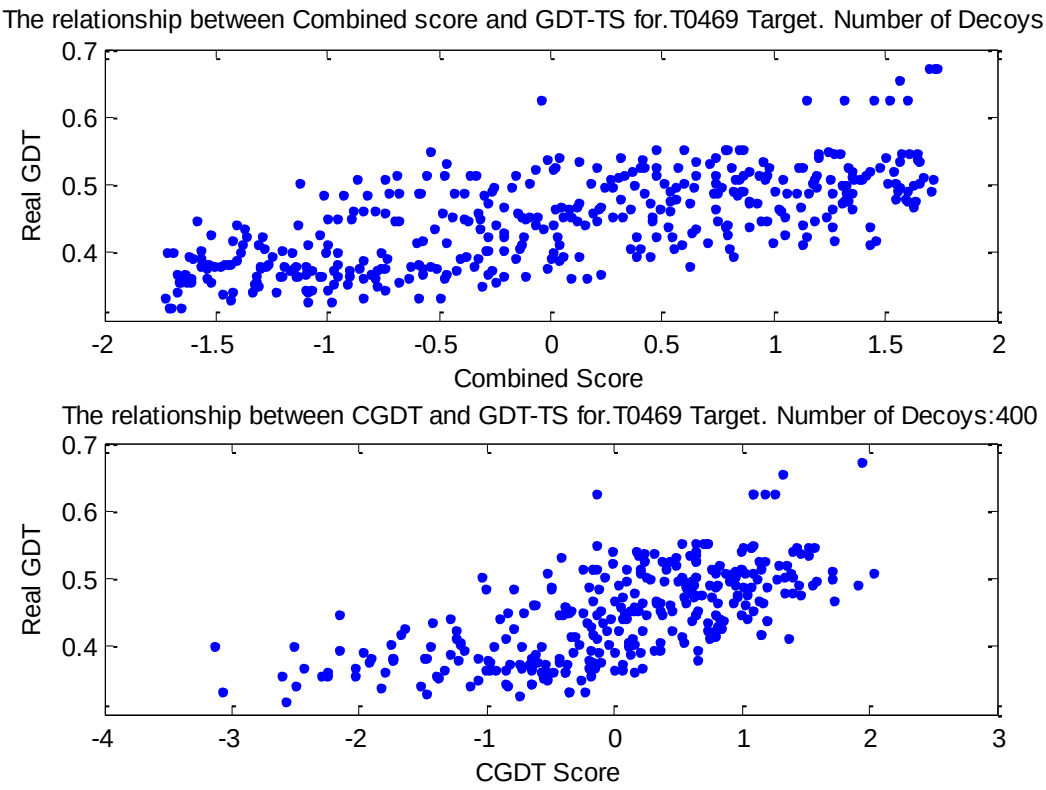

Figure 49. Comparison between combined score and CGDT in T0469 from Rosetta Data.

In the above graph, combined score performed well. The first decoy chosen by combined score had 0.67 GDT score. However, the first decoy chosen by CGDT had 0.50. Also, CGDT has a wider range and denser in the middle.

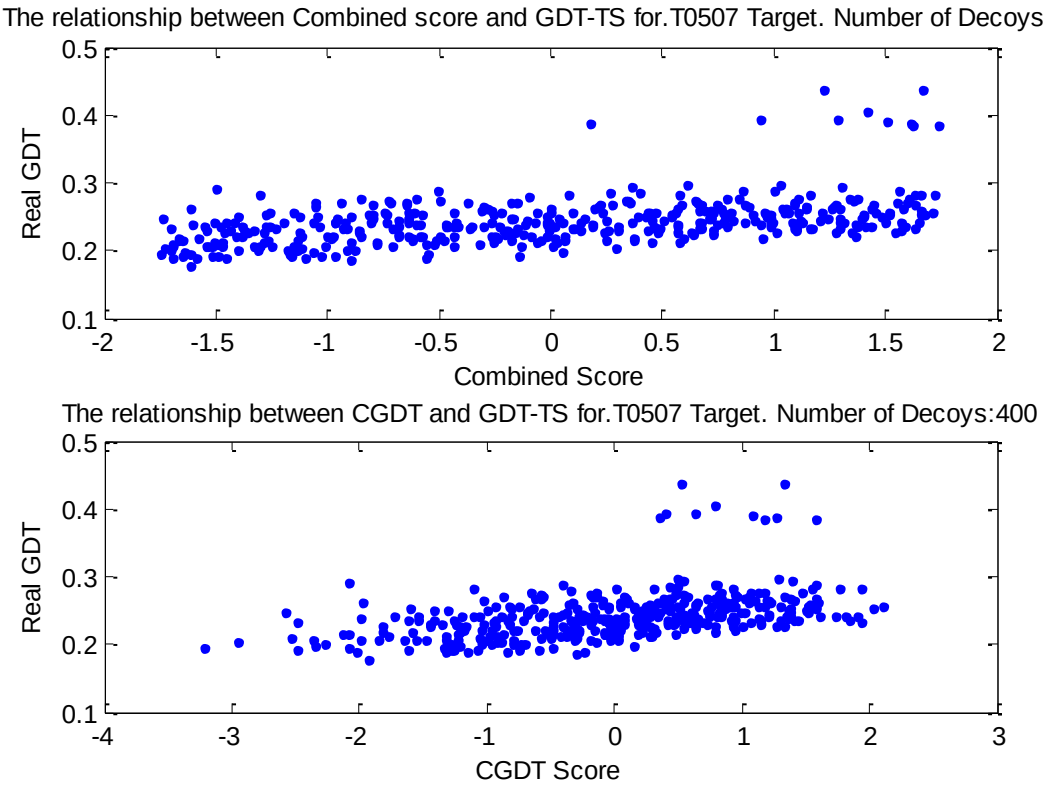

Figure 50. Comparison between combined score and CGDT in T0507 from Rosetta Data.

In the previous graph, the first decoy chosen by combined method had 0.38 GDT score. However, the first decoy chosen by CGDT had 0.26. Also, the same notice, CGDT seems to have a wider range and denser.

Future Work

The main factors helping combined score performs better than CGDT score were the CGDT score itself and the 4-mer sequence SPSP as features in training data. The main challenge is redundancy when the data generated as a pair-wise matrix for all the decoys. Removing those redundant observations more efficiently, not only from the training data but also from testing data, will improve performance significantly. Also, using regression methods might help improve performance better than classification methods. Although the performance of consensus methods was not as good as expected, the consensus methods still proved helpful in combination method. Adding more features as scores might help improve the overall scores.

CombinedMethod can be improved properly by improving some machine learning aspects such as taking the number of positive records and the number of negative records to be almost equal for training datasets. It helps the model to learn better. As a result, the performance may become better on the testing data. Also, setting different parameters for neural network should be considered.

Target classification before prediction shows how to deal with different kinds of classes. For example, easy targets should not be treated as hard targets or medium targets. Using different methods for different levels of target difficulty helps improve the score. Setting some criteria for sequence alignment search is one way to do the classification. For example, setting E-value <0.005 with BLAST hit gives an easy target case. Other hits are medium cases. No hits are hard targets.

Trying different approaches in machine learning might add some improvement. Machine learning should not be restricted to classification methods such as neural network or SVM. Other worthwhile machine learning approaches are as regression, such as SVM regression, k nearest neighbor regression, and regression by discretization based on random forest.

Deleting redundant decoys that have the same GDT score with other decoys will help the model learn better and get better results since the error will be minimized. For example, if there are three decoys that have a 0.4 GDT score, two of them should be removed for only training datasets.

Repeating CombinedMethod and adding the new combined score as a feature until convergence or until a fixed number of repetitions is achieved will improve the score. Also, deleting one feature, such as the lowest method performance, is better if the new CombinedMethod is added as a new feature.

On a larger scale, learning from a target and applying it to a similar target is another good research procedure. In this case, classifying the targets into classes needed to be done. In every class, there must be at least two targets. One target is for training; and the other is for testing, and vice versa. Classification can be done as a sequence similarity between the targets, recording sequence length, secondary structure, and solvent accessibility. If there is a target that has no other similar target in the dataset, a target from different datasets should be found for training data. In this way, the training and the testing will have something in common. Such target selection utilizing the

CombinedMethod will give higher weight to the method that fits those kinds of targets.

Chapter 5. Conclusion

Having only the prediction information for set of decoys for one protein, the challenge was how to measure a protein structure prediction, whether individually or in a set of decoys, and how to rank them and pick the best decoy which is closest to the native. Assessing a prediction is mainly based on two categories. First, the knowledge-based approach is based on previous knowledge from natives whether extracting a pattern that the natives have or characteristics in a particular angle or level since the proteins have a lot of factors they depend on. The knowledge-based class can measure the single decoys. The fact is that learning from natives and applying to decoys is not suggested because the decoys are different from natives. Second, the other class is based on consensus-based approach. In this case, it requires a set of decoys. Consensus method is a measure between the decoys in the same target, and it is a very successful method.

In this thesis, most of the methods were based on the consensus-based approach. For example, Single Position Specific Probability (SPSP) Score measures per position. PairScore measures the frequency of pair states in a set. Cons.Seq measures the decoys to the most frequent sequence. In the knowledge based approach, the general matrix was calculated based on native structures. Then, this matrix was applied on the decoys. For example, the method for calculating the probability of 4-mer sequence given Amino acid sequence is a 17 states by 20 amino acids matrix, and vice versa. Another

example, secondary structure sequence, is a 17*81 matrix. Those methods did not perform as well as expected.

This research developed a CombinedMethod approach to address the quality assessment problem for protein structure prediction. Methods based on 4-mer sequence did not get any improvement over single scores such as DFire and RW, but they contributed significantly in the combination method. Single Position Specific Probability (SPSP) Score for 4-mer sequence was based on consensus approach. It performed the best among all the other suggested methods based on 4-mer sequence. Using methods based on 4-mer sequence individually fails in model selection and does not perform better overall than single scores. However, the fact that 4-mer methods performed better in some targets including CGDT in some targets inspired the idea of using the combination method. Removing Single Position Specific Probability (SPSP) Score from the CombinedMethod consistently gives a significantly bad score--less than CGDT. In this way, Single Position Specific Probability (SPSP) Score based on 4-mer sequence is an important feature as complementary to the other scores in the CombinedMethod. Preprocessing the data helps achieve a higher score. For example, removing the observations that have GDT difference less than 0.01 contributed to the training of the model enabling it to learn better. As a result, CombinedMethod performed better than the state-of-the-art methods. Refining the method selection process can help researchers improve protein selection modeling, saving time and yielding better results in the future. Using the CombinedMethod proved to be a way to accomplish these goals in that it proved

to be more efficient. This finding can save time, yield better results, and reduce cost.

Bibliography

[1] Lazaridis, T. and M. Karplus. 1999. Discrimination of the native from misfolded protein models with an energy function including implicit solvation. *Journal of molecular biology* 288(3):477-487.

[2] Brooks, B. R., R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus. 1983. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.* 4:187–217.

[3] MacKerell, A. D., Jr., B. Brooks, C. L. Brooks III, L. Nilsson, B. Roux, Y. Won, and M. Karplus. 1998b. CHARMM: The energy function and its parameterization with an overview of the program. In *Encyclopedia of Computational Chemistry* (P. v. R. Schleyer et al., Eds.) Chichester, John Wiley & Sons. pp. 271–277.

[4] Pearlman, D. A., Case, D. A., Caldwell, J. W., Ross, W. R., Cheatham, T. W., DeBolt, S., Ferguson, D., Seibel, G., and Kollman, P. 1995. AMBER, a computer program for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to elucidate the structures and energies of molecules. *Comput. Phys. Commun.* 91:1–41.

[5] Harano, Y., R. Roth, Y. Sugita, M. Ikeguchi, and M. Kinoshita. 2007. Physical basis for characterizing native structures of proteins. *Chem. Phys. Lett.*, 437:112-116.

[6] Duan, Y., and P. A. Kollman. 1998. “Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution.” *Science* 282: 740–744.

[7] Miyazawa, S., and R. L. Jernigan. 1996. Residue–residue potentials with a favorable contact pair term and an unfavorable high packing density term. *J. Mol. Biol.* 256:623–644.

[8] Wodak, S.J., and M. J. Roomanm. 1993. Generating and testing protein folds. *Curr. Opin. Struct. Biol.* 3:247–259.

[9] Sippl, M.J. 1995. Knowledge-based potentials for proteins. *Curr. Opin. Struct. Biol.* 5:229–235.

[10] Lemer, C.M.R, M. J. Rooman, and S. J. Wodak. S.J. Protein-structure prediction by threading methods—Evaluation of current techniques. *Proteins* 23:337–355.

[11] Jernigan, R.L., and I. Bahar, I. 1996. Structure-derived potentials and protein simulations. *Curr. Opin. Struct. Biol.* 6:195–209.

[12] Goldstein, R., Z. A. Luthey-Schulten, and P. G. Wolynes. 1992. Protein tertiary structure recognition using optimized Hamiltonians with local interactions. *Proc. Natl. Acad. Sci. USA* 89:9029–9033.

[13] Maiorov, V. N., and G. M. Crippen. 1992. Contact potential that recognizes the correct folding of globular proteins. *J. Mol. Biol.* 227:876–888.

[14] Thomas, P.D., and K. A. Dill. 1996a. An iterative method for extracting energy-like quantities from protein structures. *Proc. Natl. Acad. Sci. USA* 93:11628–11633.

[15] Tobi, D., G. Shafran, N. Linial, and R. Elber. 2000. On the design and analysis of protein folding potentials. *Proteins* 40:71–85.

[16] Vendruscolo, M., and E. Domanyi. 1998. Pairwise contact potentials are unsuitable for protein folding. *J. Chem. Phys.* 109:11101–11108.

[17] Vendruscolo, M., R. Najmanovich, and E. Domany. 2000. Can a pairwise contact potential stabilize native protein folds against decoys obtained by threading? *Proteins* 38:134–148.

[18] Bastolla, U., J. Farwer, E. W. Knapp, and M. Vendruscolo. 2001**.** How to guarantee optimal stability for most representative structures in the protein data bank. *Proteins* 44:79–96.

[19] Dima, R.I., J. R. Banavar, and A. Maritan. 2000. Scoring functions in protein folding and design. *Protein Sci.* 9:812–819.

[20] Micheletti, C., F. Seno, J. R. Banavar, and A. Maritan. 2001. Learning effective amino acid interactions through iterative stochastic techniques. *Proteins* 42: 422–431.

[21] Dobbs, H., E. Orlandini, R. Bonaccini, and F. Seno. 2002. Optimal potentials for predicting inter-helical packing in transmembrane proteins. *Proteins* 49:342–349.

[22] Hu, C., Li, X., and J. Liang. 2004. Developing optimal non-linear scoring function for protein design. *Bioinformatics* 20:3080–3098.

[23] Tanaka, S., and H. A. Scheraga. 1976. Medium- and long-range interaction parameters between amino acids for predicting three-dimensional structures of proteins. *Macromolecules* 9:945–950.

[24] Miyazawa, S., and R. L. Jernigan. 1985. Estimation of effective interresidue contact energies from protein crystal structures: Quasi-chemical approximation. *Macromolecules* 18:534–552.

[25] Samudrala, R., and J. Moult. 1998. An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction. *J. Mol. Biol.* 275:895–916.

[26] Lu, H., and J. Skolnick. 2001. A distance-dependent atomic knowledge-based potential for improved protein structure selection. *Proteins* 44:223–232.

[27] Sippl, M. J. 1990. Calculation of conformational ensembles from potentials of the main force. *J. Mol. Biol.* 213:167–180.

[28] Zhou, H., and Y. Zhou, Y. 2002. Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction. *Protein Sci.* 11:2714–2726.

[29] Nishikawa, K., and Y. Matsuo. 1993. Development of pseudoenergy potentials for assessing protein 3-D–1-D compatibility and detecting weak homologies. *Protein Eng.* 6:811–820.

[30] Singh, R.K., A. Tropsha, and I. I. Vaisman. 1996. Delaunay tessellation of proteins: Four body nearest-neighbor propensities of amino acid residues. *J. Comput. Biol.* 3:213–221.

[31] X. Li and J. Liang. Computational design of combinatorial peptide library for modulating protein-protein interactions. *Pacific Symposium of Biocomputing*, pages 28–39, 2005.

[32] Samudrala, R., and J. Moult. 1998. An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction. *J. Mol. Biol.* 275:895–916.

[33] McConkey, B.J., V. Sobolev, and M. Edelman. 2003. Discrimination of native protein structures using atom-atom contact scoring. *Proc. Natl. Acad. Sci. USA* 100:3215–3220.

[34] Li, X., C. Hu, and J. Liang. 2003. Simplicial edge representation of protein structures and alpha contact potential with confidence measure. *Proteins* 53:792– 805.

[35] Mirny, L.A., and E. I. Shakhnovich. 1996. How to derive a protein folding potential? A new approach to an old problem. *J. Mol. Biol.* 264:1164–1179.

[36] Bastolla, U., J. Farwer, E. W. Knapp, and M. Vendruscolo. 2001. How to guarantee optimal stability for most representative structures in the protein data bank. *Proteins* 44:79–96.

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

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

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

[40] Zhou, H., and Y. Zhou. 2002. Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction*, Protein Sci*. 11:2714–2726.

[41] Zhang, J, and Y. Zhang. 2010. 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.

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

[43] Cheng, J., Z. Wang, A. N. Tegge, and J. Eickholt. 2009. Prediction of global and local quality of CASP8 models by MULTICOM series. *Proteins*, 77 Suppl., 9:181-184.

[44] Benkert P, S. C. Tosatto, and T. Schwede. 2009. Global and local model quality estimation at CASP8 using the scoring functions QMEAN and QMEANclust. *Proteins*, 77 Suppl., 9:173-180.

[45] Wallner B, and A. Elofsson. 2007. Prediction of global and local model quality in CASP7 using Pcons and ProQ. *Proteins*, 69 Suppl, 8:184-193.

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

[47] Wang, A., A. N. Tegge, and J. Cheng. 2008. Evaluating the absolute quality of a single protein model using structural features and support vector machines,. *Proteins: Structure, Function, and Bioinformatics* 75:638–647

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

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

[50] Zemla, A. 2003. LGA: A method for finding 3D similarities in protein structures. *Nucleic Acids Res* 31(13):3370–3374.

[51] Zhang, Y., and J. Skolnick. 2004. Scoring function for automated assessment of protein structure template quality, P*roteins: Structure, Function, and Bioinformatics*. 57:702–710.

[52] McGuffin L. J. 2011. Benchmarking consensus model quality assessment for protein fold recognition. *BMC Bioinformatics* 8(1)**:**345

[53] He, Zhiquan. 2011. Protein structural model selection based on protein-dependent scoring function. *Statistics and Its Interface,* Vol. 0 (2011):1–7.

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

[55] Cozzetto, D, and A. Tramontano A. 2008. Advances and pitfalls in protein structure prediction. *Curr Protein Pept Sci* 9(6):567-577.

[56] Petrey, D, and B. Honig. 2005. Protein structure prediction: inroads to biology. *Mol Cell* 20(6):811-819.

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

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

[59] Kihara D, H. Chen, and Y. D. Yang. 2009. Quality assessment of protein structure models. *Curr Protein Pept Sci* 10(3):216-228.

[60] Lazaridis, T., and M. Karplus. 1999. Discrimination of the native from misfolded protein models with an energy function including implicit solvation. *J Mol Biol* 288(3):477-487.

[61] Petrey, D. and B. Honig. 2000. Free energy determinants of tertiary structure and the evaluation of protein models. *Protein Sci* 9(11):2181-2191.

[62] Benkert, P, S. C. Tosatto, and T. Schwede. 2009. Global and local model quality estimation at CASP8 using the scoring functions QMEAN and QMEANclust. *Proteins,* 77 Suppl. 9:173-180.

[63] Cheng, J, Z. Wang, A. N. Tegge, and J. Eickholt. 2009. Prediction of global and local quality of CASP8 models by MULTICOM series. *Proteins*, 77 Suppl 9:181-184.

[64] Wallner, B, and A. Elofsson. 2007. Prediction of global and local model quality in CASP7 using Pcons and ProQ. *Proteins* 2007, 69 Suppl 8:184-193.

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

[66] Qiu, J, W. Sheffler W, D. Baker D, and W. S. Noble. 2008. Ranking predicted protein structures with support vector regression. *Proteins* 71(3):1175-1182.

[67] Kim, D. E., D. Chivian, and D. Baker. 2004. Protein structure prediction and analysis using the Robetta server. *Nucleic Acids Res*., 32(Web Server issue):W526-531.

[68] Simons K.T., R. Bonneau, I. Ruczinski, and D. Baker. 1999. Ab initio protein structure prediction of CASP III targets using ROSETTA. *Proteins,* Suppl. 3:171-176.

[69] Roy A, Kucukural A, Zhang Y: I-TASSER: a unified platform for automated protein structure and function prediction. *Nat Protoc 2010,* 5(4):725-738.

[70] Xu, Y., D. Xu, and J. Liang (Eds.). 2007. Computational methods for protein structure and modeling, *Springer,* Berlin, ISBN: 978-1-4419-2206-9.

[71] Zheng WM, Liu X. A protein structural alphabet and its substitution matrix CLESUM. In: Priami C, Zelikovsky A, editors. Lecture notes in Bioinformatics 3680. Berlin: *Springer* Verlag; 2005. pp 59–67.

[72] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; *SIGKDD Explorations*, Volume 11, Issue 1.

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