MACHINE LEARNING METHODS FOR EVALUATING THE QUALITY OF A SINGLE PROTEIN MODEL USING ENERGY AND STRUCTURAL PROPERTIES

A Thesis

Presented to

The Faculty of the Graduate School

University of Missouri-Columbia

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

By

Junlin Wang

Dr. Yi Shang, Thesis Supervisor

July 2015

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

MACHINE LEARNING METHODS FOR EVALUATING THE QUALITY OF A SINGLE PROTEIN MODEL USING ENERGY AND STRUCTURAL PROPERTIES

Presented by Junlin Wang,

A candidate for the degree of

Master of Science

_____

Professor Yi Shang

_____

Professor Dong Xu

_____

Professor Ioan Kosztin

ACKNOWLEDGEMENT

Firstly, I am sincerely presenting my gratitude to my advisor, Dr. Yi Shang for his continuous support and guiding during my program in University of Missouri Columbia. Without his enlightening instruction, strictness and patience, I could not have completed my thesis and research work. It is my great honor to be his student and learn from him.

I would also like to thank Dr. Dong Xu and Dr. Ioan Kosztin not only for being my committee members but also their valuable advice on my research and thesis. Their suggestions helped me find the basic idea and finally became my thesis.

I would also like to thank my colleagues in my lab, they have always been very helpful and they gave me important ideas about my thesis. It is my greatest pleasure to work with them.

Finally, I would like to thank my parents and my girlfriend who loves me and gives me strength, many people made my story colorful, but they made me here to start it.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Computational protein structure prediction is one of the most important problems in bioinformatics. In the process of protein three-dimensional structure prediction, assessing the quality of generated models accurately is crucial. Although many model quality assessment (QA) methods have been developed in the past years, the accuracy of the state-of-the-art single-model QA methods is still not high enough for practical applications. Although consensus QA methods performed significantly better than single-model QA methods in the CASP (Critical Assessment of protein Structure Prediction) competitions, they require a pool of models with diverse quality to perform well.

In this thesis, new machine learning based methods are developed for single-model QA and top-model selection from a pool of candidates. These methods are based on a comprehensive set of model structure features, such as matching of secondary structure and solvent accessibility, as well as existing potential or energy function scores. For each model, using these features as inputs, machine learning methods are able to predict a quality score in the range of [0, 1]. Five state-of-the-art machine learning algorithms are implemented, trained, and tested using CASP datasets on various QA and selection tasks. Among the five algorithms, boosting and random forest achieved the best results overall. They outperform existing single-model QA methods, including DFIRE, RW and Proq2, significantly, by up to 10% in QA scores.

CHAPTER 1: INTRODUCTION

The three dimensional (3D) structure of a protein is essential on studying its functions [1]. As the progress of genome sequencing program, the number of protein sequence data growing rapidly [2]. Nowadays, X-ray crystallography is the most widely used experimental method on 3D structures of protein sequences. Besides, the electron microscopes and the nuclear magnetic resonance (NMR) are also useful experimental methods. However, all of these experimental methods are costly, difficult and time consuming [3]. Comparing with these experimental methods, computational methods can generate numerous alternative models for a given protein sequence in limit amount of time [4]. Therefore, it is a challenging task in bioinformatics on getting a 3D structure of a protein base on only its sequence information. Since computational methods always generate large numbers of alternative models, then how to pick up the best model from the model pool becomes one of the most important aspects. [5]

The Critical Assessment of protein Structure Prediction (CASP) is a biennial worldwide contest aimed at establishing the current state of the art in protein structure prediction. A plenty of protein structure prediction methods and quality assessment (QA) methods have been tested there after developed [6,7]. For most of the prediction methods, like I-TASSER [8] and Rosetta [9] prefer to utilize the sampling-and-selection strategy, which means they will generate large amount of 3D

structures at first, and then use quality assessment methods to screen out the most protein-like model. In order to get the best model among the large number of alternatives in the pool, an effective method to evaluate the qualities of given 3D structures also matters.

The quality methods are mainly divided into two categories: single-model QA methods and consensus-based QA methods. Consensus-based methods perform really successfully in many targets of CASP [10,11], when most of the models in the pool are similar to native structure, the correlation between consensus and true score will be very good. However, there are also limitations for the consensus-based methods. In some circumstances, when the poor model is overwhelming in the pool, or the structure of a sequence is hard to be determined, most of the models in the pool are not similar to each other, consensus idea will no longer works properly. Also, if the pool is really small (or even sometimes we just want to evaluate the quality of one model), or the models in the pool are quite similar, then consensus-based methods are always not workable to choose the best model. In terms of the single-model methods, the commonly used ones are potential functions and machine learning methods, which is very popular in recent years. Potential functions include physics-based potential functions and knowledge-based potential. Physics-based potential functions can be derived from the laws of physics, such as AMBER [12] and CHARMM [13]. Knowledge-based potential will generate a model based on statistical knowledge of the solved protein in a special database or a PDB library. DFIRE [14], DOPE [15] and

RAPDF [16] are all Knowledge-based methods. In some single-model QA methods, structure properties are utilized to improve the performance. By combining the statistical potential with structure properties, some single-model got improvement and good performance, like Proq2 [17].

In recent years, machine learning methods have been more and more popular on QA problem, such as support vector machine (SVM), neural network (NN) and random forest (RF) [18-20]. Actually, machine learning methods cannot be separated from other methods since many knowledge-based potential functions are used as the features of machine learning functions. Besides of these functions, other attributes extracted from sequences and structures of protein are also used as input features. Based on these inputs, programs try to study the "rules" and generate a model for evaluation, to display the quality of a model.

In this work, we tested five popular machine learning methods to evaluate the quality of a single model. Besides of the three methods mentioned above, decision tree and boosting will also be discussed. For the using feature set, energy knowledge and structure knowledge are combined together, which means statistical potential terms, secondary structures and solvent accessibility are all utilized to collect more information about the model and sequence. In this program, Proq2, Dope, RW, RAPDF, DDFIRE, DFIRE and OPCU-Cα are all used as inputs, and we also generate 5 features and 4 features based on secondary structure and solvent

accessibility. GDT-TS score and TM-score will be used as labels, and a score between 0 and 1 will be used to rank the quality of the models.

In Chapter 2 of this thesis, features extraction methods and related machine learning methods will be introduced. Chapter 3 will focus on the evaluation of parameters for each algorithm, Chapter 4 describes the experimental results compared with other QA methods. Conclusions and future work are shown in Chapter 5

CHAPTER 2: RELATED WORK

2.1 Protein Quality Evaluation

To solve the problem of selecting the most suitable candidates from a model pool, many different kinds of methods were developed. The QA methods using today can be separate into two main classes. The first group of methods just focus on the knowledge of a model itself, which is called single model QA methods, other methods use a model cluster base on the theory that good models have more structural neighbors or more similarity with each other, these methods are multiple based QA methods.

2.1.1 Consensus-based Method

The pairwise similarity could be retrieved from the root-mean-squared deviation (RMSD), the template Modeling Score (TM-score) and the total score of global distance test (GDT-TS) between each protein model pairs.

RMSD [21] is the used to calculate the average distance between the C-α atoms of two protein models. It can be used to measures the similarity in three-dimensional structure of the C-α atomic coordinates. Low RMSD score means the C-α atomic are similar, which indicates two protein models are similar. Given two sets of n points v and w, the RMSD is defined as follows:

$$RMSD_{(v,m)} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\|v_i - w_i\|^2}$$

$$= \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left((v_{ix} - w_{ix})^2 + \left(v_{iy} - w_{iy}\right)^2 + (v_{iz} - w_{iz})^2\right)}$$

TM-score [22, 23] is an algorithm to measure the similarity of topologies between two protein models. TM-score is considered as the close matches more than the distant matches. The range of TM-score is (0,1) for each model, where 1 means a identical match between two protein models. The models in test dataset will be compared with the native structure using TM-score, then the similarity between the model and native structure will be used as its quality.

GDT-TS [24] is a global quality measure of the correct positioning of amino acid sequences between two protein structures. It is the most common used parameter on checking the quality of a model. It is calculated by averaging percentage of residues with C-α atom distance in the model structure within certain distance cutoff of their positions. GDT-TS has range from 0 to 1, higher value indicating better accuracy. If the value is 1, it means the structures being compared are the same. The GDT is defined as follow:

$$GDT - TS(s_i, s_j) = \frac{(P_1 + P_2 + P_3 + P_4)}{4}$$

Where s_i and s_j are two protein 3D structures and P_d is the percentage of amino acid residues' alpha carbon atoms from s_i that can be superimposed with

corresponding residues from s_j within a defined distance cutoff d, $d \in \{1, 2, 4, 8\}$ [25].

2.1.2 Energy or Scoring Function

Proq2 is known as the best single-model method inCASP10, which could be used to evaluate the model's quality. Proq2 uses support vector machines to predict local as well as global quality of protein models. Its improved performance is obtained by combining the features used by its predecessor, Proq2, with updated structural and predicted features. The largest performance increase in local prediction accuracy is obtained by using the predicted and actual secondary structure and solvent accessibility. The using of profile weighting and the information per position from PSSM also helps improving the performance of new Proq2 [17].

OPCU-Cα is a knowledge-based potential function, only uses the knowledge of $C\alpha$ position [26]. Pseudo-positions artificially built from a $C\alpha$ trace for auxiliary purposes were used to establish the contributions from other atomic positions. Seven major representative molecular interactions in proteins including distance-dependent pairwise energy with orientation preference, hydrogen bonding energy, short-range energy, packing energy, tri-peptide packing energy, three-body energy, and solvation energy were used to calculate the potential function.

DFIRE [14] is a statistical energy function built on distance-scaling, which is a reference state of uniformly distributed ideal gas points, and the statistics of the distance between two atoms in known protein structures. dDFIRE [14] is short for a

dipolar DFIRE, which adds orientation dependence. dDFIRE separate polar atoms from nonpolar atoms by define carbon atoms as nonpolar atoms. Polar atoms include nitrogen and oxygen atoms in all residues, and sulfur atom in Cys. The orientation of a dipole is mimicked by polar atom's reference direction, and the function is base on the distance between two atoms and the orientation angles in the orientation of dipoles. By using dDFIRE defined reference directions and orientations, the positions of hydrogen atoms are not required, the function can be developed for heavy atoms only.

RW [27] was developed by Zhang's lab, it is a new distance-dependent atomic potential using a random-walk ideal as the reference state. A side-chain orientation-dependent term generated base on a non-redundant high-resolution structural database is also added into RW.

DOPE (Discrete Optimized Protein Energy) [15] is an atomic distance-dependent statistical potential extracted from a non-redundant set of 1472 crystallographic structures. DPOE doesn't depend on any parameters but accounts for the finite and spherical shape of the native structures in a homogeneous sphere, the radius if the sphere depends on a sample native structure.

RAPDF [16] is an all-atom distance-dependent conditional probability discriminatory function, it will generate three discriminatory functions including two virtual atom representations and one all-heavy atom representation.

2.2 Machine Learning Methods

In this work, five popular machine learning methods including linear model, neural network, decision tree, random forest and boosting were used to do the regression.

2.2.1 Linear Model

Linear regression is a common used statistic method for modeling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variable) denoted X. If only one explanatory variable, it is called simple linear regression. For more than one explanatory variable case, it is called multiple linear regression [30]. Linear regression was the first type of regression analysis to be studied rigorously, and to be used extensively in practical applications [31]. This is because models which depend linearly on their unknown parameters are easier to fit than models which are non-linearly related to their parameters and because the statistical properties of the resulting estimators are easier to determine [32].

When used for prediction, linear regression can be used to fit a predictive model to an observed data set of y and X values. After developing such a model, if an additional value of X is then given without its accompanying value of y, the fitted model can be used to make a prediction of the value of y [32].

In our test, the input is more than one feature, which means each input X has more than one dimension, so it should be Linear Regression with multiple variables. Let n be the number of features, then each input x got $(n+1)$ dimensions $[x_0 \dots x_n]$. Let $\{\theta\}$ be the parameters of each feature, $x^{(i)}$ be the input of i^{th} training example,

$x_j^{(i)}$ be the value of feature j in i^{th} training example, $x_0 = 1$, then $h_\theta(X)$ is defined as

$$h_\theta(X) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \cdots + \theta_n x_n$$

Let m be the size of the observed data set, the cost function is defined as

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

In order to minimize $J(\theta)$, for the size of the training data set, update θ_j for $j = 0, \ldots, n$ simultaneously with learning rate α,

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

Repeat update θ until convergence.

2.2.2 Decision Tree

Decision tree can builds regression or classification models with a form of a tree structure. It brakes down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. And finally result in a tree with decision nodes and leaf nodes. A decision node has two or more branches representing values of the attribute tested. Leaf node represents a decision on the numerical target. Decision trees can handle both categorical and numerical data [33].

The common used algorithm used to building decision trees is ID3 by J. R. Quinlan [34]. It will employs a top-down, greedy search through the space of possible branches with no backtracking. By replace gain information with Standard Deviation Reduction, ID3 can be used to generate a decision tree for regression. When decision

tree separate the data into different subsets, it will put tuples with similar values (homogenous) into one subset. It use standard deviation to calculate the homogeneity of a numerical sample, standard deviation equals zero means completely homogeneous [33].

Let S stand for standard deviation, x for a instance, n for the size of training dataset, and μ means the mean value, then S is defined as

$$S = \sqrt{\frac{\sum(x - \mu)^2}{n}}$$

And for more than one attributes,

$$S(Y, X) = \sum_{j \epsilon X} P(j) S(j)$$

Every time the dataset is split, the information change is calculated base on the decrease in standard deviation, the progress to build a decision tree is to find attribute that gets the highest standard deviation reduction.

$$SDR(Y, X) = S(Y) - S(Y, X)$$

The dataset will be split base on different strategy, the standard deviation reduction will be calculated for each strategy, then one with the largest standard deviation reduction is chosen for the decision node. After the decision node decided, the dataset will be split. By repeat this progress, the standard deviation will be continue decrease.

In practice, we can let the decision tree always growing or set some termination criteria. For example, when standard deviation for the branch becomes smaller than a certain fraction (e.g., 3%) of standard deviation for the full dataset or when too few

instances remain in the branch (e.g., %1 of the training dataset) [33]. If the tree grown with no limit, it may lead to overfitting, so a termination criteria is always needed, or we can do prune to the tree, in case that there are too many leaf nodes.

2.2.3 Neural Network

Artificial neural networks (ANNs) are a family of statistical learning algorithms inspired by biological neural networks (the central nervous systems of animals, in particular the brain) in machine learning and cognitive science. These algorithms are used to estimate functions base on a mass of inputs. Artificial neural networks are generally presented as systems of interconnected "neurons" which can compute values from inputs, and as they are adaptive nature, they are also capable of machine learning as well as pattern recognition [42].

In supervised learning, by given a set of instance pairs (x, y), and the aim is to find a function with $f(x) \rightarrow y$ that match the examples, that means the target is infer the mapping implied by the data. The cost function is related to the errors between our result and the data and it implicitly contains prior knowledge about the problem domain [42].

A commonly used cost is the mean-squared error, which tries to minimize the average squared error between the network's output $f(x)$, and the dependent variable y in the training dataset. When we try to minimize this cost using gradient descent for the class of neural networks called multilayer perceptron, then we will get the back propagation algorithm for training neural networks [42].

2.2.4 Boosting

Boosting is a machine learning ensemble meta-algorithm for reducing bias primarily and also variance [39] in supervised learning, and a family of machine learning algorithms that convert weak learners to strong ones [40]. When a classifier is only slightly correlated with the true classification (or same in regression case), it is consider as a weak learner. On the contrary, a learner arbitrarily well correlated with the true classification will be a strong learner [41]. Like bagging, boosting is a general approach that can be applied to many statistical learning methods for regression.

When boosting is used on decision tree, given a training dataset D of size n, let B be the tree number and λ be the shrunken rate, boost will keep generate new trees to revise the model, by using the result of the preceding trained tree to update the dependent variable y and the new fitted model, the algorithm is showed as follow,

First, set $f(x) = 0$ and $r_i = y_i$ for all the instances in training dataset.

For $b = 1, 2, \dots, B$, repeat:

a) Fit a decision tree f^b with d splits ($d + 1$ terminal nodes) to the training data (X, r)

b) Update f by adding in a shrunken version of the new tree,

$$f(x) \leftarrow f(x) + \lambda f^b(x)$$

c) Update the residuals r,

$$r_i \leftarrow r_i - \lambda f^b(x_i)$$

Get the final result of boosting model,

$$f(x) = \sum_{b=1}^{B} \lambda f^b(x)$$

Boosting will add the result of all the generated trees and gives the final result, but as the residuals r (at first equals to y) is smaller and smaller, the following trees will affect the result smaller, it is used for revise the model, but not all the model have the same weight like random forest.

2.2.5 Random Forest

The random forest algorithm is developed by Leo Breiman and Adele Cutler [34], it combines the idea of "bagging" and the random selection of features, it is an ensemble learning method for classification, regression and other tasks, that operate by constructing a large number of decision trees at training time and outputting the class that is the mode of the classes for classification case or of the individual trees mean prediction for regression case [35]. Random forests correct the overfitting problem of decision tree, and it provides the importance information of each input variable, which is suitable for information retrieving from a dataset of high dimension with noise [36], with these advantages, random forest became a state-of-the-art machine learning method and widely used so solve different biological problems [37,38].

By given a training dataset D of size n, random forest use bootstrap sample to generate different sample used to grow large number of unpruned regression tree. For growing m different decision trees, bootstrap sample will generate m new sample

datasets with size n by sampling from D uniformly and with replacement. And as mentioned before, for each node, t features will be randomly chosen and used to find the best split, which will which maximizes the information gain measure by Gini impurity (In ID3 it is the largest standard deviation). Decision trees will stop growing when reach the termination criteria. After repeat m times, a forest with m trees is generated.

The data not used for each tree in growing decision tree is called out of bag samples, which is used to estimate the error rate of the tree as well as the importance of each variable [36]. When used for prediction, all trees in the forest will be tested and give their own outputs, the average of outputs from all the trees will be used as the final result.

CHAPTER 3: MACHINE LEARNING METHODS FOR SINGLE MODEL QUALITY ASSESSMENT

3.1 Dataset Prepare

In this study, the targets of CASP8 (124) and CASP9 (117) are used to prepare the training dataset, including single domain targets and multiple domain targets, and some models were removed as they got errors when calculating secondary structure results. The final dataset contains 68403 (34266 from CASP8 and 34137 from CASP9) server models of 241 targets. All server models can be download from the CASP website (http://predictioncenter.org/download_area/).

3.2 Feature Extraction

In this work, 15 features were used, 7 from energy or scoring functions, 5 from secondary structure and 3 from solvent accessibility. These features are as follow:

3.2.1 Energy or Scoring Function (7 features)

For all the models in each target, seven single model QA software (Proq2, DDFIRE, DFIRE, DOPE, RW, RAPDF, OPCU-Cα) were used to generate seven scores, these scores will be used as seven features of the machine learning input.

3.2.2 Protein Secondary Structure (5 features)

For each target, the secondary structure of its sequence can be predicted, and for each 3D model, the secondary also can be calculated. Then the consistency between predicted and virtual secondary structures will be an important point to evaluate the

quality of a model. In this work, the secondary structure of models are calculated by DSSP [28], the predict results of sequences are calculate by PSIPRED [29].

For each model, the consistency of secondary structure elements (helix, strand and coil) between the result from DSSP and PSIPRED are calculated, and then converted into %helix, %sheet and %coil by dividing them by the total compared chain length, the result were used as three features. The formula is show as follow:

$$\%Helix\ (Sheet, Coil) = \frac{n}{length}$$

Which n means the number of matching Helix (Sheet, Coil) between DSSP and PSIPRED, length means the length of the sequence compared. The total matching of three secondary structures is also calculated and used as a feature:

$$\%Total\ = \frac{m}{length}$$

Which m means the number of matching of secondary structure between DSSP and PSIPRED.

For each amino acid position i, PSIPRED will give a confidence value C, which means how confident they say the predict is right, and by compare the secondary structure type S_p^i and S_d^i calculated by DSSP and PSIPRED, The secondary structure confidence score is defined as:

$$SC\ = \sum_{i\,=\,1}^{length} C\ *\ d(S_d^i, S_p^i)$$

Where $C \in \{0 \ldots 9\}$, $S_p^i, S_d^i \in \{H, E, C\}$ and $d(S_p^i,\ S_d^i)$ gives 1 if S_p^i and S_d^i are identical, otherwise 0. The result SC will be used as the fifth feature.

3.2.3 Solvent Accessibility (3 features)

The consistency of solvent accessibility between predicted and real is also important to indicate the quality of a model. The absolute solvent accessibility surface (ASAS) of each amino acid of a model is calculated by DSSP [28], the result is divided by their accessible surface area and got a ratio value. Except using this result, 0.2 is used as the threshold to determine if the amino acid is buried or exposed [43-45]. If the ratio value is lower than 0.2, it is regarded as a buried amino acid, otherwise it is exposed amino acid.

The predicted result that if an amino acid is buried or exposed can be given by SCRATCH [46,47], then match between predicted and the result calculated base on DSSP can be calculated as matching of buried amino acid, matching of expose amino acid and matching of solvent accessibility. By dividing them by the total compared chain length, we will get %buried, %exposed and %match solvent accessibility, and the value will be used as three features:

$$\%SA\ (Buried, exposed)\ = \frac{m}{length}$$

Where m means the number of matching and length means the length of the compared sequence.

3.3 Parameters Optimization of Machine Learning Methods

When using machine learning methods to solve different problems, it is important to do the parameters optimization. In general case, there are several statistical parameters can be tuned in order to improve the learning performance. For some of

the complex methods, the combination of parameters will lead to an obvious difference on its performance.

3.3.1 Linear Model

During the parameters optimization of linear model, interactions, stepwise and the combining of interactions and stepwise were tested. As alternative parameters, “linear” will just take 15 features as terms, “interactions” will also take the natural join of features as terms. Stepwise will add or remove terms one by one (choice of predictive variables is carried out by an automatic procedure based on f-test). By testing on three datasets, basic linear model result in the best performance, and it is the most stable machine learning method.

3.3.2 Decision Tree

In the optimization of the parameters for decision tree, we tried fitting a decision tree with tree prune and with out tree prune, and the features used every time split the dataset from one to fifteen. From the testing performance it seems decision tree with prune and more than ten features when branch got better result, decision tree use thirteen features each time split dataset got the best result.

3.3.3 Neural Network

As known, neural networks is the basic idea of deep learning, and the setting of network will affect the performance of the model. In this study, we mainly focus on the layer number, the size of each layer, the choosing of train function and the choosing of transfer function of each layer. For layer number, we just test one layer

and two layers as there are just 15 features, for the size of each layer from the set {10, 30, 50, 100}, three training functions including 'trainlm', 'trainbr', 'trainscg' is tested and three transfer function 'tansig', 'logsig', 'purelin' is used.

From testing, we found that network with 2 layers and size 100 got better performance, and train liner model (trainlm) as a most common used method gives better result. For the choosing of transfer function, combination {'logsig', 'logsig'} and {'logsig', 'tansig'} got the best result, and for all strategy, the performance using TM-score as label is always better than use GDT-TS score as label.

3.3.4 Boosting

The most important parameters affect the performance of boosting is the number of the trees, and the shrunken rate, in this work, we optimized the parameters in the following ranges: the number of trees from 100 to 1000 with the step of 100, and shrunken rate from 0.01 to 0.1 with the step of 0.01. After tested on the three test dataset, the performance with 800 and 900 trees and 0.09 shrunken rate is better than other combination, and the combination of higher tree number and shrunken rate is better than lower combination.

3.3.5 Random Forest

Several statistical parameters can be used to try to improve the performance of random forest, and among these parameters, the number of the trees in the forest m_{tree} and the number of features random chosen t_{chosen} for dataset split are the most important. In this study we optimized the parameters in the following sets:

$m_{tree} \in \{50, 100, 300, 500, 1000\}$, and $t_{chosen} \in \{1, \dots, 15\}$. Finally, random forest with more than 500 trees (from 500 to 1000 there is not a significant improve, the correlation is almost the same) and two features (some times one features and three features also got good results) result in a better performance on most times. The generated random forest model is vary large, the model with 500 trees can arrived more than 4 gigabyte, and from 500 trees to 1000 trees there is no significant improvement, so in order to save the memory resource, it is better to use less than 1000 as the size of the forest.

CHAPTER 4: EXPERIMENTAL RESULTS

This chapter presents the knowledge of the test dataset, list the result of the methods mentioned in chapter 2 and compare the result between different QA methods by illustrating two measures described in 4.2.

4.1 Data Set

Five machine learning methods will be tested on three test dataset, all these three dataset is from CASP10. The first dataset is CASP10 whole dataset, for totally 103 targets, each has more than 250 models from different server. For all the models in these targets, 15 features base on the model itself were calculated. Some models with errors on secondary structure that could not get DSSP [28] result will not be tested. The final test dataset one (D_1) include 26156 models from 103 targets.

The second test dataset is CASP10 stage one dataset (D_2). For this dataset, each target just got 20 models that can reflect the tendency of the whole model pool (test dataset one). 15 features base on the model itself were calculated for all these models.

The third dataset is CASP10 stage two dataset (D_3). For totally 103 targets, each target has 150 best models, which are selected from the whole dataset (test dataset one), Same as those two above, 15 features base on the model itself were calculated for all the models.

All the test dataset can be download from the CASP official web site: http://www.predictioncenter.org/download_area/CASP10/server_predictions/.

4.2 Evaluation Parameters

In this work, the performance of the five mentioned machine learning methods will be compared with DDFIRE, DFIRE, DOPE, OPCU-Cα, RW, Proq2, RAPDF and consensus based methods. The performance will be evaluated by two measures: the average of Pearson's correlation and the quality lost.

4.2.1 Pearson's Correlation

Pearson's correlation coefficient is computed as following:

$$P = \frac{\sum_{i=1}^{n}(x_i - \bar{x})\,(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

x is the predicted value and y is the true score (label), n is the number of the model and i is the model index.

4.2.2 Quality Score

Two different labels, the template Modeling Score (TM-score) and the total score of global distance test (GDT-TS) will be utilized.

4.2.3 Quality Lost

In this study, the Quality lost is defined as the difference between the real best model in the test dataset and the model chosen by the QA methods. After testing, the QA method will gives out a quality score for each model in the test dataset. Then we will choose a best model base on this predict quality score. In most of the cases, the model chosen based on the predicted quality score may not be the real best model in the pool. Also, the chosen model $m_{selected}$ has its own GDT-TS score $GDT-$

$TS_{selected}$ and TM-score value $TM_{selected}$ when comparing with the native structure. Then the quality lost can be defined as:

$$GDT - TS_{lost} = GDT - TS_{best} - GDT - TS_{selected}$$

$$TM_{lost} = TM_{best} - TM_{selected}$$

Where $GDT - TS_{best}$ and TM_{best} means the GDT-TS score and the TM-score of the real best model in the dataset.

4.3 Features' Correlation Coefficient

In order to show if the algorithm got some improvement base on the features used, we list the features' correlation coefficient with the labels for comparison.

Test dataset one (D_1):

Table 4.3.1 Features' Correlation Coefficient of CASP10 All model

Correlation Coefficient	DGT-TS label	TM-score label
Percentage of secondary structure	0.4206	0.4349
Percentage of Helix	0.1772	0.1672
Percentage of Sheet	0.5121	0.5102
Percentage of Coil	0.1418	0.1749
Consistence Score of Secondary Structure	0.4376	0.4485
Percentage of matching Solvent Accessibility	0.5308	0.5570
Matching of Bury Amino Acid	0.5037	0.5355
Matching of expose Amino Acid	0.3398	0.3451
DDFIRE	0.2653	0.2760

DFIRE (too many Inf)	0	0
DOPE	0.2027	0.1673
OPUS_CA	0.2993	0.2941
Proq2	0.4539	0.4404
RAPDF	0.2368	0.2546
RW	0.2564	0.2852

Test dataset one (D_2):

Table 4.3.2 Features' Correlation Coefficient of CASP10 Stage 1 Set20

Correlation Coefficient	DGT-TS label	TM-score label
Percentage of secondary structure	0.4382	0.4361
Percentage of Helix	0.2619	0.2428
Percentage of Sheet	0.3730	0.3676
Percentage of Coil	0.1275	0.1439
Consistence Score of Secondary Structure	0.4605	0.4559
Percentage of matching Solvent Accessibility	0.5895	0.5889
Matching of Bury Amino Acid	0.5002	0.5161
Matching of expose Amino Acid	0.2903	0.2641
DDFIRE	0.3219	0.3128
DFIRE	0.2885	0.2792
DOPE	0.2513	0.2237
OPUS_CA	0.3853	0.3716
Proq2	0.4947	0.4846
RAPDF	0.2872	0.2832
RW	0.3503	0.3590

Test dataset one (D_3):

Table 4.3.3 Features' Correlation Coefficient of CASP10 Stage 2 Set150

Correlation Coefficient	DGT-TS label	TM-score label
Percentage of secondary structure	0.3020	0.3125
Percentage of Helix	0.2744	0.2729
Percentage of Sheet	0.2431	0.2003
Percentage of Coil	0.0148	0.0256
Consistence Score of Secondary Structure	0.3164	0.3214
Percentage of matching Solvent Accessibility	0.3351	0.3562
Matching of Bury Amino Acid	0.2721	0.3087
Matching of expose Amino Acid	0.1726	0.1558
DDFIRE	0.3171	0.3142
DFIRE (too many Inf)	0	0
DOPE	0.2708	0.2465
OPUS_CA	0.3642	0.3518
Proq2	0.3292	0.3200
RAPDF	0.3087	0.3240
RW	0.2585	0.2838

Based on the correlation coefficient results showed above, we can see that in CASP10 all model dataset (dataset one) and QA stage one dataset (dataset two), secondary structure based features, solvent accessibility based features and Proq2 showed better performance. The common ground between these two datasets is that they all contain the knowledge of all the models from different servers. That means they include good, medium, and bad models in the datasets. Likewise, structure based features and Proq2 work well on jagged-quality models pool.

QA stage two dataset consist of 150 best models in dataset one, which means most of the models in this dataset got a better quality. When tested on QA stage two dataset (dataset three), the performance of secondary structure based features, solvent accessibility based features and Proq2 b are worse than dataset one. Since Proq2 generates the result in structure knowledge, which is one of the most primary improvements comparing with Proq, it can be infer that structure knowledge works well on separating good models and bad models. However, when facing similar quality models or similar structures, it may not be that persuasive.

4.4 CASP10 Model

This part shows the result of five machine learning methods when use dataset one as test dataset. The overview result will use GDT-TS and TM-score as label, which will be listed in table 4.4.1 and 4.4.2. Afterwards, the result will be comparing with consensus-based method and Proq2.

Table 4.4.1 Performance of Different QA Methods for CASP10 All Model Using GDT-TS Score as Label

	Gdtts Correlation	Gdtts lost	Find Best Models
Consensus	0.8948	0.0553	6
Linear	0.5716	0.0725	10
Decision Tree	0.4195	0.1123	4
Neural Network	0.3705	0.0902	3
Boosting	0.5746	0.0728	7
Random Forest	0.5977	0.0776	5

DDFIRE	0.2653	0.1421	2
DOPE	0.2027	0.1955	0
OPUS_CA	0.2993	0.1552	6
Proq2	0.4539	0.1063	10
RAPDF	0.2368	0.1517	2
RW	0.2564	0.1632	3

Table 4.4.2 Performance of Different QA Methods for CASP10 All Model Using TM-score as Label

	TM Correlation	TM lost	Find Best Models
Consensus	0.8923	0.0555	4
Linear	0.5862	0.0709	7
Decision Tree	0.4536	0.0986	4
Neural Network	0.3530	0.0944	5
Boosting	0.5831	0.0764	6
Random Forest	0.6046	0.0725	7
DDFIRE	0.2760	0.1454	2
DOPE	0.1673	0.2082	0
OPUS_CA	0.2941	0.1614	2
Proq2	0.4404	0.1140	9
RAPDF	0.2546	0.1567	2
RW	0.2852	0.1621	2

From table 4.4.1 and 4.4.2 we can see when tested on dataset one, consensus-based method is much better than all the single model QA methods on correlation. Even though some machine learning methods reached 0.07 on quality lost which is better than other traditional single QA methods, consensus still got better performance.

Column “Find Best Models” is an auxiliary measure for showing the ability of “best model directly picking up” from the model pool. Nevertheless, we can see consensus is not very strong on this aspect from this table.

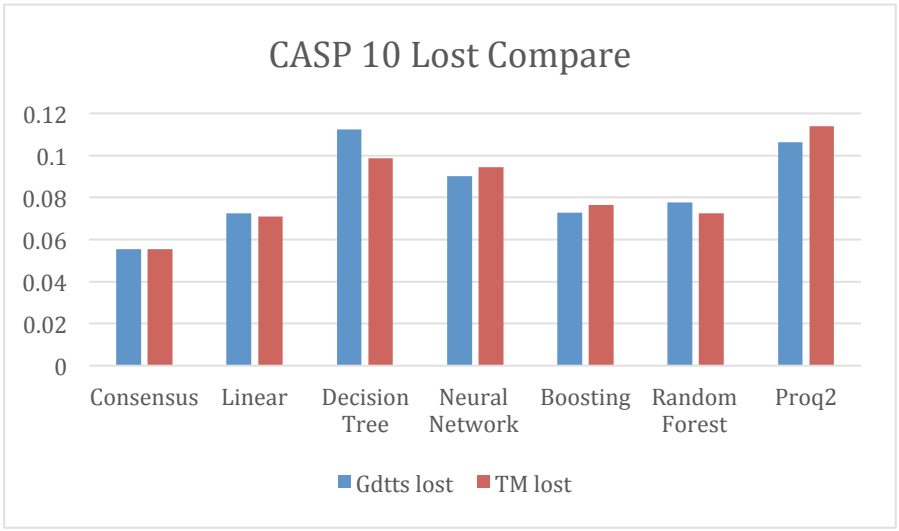

Figure 4.4.1 Quality Lost Performance of Different QA Methods for Dataset One

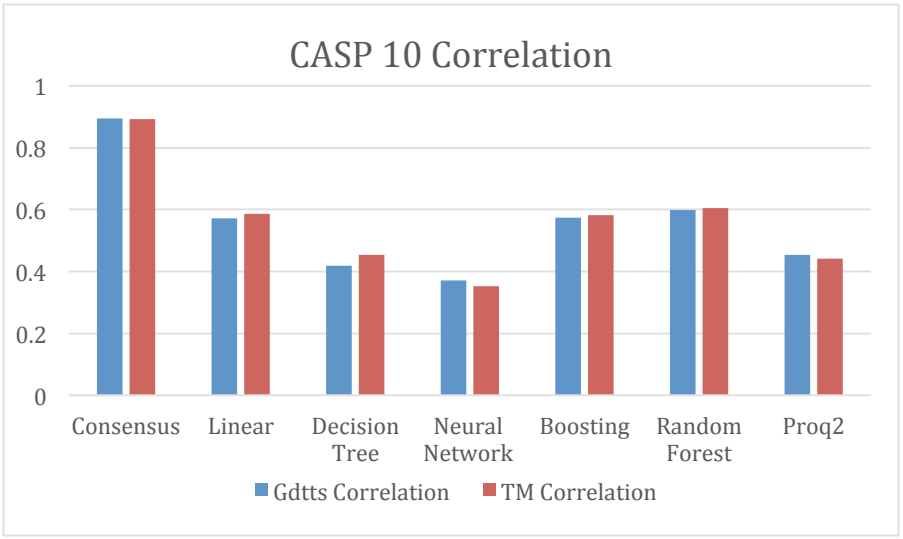

Figure 4.4.2 Pearson’s Correlation Performance of Different QA Methods for Dataset One

4.4.1 Linear Model

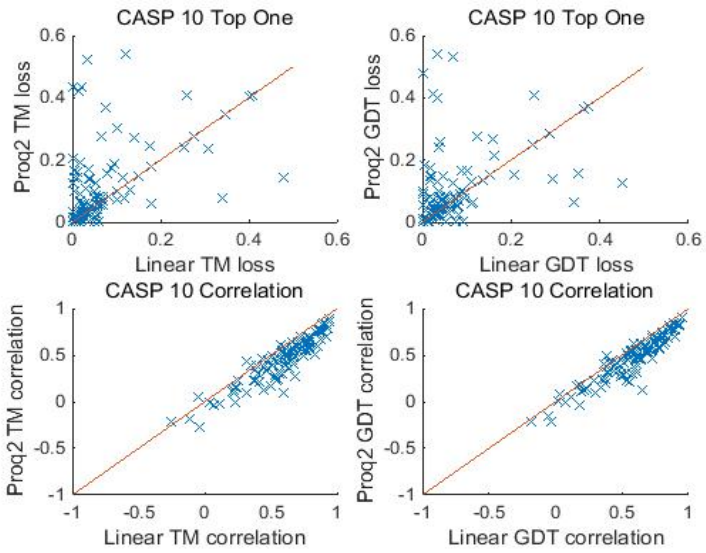

Figure 4.4.3 Compare the Result of Linear Model with Proq2 on Dataset One

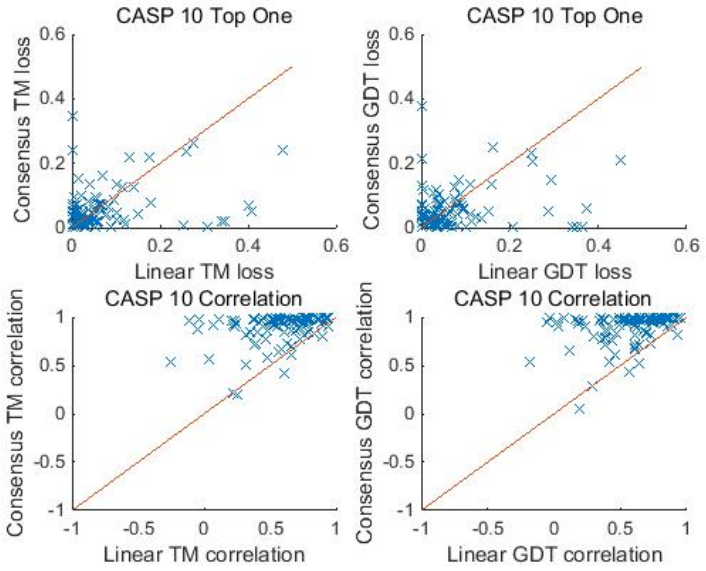

Figure 4.4.4 Compare the Result of Linear Model with Consensus-based Method on Dataset One

When tested on dataset one, linear model shows a comparable result (0.0725, 0.0709) with boosting and random forest on quality lost, which is better than Proq2

(0.1063, 0.1140), but worse than consensus-based method (0.0553, 0.0555). Among all the 103 targets, 45 (48) got lower GDT-TS (TM-score) lost using linear model.

The Pearson's correlation of linear model (0.5716, 0.5862) is 12 percent higher than Proq2, but still lower than consensus-based method. Actually, almost all of those targets are worse than consensus

4.4.2 Decision Tree

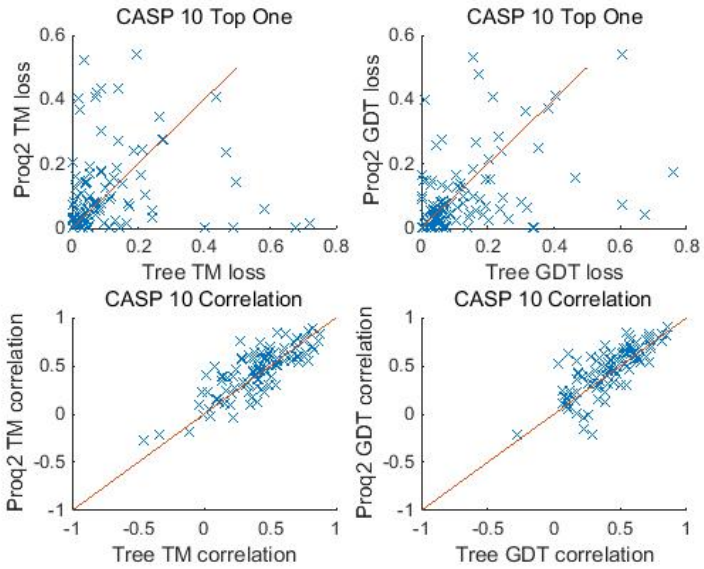

Figure 4.4.5 Compare the Result of Decision Tree with Proq2 on Dataset One

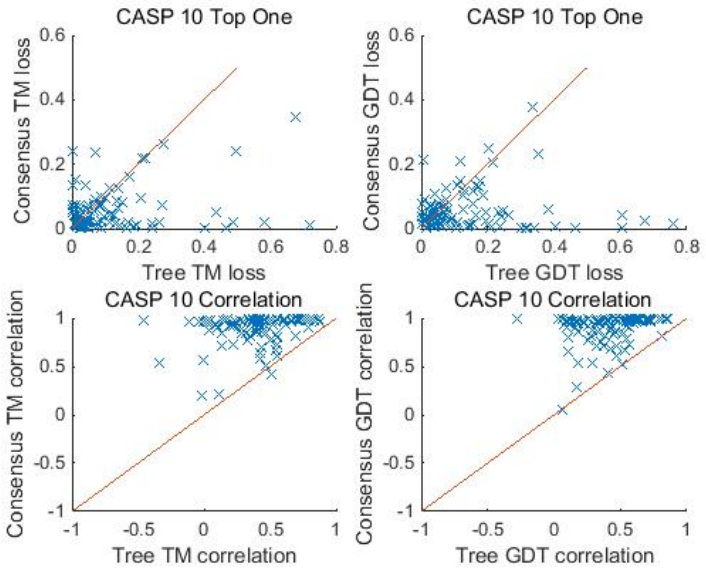

Figure 4.4.6 Compare the Result of Decision Tree with Consensus-based Method on Dataset One

When tested on dataset one, the performance of decision tree is comparable with Proq2, for some of the targets Proq2 got a very bad result, decision tree got much better result. But on both quality lost and Pearson's correlation, decision tree is obviously worse than consensus-based method, and only a few targets got a lower quality lost.

4.4.3 Neural Network

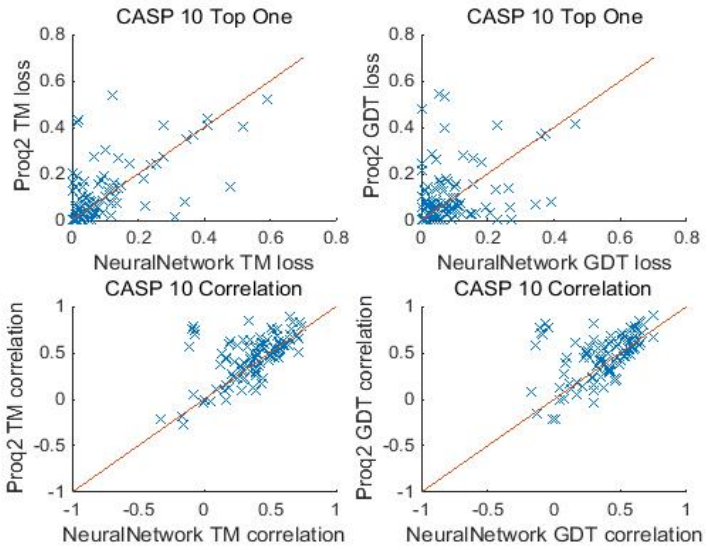

Figure 4.4.7 Compare the Result of Neural Network with Proq2 on Dataset One

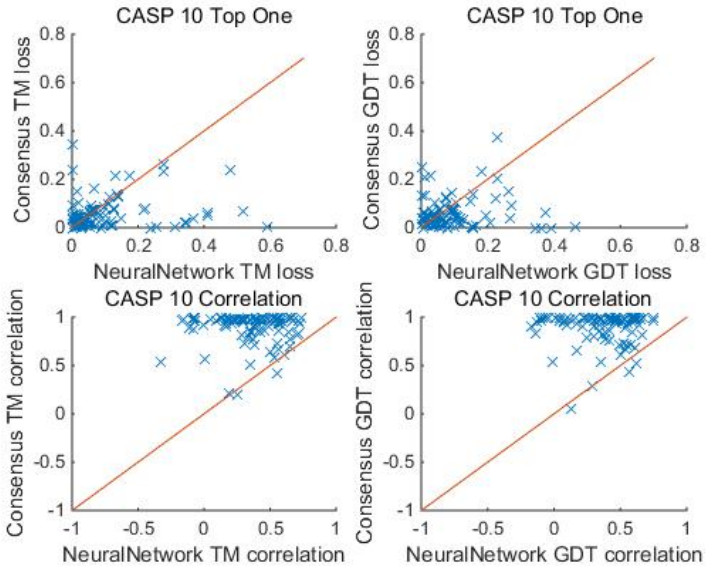

Figure 4.4.8 Compare the Result of Neural Network with Consensus-based Method on Dataset One

Neural network got a similar performance with decision tree on dataset one. The quality lost is slightly better than Proq2 but the correlation is even worse than

decision tree. On both quality lost and Pearson's correlation, decision is obviously worse than consensus-based method, and only a few targets got a lower quality lost.

4.4.4 Random Forest

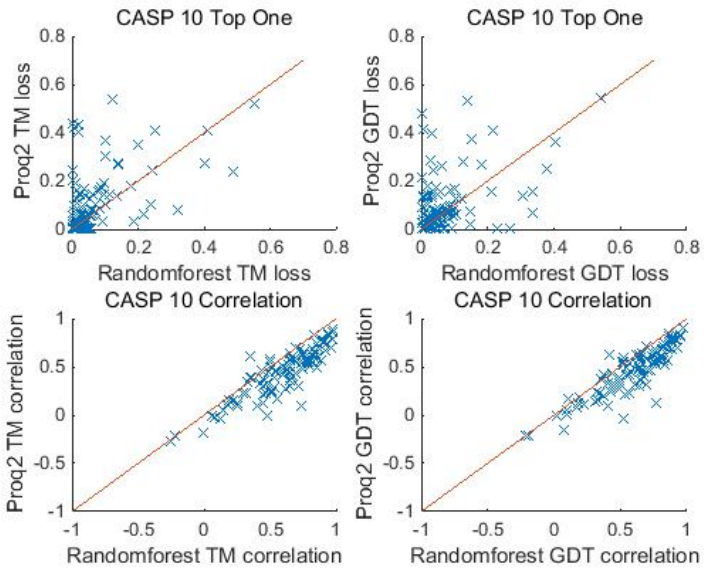

Figure 4.4.9 Compare the Result of Random Forest with Proq2 on Dataset One

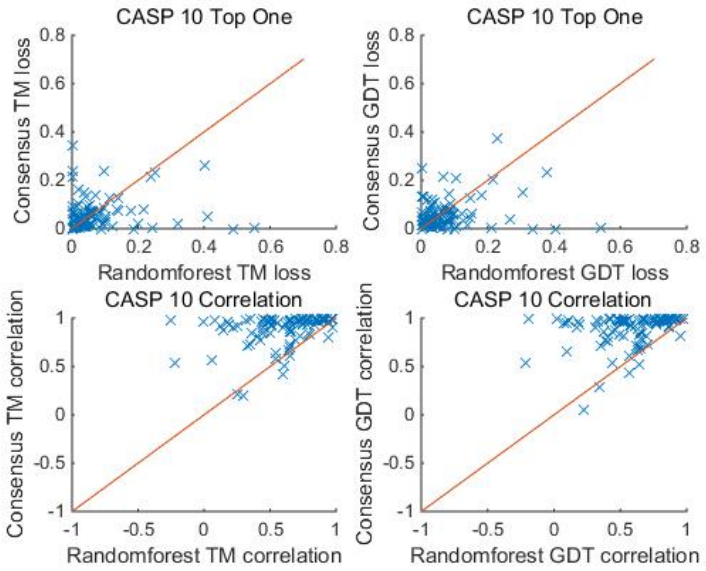

Figure 4.4.10 Compare the Result of Random Forest with Consensus-based Method on Dataset One

Random forest got a performance (0.0776, 0.0725) better than Proq2 (0.1063, 0.1140) but still worse than consensus-based method (0.0553, 0.0555) on quality lost when tested on dataset one, Among all of the 103 targets, 42 (43) got lower GDT-TS (TM-score) lost when using linear model.

The Pearson's correlation of Random forest (0.5977, 0.6046) is about 15 percent higher than Proq2 but still lower than consensus-based method. Almost all the targets are worse than consensus.

4.4.5 Boosting

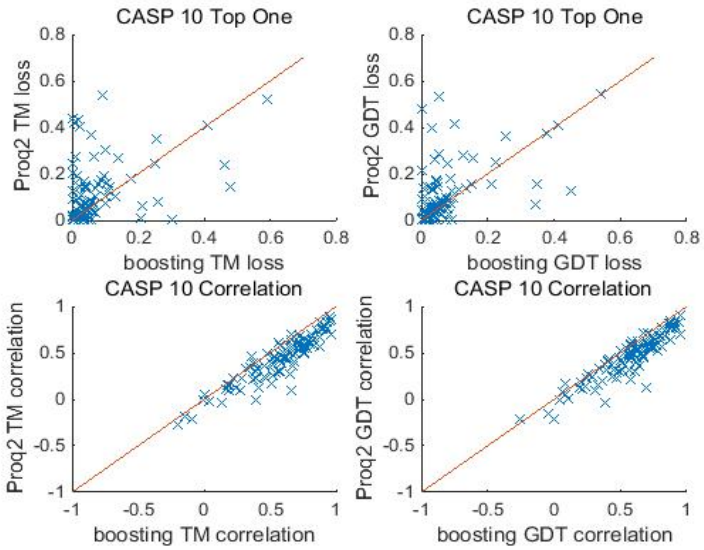

Figure 4.4.11 Compare the Result of Boosting with Proq2 on Dataset One

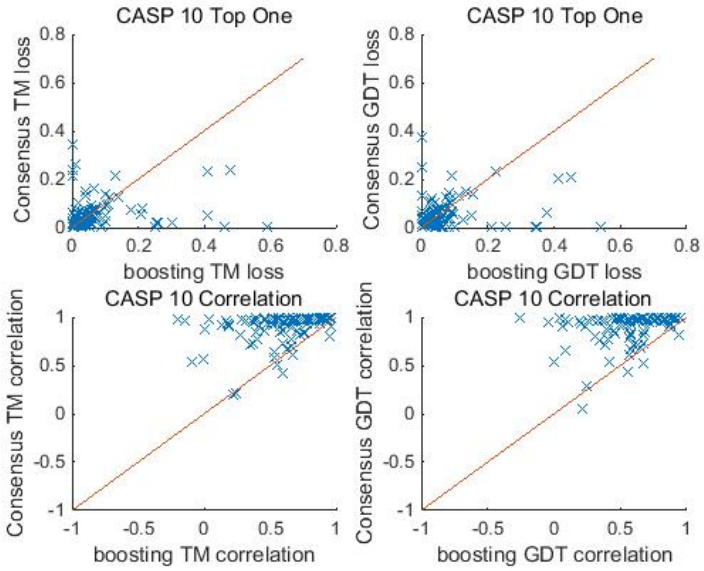

Figure 4.4.12 Compare the Result of Boosting with Consensus-based Method on Dataset One

When tested on dataset one, Boost got a performance (0.0728, 0.0764) better than Proq2 (0.1063, 0.1140) but worse than consensus-based method (0.0553, 0.0555) on

quality lost, Among all of the 103 targets, 50 (45) got lower GDT-TS (TM-score) lost using linear model.

The Pearson's correlation of Random forest (0.5746, 0.5831) is about 12 percent higher than Proq2, but still lower than consensus-based method. Almost all of the targets are worse than consensus.

4.5 CASP10 Stage One Model (set 20)

This part shows the result of five machine learning methods when using dataset two as test dataset. The overview result using GDT-TS and TM-score as label will be listed in table 4.5.1 and 4.5.2. Then the result will be compared with consensus-based method and Proq2.

Table 4.5.1 Performance of Different QA Methods for CASP10 Stage 1 Set20 Using GDT-TS Score as Label

	Gdtts Correlation	Gdtts lost	Find Best Models
Consensus	0.6742	0.0647	18
Linear	0.5848	0.0632	21
Decision Tree	0.3930	0.0855	9
Neural Network	0.4818	0.0608	21
Boosting	0.5860	0.0512	25
Random Forest	0.5858	0.0607	25
DDFIRE	0.3219	0.1107	15
DOPE	0.2513	0.1105	10
OPUS_CA	0.3853	0.1061	12
Proq2	0.4947	0.0846	17

RAPDF	0.2872	0.0996	13
RW	0.3503	0.1128	15

Table 4.5.2 Performance of Different QA Methods for CASP10 Stage 1 Set20 Using TM-score as Label

	TM Correlation	TM lost	Find Best Models
Consensus	0.6761	0.0655	17
Linear	0.5913	0.0637	25
Decision Tree	0.3707	0.0811	12
Neural Network	0.4913	0.0661	22
Boosting	0.5896	0.0599	22
Random Forest	0.5927	0.0549	23
DDFIRE	0.3128	0.1140	14
DOPE	0.2237	0.1136	11
OPUS_CA	0.3716	0.1074	14
Proq2	0.4846	0.0844	17
RAPDF	0.2832	0.0992	15
RW	0.3590	0.1121	16

From table 4.5.1 and 4.5.2 we can see when tested on dataset two, consensus-based method is still better than all the single model QA methods on correlation but not as obvious as dataset one.

If focus on the quality aspect, we can see that linear model, random forest and boosting got a lower average GDT-TS (TM-score) lost compared with consensus-based method. Among these methods boosting and random forest got the best result on GDT-TS lost (0.0512) and TM-score lost (0.0549).

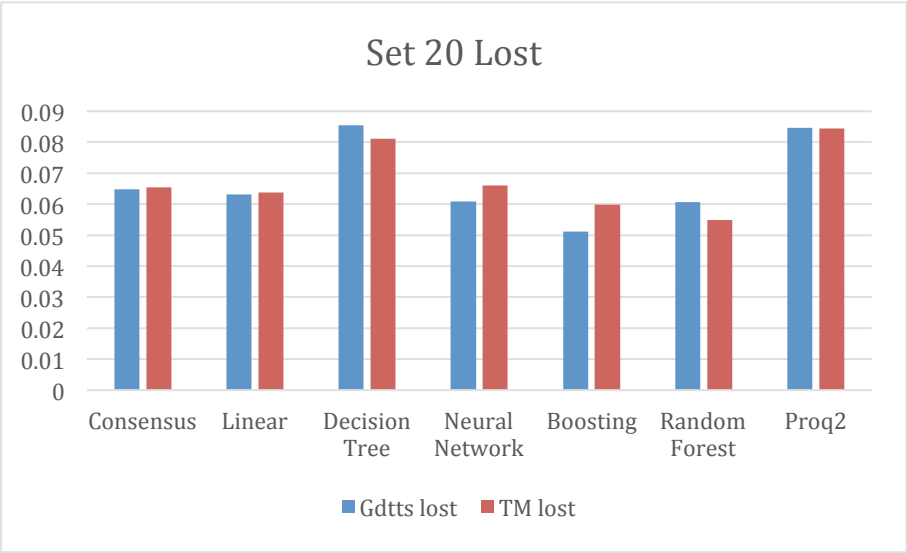

Figure 4.5.1 Quality Lost Performance of Different QA Methods for Dataset Two

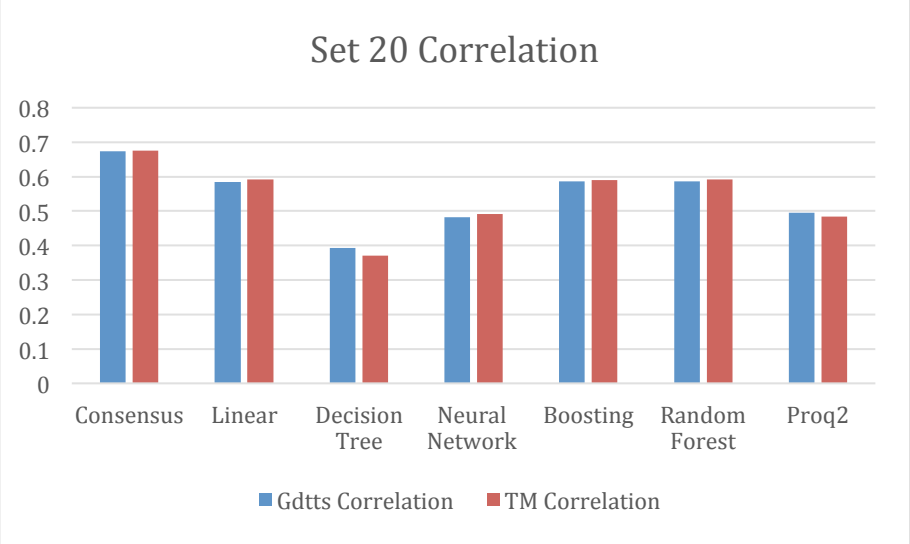

Figure 4.5.2 Pearson's Correlation Performance of Different QA Methods for Dataset Two

4.5.1 Linear Model

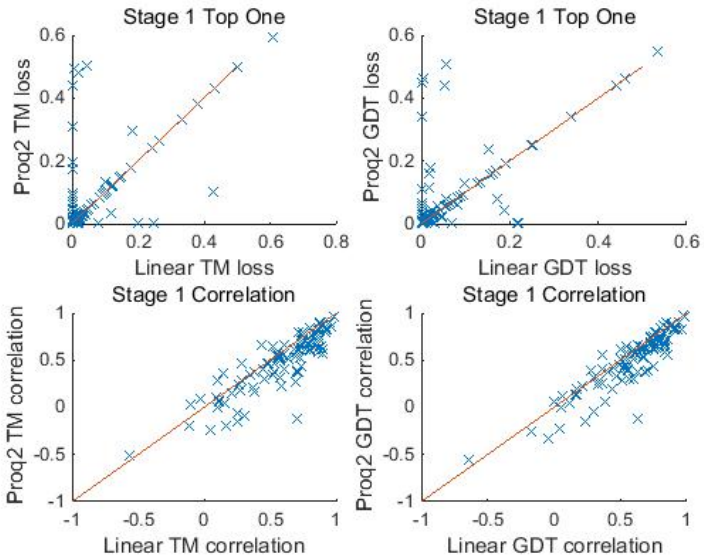

Figure 4.5.3 Compare the Result of Linear Model with Proq2 on Dataset Two

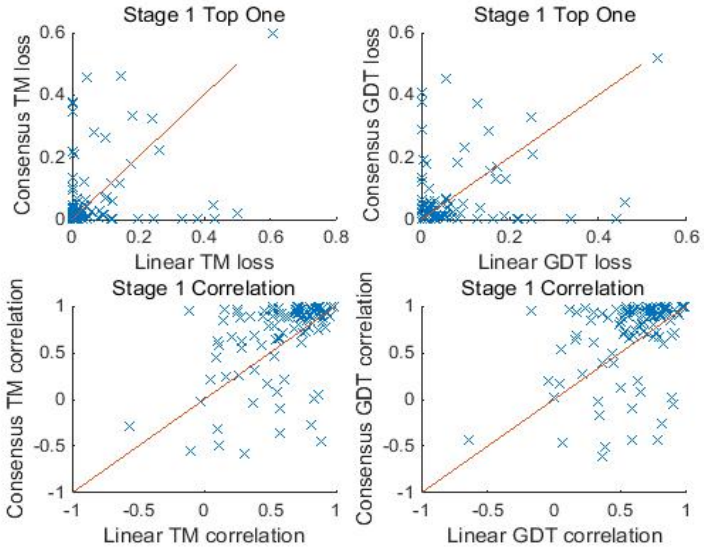

Figure 4.5.4 Compare the Result of Linear Model with Consensus-based Method on Dataset Two

When tested on dataset two, linear model got a lower quality lost (0.0632, 0.0637) than Proq2 (0.0846, 0.0844) and consensus-based method (0.0647, 0.0655). Among all of the 103 targets, 60 (57) got lower GDT-TS (TM-score) lost using linear model.

The Pearson's correlation of linear model (0.5848, 0.5913) is about 10 percent higher than Proq2, but still 10 percent lower than consensus-based method. Among all the 103 targets, only 24 (27) of the targets got a higher Pearson's correlation than consensus.

4.5.2 Decision Tree

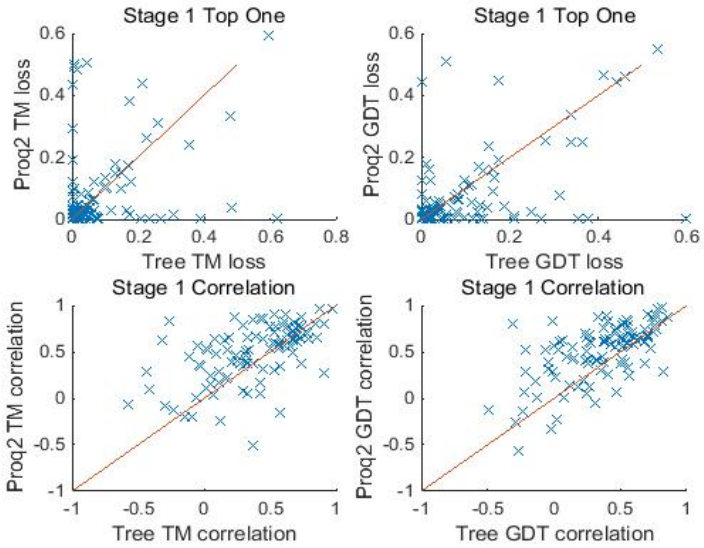

Figure 4.5.5 Compare the Result of Decision Tree with Proq2 on Dataset Two

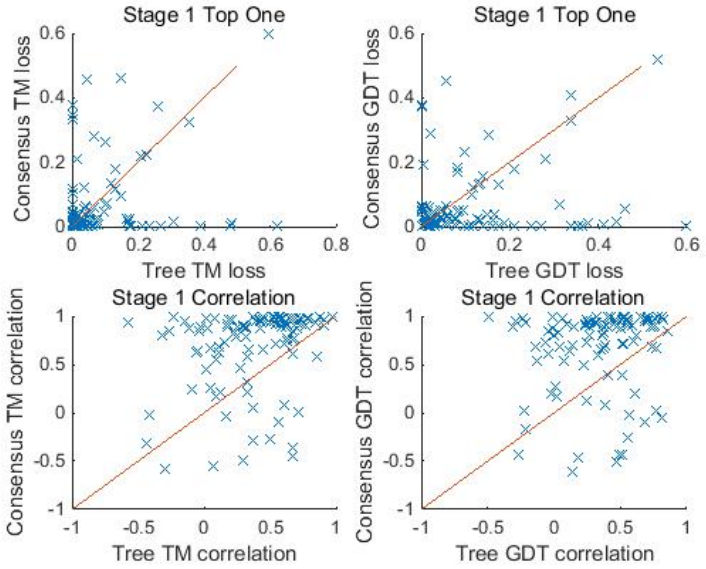

Figure 4.5.6 Compare the Result of Decision Tree with Consensus-based Method on Dataset Two

When tested on dataset two, although the performance of decision tree on Pearson's correlation is worse than Proq2, the quality lost result is comparable with Proq2, for some of the targets Proq2 got a very bad result, decision tree got much better result.

On both quality lost and Pearson's correlation, decision is worse than consensus-based method, 45 (47) among 103 targets got a lower quality lost, and 19 (17) got a higher Pearson's correlation.

4.5.3 Neural Network

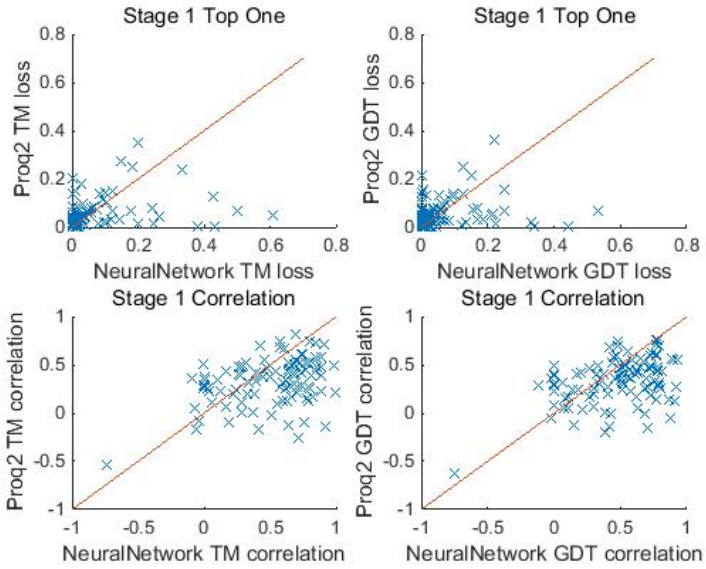

Figure 4.5.7 Compare the Result of Neural Network with Proq2 on Dataset Two

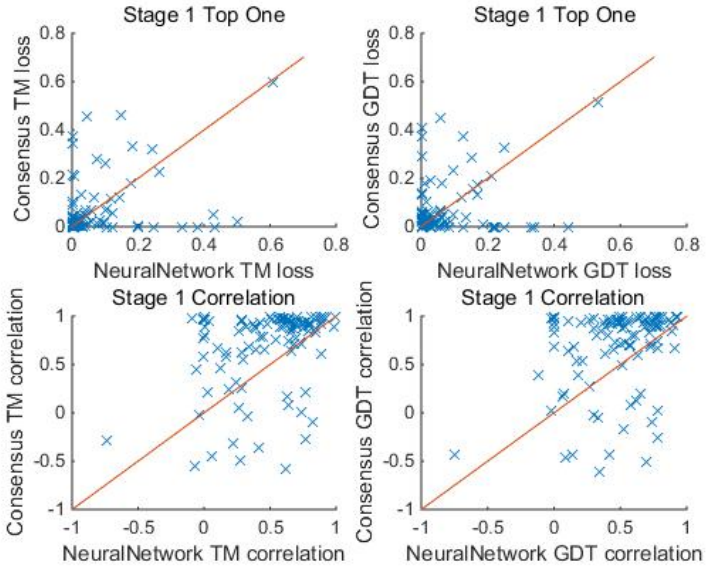

Figure 4.5.8 Compare the Result of Neural Network with Consensus-based Method on Dataset Two

When tested on dataset two, neural network got a good performance on quality lost, the GDT-TS lost of neural network (0.0608) is lower than both Proq2 (0.0846) and consensus-based method (0.0647). Also, the TM-score lost (0.0661) is also lower than Proq2 (0.0844) and comparable with consensus (0.0655). Among all the 103 targets, 61 (58) got lower GDT-TS (TM-score) lost using neural network.

The Pearson's correlation result of neural network (0.4818, 0.4913) is almost the same as Proq2 and it is about 20 percent lower than consensus-based method. Among all of the 103 targets, 22 (20) got higher Pearson's correlation when using neural network.

4.5.4 Random Forest

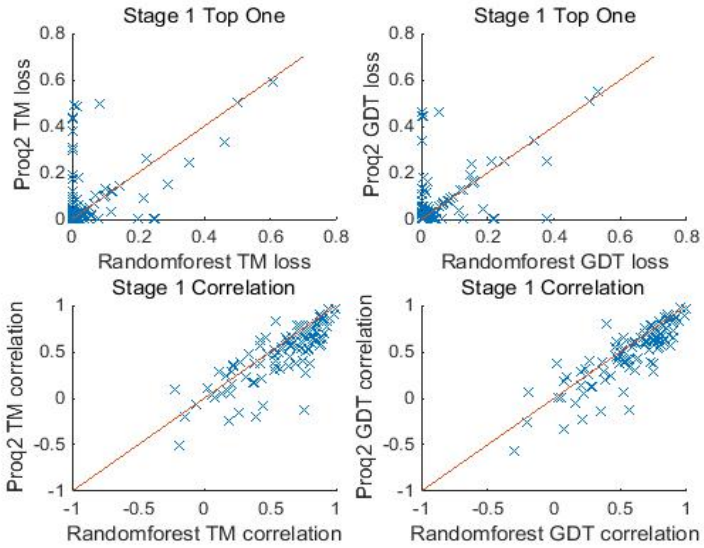

Figure 4.5.9 Compare the Result of Random Forest with Proq2 on Dataset Two

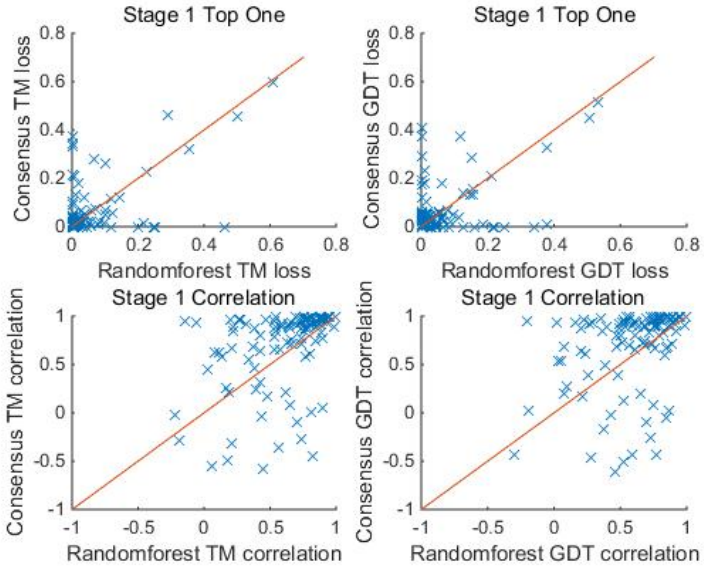

Figure 4.5.10 Compare the Result of Random Forest with Consensus-based Method on Dataset Two

When tested on dataset two, Random forest got a performance (0.0607, 0.0549) better than Proq2 (0.0846, 0.0844) and consensus-based method (0.0647, 0.0655) on quality lost. Among all of the 103 targets, 62 (59) got lower GDT-TS (TM-score) lost when using Random forest.

The Pearson's correlation of Random forest (0.5858, 0.5927) is about 10 percent higher than Proq2, but still lower than consensus-based method. Only 25 (24) among 103 targets got a higher Pearson's correlation than consensus-based method.

4.5.5 Boosting

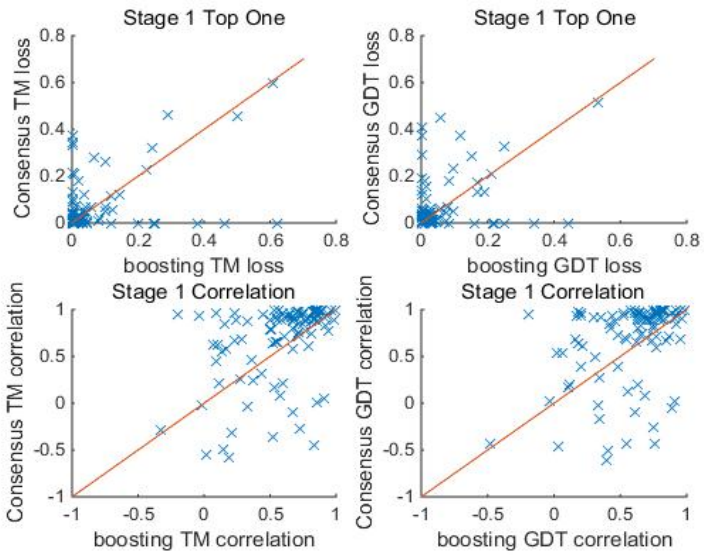

Figure 4.5.11 Compare the Result of Boosting with Proq2 on Dataset Two

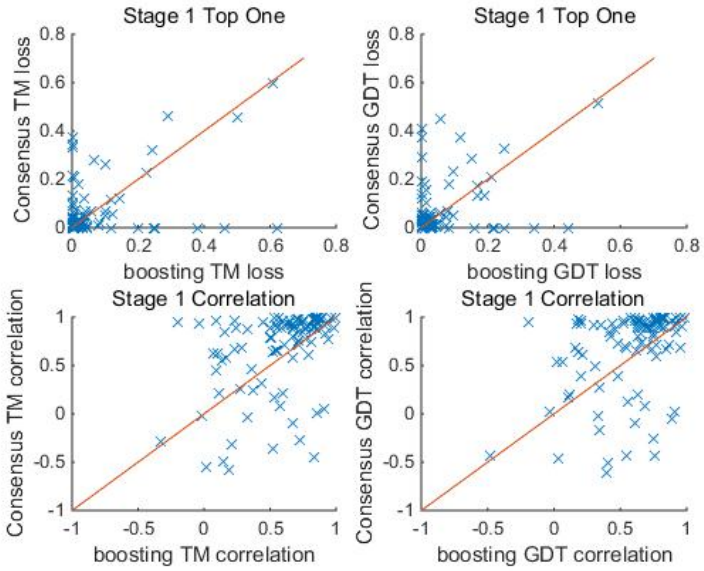

Figure 4.5.12 Compare the Result of Boosting with Consensus-based Method on Dataset Two

When tested on dataset two, Boosting got a performance (0.0512, 0.0599) better than Proq2 (0.0846, 0.0844) and consensus-based method (0.0647, 0.0655) on quality

lost. Among all of the 103 targets, 64 (58) got lower GDT-TS (TM-score) lost when using Boosting.

The Pearson's correlation of Boosting (0.5860, 0.5896) is about 10 percent higher than Proq2, but still lower than consensus-based method. And only 24 (24) among 103 targets got a higher Pearson's correlation than consensus-based method.

4.6 CASP10 Stage Two Model (set 150)

This part shows the result of five machine learning methods when using dataset three as test dataset. The overview result illustrates GDT-TS and TM-score as label, and they will be listed in table 4.5.1 and 4.5.2. Then the result will be compared with consensus-based method and Proq2.

Table 4.6.1 Performance of Different QA Methods for CASP10 Stage 2 Set150 Using GDT-TS Score as Label

	Gdtts Correlation	Gdtts lost	Find Best Models
Consensus	0.5003	0.0541	6
Linear	0.4083	0.0608	9
Decision Tree	0.2158	0.0640	1
Neural Network	0.2590	0.0581	4
Boosting	0.4039	0.0513	6
Random Forest	0.3849	0.0583	6
DDFIRE	0.3171	0.0695	3
DOPE	0.2708	0.0751	4
OPUS_CA	0.3642	0.0685	8
Proq2	0.3292	0.0566	12

RAPDF	0.3087	0.0617	4
RW	0.2585	0.0677	5

Table 4.6.2 Performance of Different QA Methods for CASP10 Stage 2 Set150 Using TM-score as Label

	TM Correlation	TM lost	Find Best Models
Consensus	0.4905	0.0537	4
Linear	0.4062	0.0568	6
Decision Tree	0.2492	0.0635	1
Neural Network	0.2060	0.0627	6
Boosting	0.3929	0.0506	7
Random Forest	0.3816	0.0519	6
DDFIRE	0.3142	0.0665	3
DOPE	0.2465	0.0749	3
OPUS_CA	0.3518	0.0666	4
Proq2	0.3200	0.0549	9
RAPDF	0.3240	0.0592	4
RW	0.2838	0.0622	5

From table 4.6.1 and 4.6.2 we can see that all the methods used structure knowledge got a decrease on correlation (consensus also used structure knowledge as it uses the similarity of structures between different models), and the gap between consensus-based method and machine learning methods is also not as large as dataset one.

When concentrating the quality aspect, we can see that random forest and boosting got a lower average GDT-TS (TM-score) lost than consensus-based method. Linear model and Proq2 also got very good result which is comparable with consensus-based method. Among these methods boosting got the best result on GDT-TS lost (0.0513) and TM-score lost (0.0506). Among all the methods, Proq2 got a best performance on direct best model picking up (12/103, 9/103).

Figure 4.6.1 Quality Lost Performance of Different QA Methods for Dataset Three

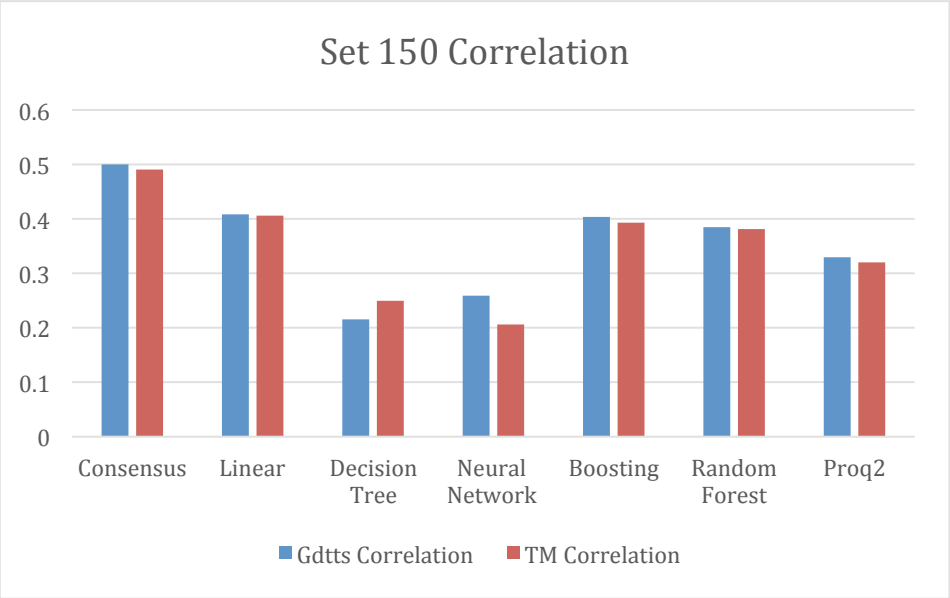

Figure 4.6.2 Pearson's Correlation Performance of Different QA Methods for Dataset Three

4.6.1 Linear Model

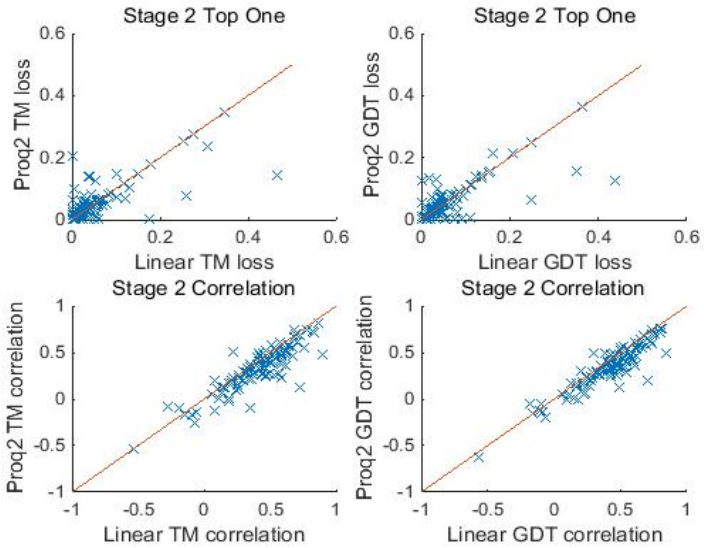

Figure 4.6.3 Compare the Result of Linear Model with Proq2 on Dataset Three

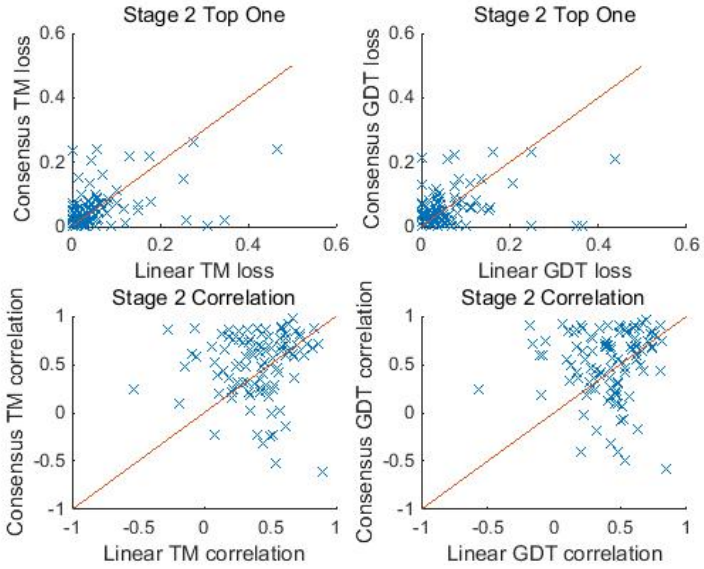

Figure 4.6.4 Compare the Result of Linear Model with Consensus-based Method on Dataset Three

When tested on dataset three, linear model got a quality lost (0.0608, 0.0568) comparable with Proq2 (0.0566, 0.0549) and consensus-based method (0.0541, 0.0537). Among all of the 103 targets, 51 (54) got lower GDT-TS (TM-score) lost using linear model.

The Pearson's correlation of linear model (0.4083, 0.4062) is about 8 percent higher than Proq2, but still 10 percent lower than consensus-based method. Among all of the 103 targets, 38 (38) of the targets got a higher Pearson's correlation than consensus.

4.6.2 Decision Tree

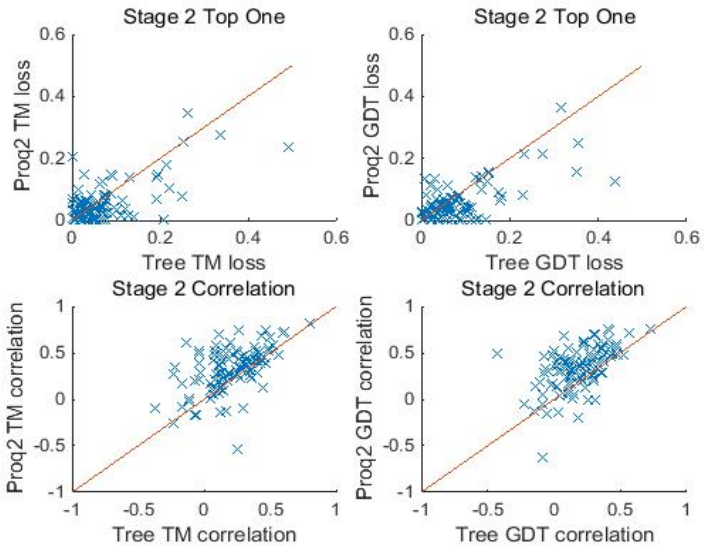

Figure 4.6.5 Compare the Result of Decision Tree with Proq2 on Dataset Three

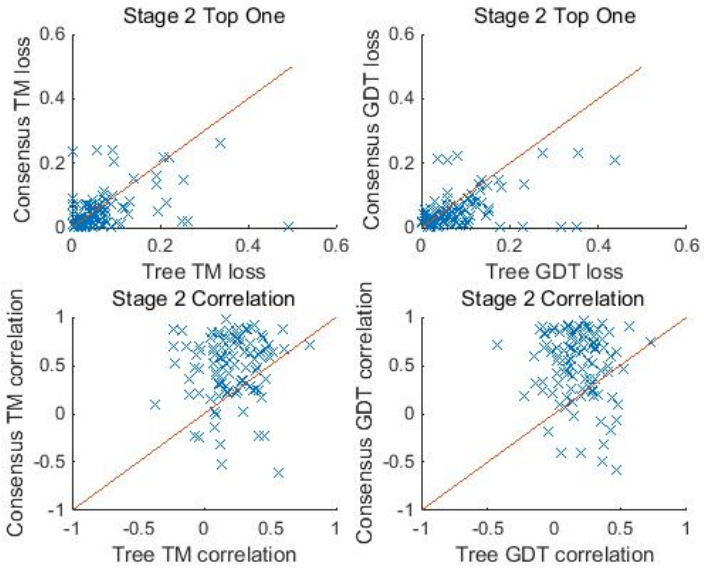

Figure 4.6.6 Compare the Result of Decision Tree with Consensus-based Method on Dataset Three

When tested on dataset three, decision tree got a quality lost (0.0640, 0.0635) which is worse than Proq2 (0.0566, 0.0549) and consensus-based method (0.0541,

0.0537). Among all of the 103 targets, only 33 (43) got lower GDT-TS (TM-score) lost when using decision tree.

The Pearson's correlation of decision tree is still much worse than Proq2 and consensus-based method. Among all of the 103 targets, only 20 (20) of the targets got a higher Pearson's correlation than consensus.

4.6.3 Neural Network

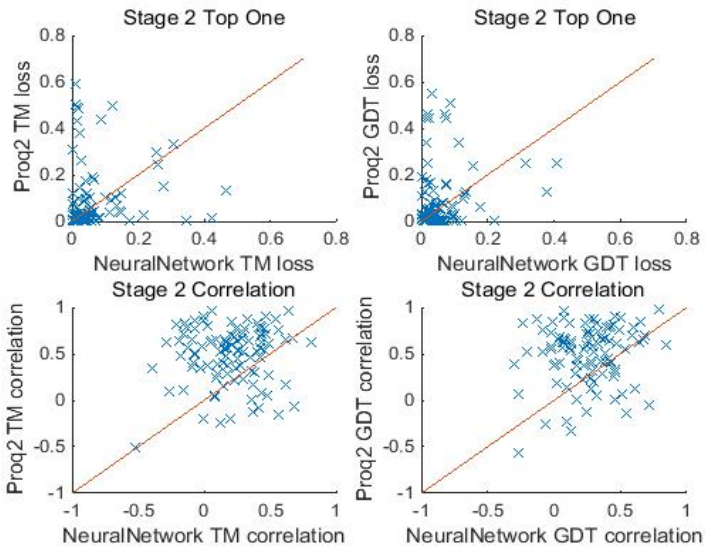

Figure 4.6.7 Compare the Result of Neural Network with Proq2 on Dataset Three

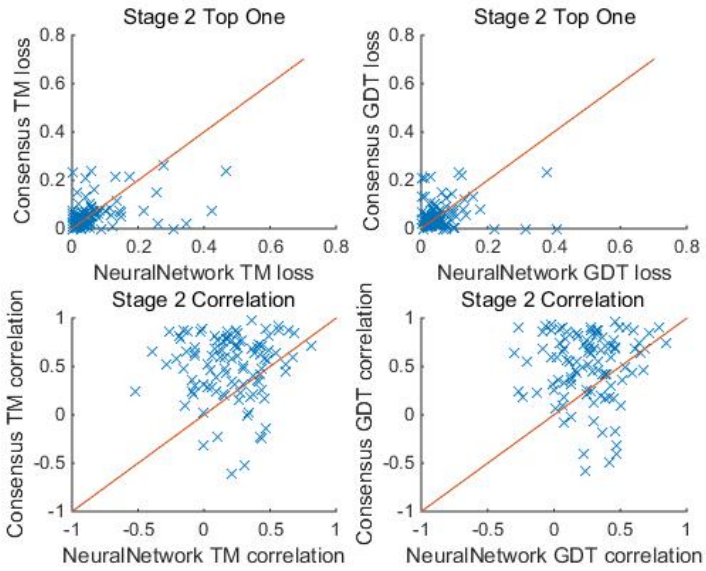

Figure 4.6.8 Compare the Result of Neural Network with Consensus-based Method on Dataset Three

When tested on dataset three, neural network got a quality lost (0.0581, 0.0627) which is slightly worse than Proq2 (0.0566, 0.0549) and consensus-based method (0.0541, 0.0537). Among all of the 103 targets, only 47 (50) got lower GDT-TS (TM-score) lost when using neural network.

The Pearson's correlation of neural network is much worse than Proq2 and consensus-based method. Among all of the 103 targets, only 26 (22) of the targets got a higher Pearson's correlation than consensus.

4.6.4 Random Forest

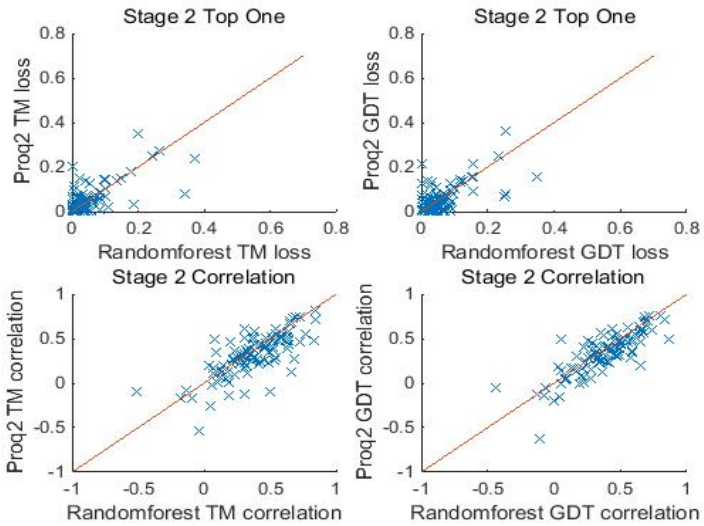

Figure 4.6.9 Compare the Result of Random Forest with Proq2 on Dataset Three

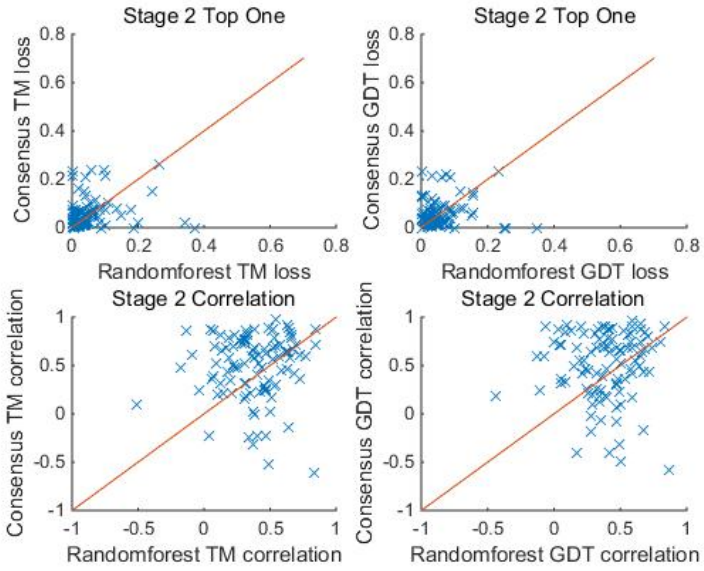

Figure 4.6.10 Compare the Result of Random Forest with Consensus-based Method on Dataset Three

When tested on dataset three, random forest got a quality lost (0.0583, 0.0519) which is comparable with Proq2 (0.0566, 0.0549) and consensus-based method (0.0541, 0.0537). Among all of the 103 targets, only 51 (51) got lower GDT-TS (TM-score) lost when using random forest.

The Pearson's correlation of random forest is 6 percent higher than Proq2 but still worse than consensus-based method. Among all of the 103 targets, only 38 (33) of the targets got a higher Pearson's correlation than consensus.

4.6.5 Boosting

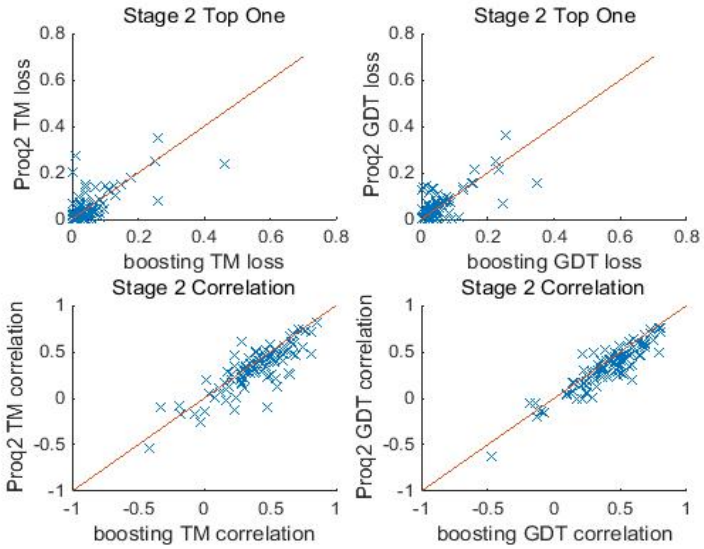

Figure 4.6.11 Compare the Result of Boosting with Proq2 on Dataset Three

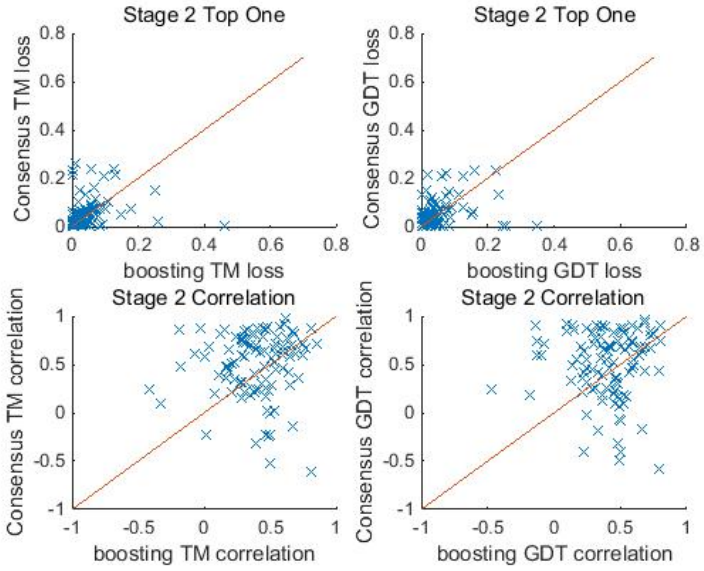

Figure 4.6.12 Compare the Result of Boosting with Consensus-based Method on Dataset Three

When tested on dataset three, boosting got a quality lost (0.0513, 0.0506) which is lower than Proq2 (0.0566, 0.0549) and consensus-based method (0.0541, 0.0537). Among all of the 103 targets, only 59 (59) got lower GDT-TS (TM-score) lost using boosting.

The Pearson's correlation of boosting is 8 percent higher than Proq2 but still 10 percent worse than consensus-based method. Among all of the 103 targets, only 37 (38) of the targets got a higher Pearson's correlation than consensus.

CHAPTER 5: CONCLUSIONS AND FUTURE WORK

The current project implemented five machine learning methods including linear model, decision tree，neural network，random forest and boosting and tested them on three CASP10 QA datasets. The performance of these methods was compared with Proq2 and consensus-based method on Pearson's correlation and quality lost.

Linear model is an efficient and stable method with the smallest cost. It showed a comparable performance with random forest and boosting on dataset one, and comparable performance with consensus-based method on dataset two and dataset three. For all of the three tested datasets, linear model showed a strong ability to select the best model directly form the model pool.

Neural Network and decision are not stable comparing with linear model. Single decision tree shows a quite unstable performance when tested on three datasets. When using decision tree, it is better to choose a larger feature number when branch. The unstableness of Neural Network is caused by its complexity and there are too many combinations of parameters. Some times neural network and decision tree will gives really good results, yet due to the difficulty on performance controlling, it still needs other process to check the result.

Random forest and Boosting are two advanced decision tree algorithms. These two methods showed best performance among all three datasets. And in dataset two and three their result is slightly better than consensus-based result. Comparing with

decision tree, random forest uses large number of trees to overcome the overfitting problem and when the tree number became larger, the performance became much more stable. Boosting also make the performance of algorithm stable and better by fix the initial model with the shrunken version of the following trees. These two methods also showed strong ability on selecting the best models directly.

Among these five methods, linear model and decision got the smallest system cost, and it is very fast. Neural Network is more complex but the procedure is not time consuming. Comparing with them, boosting and random forest will generate a very large model which can arrived a few gigabytes, and consume much more time for generating large number of decision trees.

In this study, even though different machine learning methods showed really good performance, more feature work is still needed. The first problem is some of the applications used to generate features are not the newest version, by update these applications, the quality of features may be improved. The second issue is the tree number of random forest and boosting just reached 3,000 and 1,000 in this work, but in some other researches [36], this parameter is tested up to 10000. Especially when testing boosting, it still showed a tendency of decreasing on quality lost when the tree number grew. Another important problem is that though random forest and boosting result in a lower correlation than consensus-based method, they still got a better performance on quality lost, this may hints machine learning methods is more suitable on some kinds of special targets or models. And on some targets that consensus-based

method got a bad result, machine learning methods works well, so if we can do a classification before quality assessment rather than using one method on all targets, the final performance may be improved again.

REFERENCE

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

[2] Lander ES, Linton LM, Birren B, Nusbaum C, Zody MC, et al. (2001) Initial sequencing and analysis of the human genome. Nature 409: 860-921.

[3] M.S. Johnson, N. Srinivasan, R.Sowdhamini, TL.Blundell, "Knowledge-based protein modeling," Crit Rev Biochem Mol Biol, vol. 29, pp. 1-68, 1994.

[4] Eisenhaber, F., B. Persson, and P. Argos, Protein structure prediction: recognition of primary, secondary, and tertiary structural features from amino acid sequence. Crit Rev Biochem Mol Biol, 1995. 30(1): p. 1-94.

[5] Kihara D, Chen H, Yang YD (2009) Quality assessment of protein structure models. Curr Protein Pept Sci 10: 216-228.

[6] Kryshtafovych A, Venclovas C, Fidelis K, Moult J (2005) Progress over the first decade of CASP experiments. Proteins 61 Suppl 7: 225-236.

[7] Moult J (2005) A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction. Curr Opin Struct Biol 15: 285-289.

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

[9] Simons KT, Bonneau R, Ruczinski I, Baker D (1999) Ab initio protein structure prediction of CASP III targets using ROSETTA. Proteins Suppl 3: 171-176.

[10] Cheng J, Wang Z, Tegge AN, Eickholt J (2009) Prediction of global and local quality of CASP8 models by MULTICOM series. Proteins 77 Suppl 9: 181- 184.

[11] Larsson P, Skwark MJ, Wallner B, Elofsson A (2009) Assessment of global and local model quality in CASP8 using Pcons and ProQ. Proteins 77 Suppl 9: 167-172.

[12] Case DA, Cheatham TE, Darden T, Gohlke H, Luo R, et al. (2005) The Amber biomolecular simulation programs. Journal of Computational Chemistry 26: 1668–1688.

[13] Brooks BR, Brooks CL, Mackerell AD, Nilsson L, Petrella RJ, et al. (2009) CHARMM: The Biomolecular Simulation Program. Journal of Computational Chemistry 30: 1545–1614.

[14] Zhou HY, Zhou YQ (2002) Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction. Protein Science 11: 2714–2726.

[15] Shen MY, Sali A (2006) Statistical potential for assessment and prediction of protein structures. Protein Science 15: 2507–2524.

[16] Samudrala R, Moult J (1998) An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction. Journal of Molecular Biology 275: 895–916.

[17] A. Ray, E. Lindahl and B. Wallner, "Improved model quality assessment using ProQ2," BMC Bioinformatics, vol. 13, p. 224, 2012.

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

[19] Manavalan, B., J. Lee, and J. Lee, Random Forest-Based Protein Model Quality Assessment (RFMQA) Using Structural Features and Potential Energy Terms. PloS one, 2014. 9(9): p. e106542.

[20] Wang Z, Tegge AN, Cheng J (2009) Evaluating the absolute quality of a single protein model using structural features and support vector machines.
Proteins 75: 638-647.

[21] Hyndman, Rob J. Koehler, Anne B.; Koehler (2006). "Another look at measures of forecast accuracy". International Journal of Forecasting 22 (4): 679–688. doi:10.1016/j.ijforecast.2006.03.001.

[22] Y.Zhang and J.Skolnick, "Scoring function for automated assessment of protein structure template quality," Proteins, vol. 57, no. 4, pp. 702-710, 2004.

[23] Y.Zhang and J.Skolnick, "TM-align: a protein structure alignment algorithm based on the TM-score," Nucleic Acids Res, vol. 33, no. 7, pp. 2302-2309, 2005.

[24] A. Zemla, "LGA: a method for finding 3D similarities in protein structures," Nucleic acids research, vol. 31, pp. 3370-3374, 2003.

[25] Q. Wang, Y.Shang, and D. Xu, "Improving a Consensus Approach for Protein Structure Selection by Removing Redundancy," IEEE/ACM transactions on computational biology and bioinformatics, vol. 8, no. 6, pp. 1708-1715, 2011.

[26] Y. Wu,M. Lu,M. Chen,J. Li and J. Ma, "OPUS-Ca: A knowledge-based potential function requiring only Ca positions," Protein Science, vol. 16, no. 7, pp. 1449-1463, 2007.

[27] J.Zhang,Y .Zhang, "A Novel Side-Chain Orientation Dependent Potential Derived from Random-Walk Reference State for Protein Fold Selection and Structure Prediction," PLoS ONE, vol. 5, no. 10, pp. 1-13, 2010.

[28] Kabsch W, Sander C (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 22: 2577-2637.

[29] McGuffin LJ, Bryson K, Jones DT (2000) The PSIPRED protein structure prediction server. Bioinformatics 16: 404-405.

[30] Freedman D A. Statistical models: theory and practice[M]. cambridge university press, 2009.

[31] Yan X. Linear regression analysis: theory and computing[M]. World Scientific, 2009.

[32] Linear regression, in Wikipedia http://en.wikipedia.org/wiki/Linear_regression.

[33] Decision Tree – Regression http://www.saedsayad.com/decision_tree_reg.htm.

[34] Quinlan, J. R. 1986. Induction of Decision Trees. Mach. Learn. 1, 1 (Mar. 1986), 81-106

[34] Breiman L (2001) Random forests. Machine learning 45: 5-32.

[35] Random Forest, in Wikipedia http://en.wikipedia.org/wiki/Random_forest.

[36] Balachandran Manavalan, Juyong Lee, Jooyoung Lee (2014) Random Forest-Based Protein Model Quality Assessment (RFMQA) Using Structural Features and Potential Energy Terms. DOI: 10.1371/journal.pone.0106542.

[37] Sikic M, Tomic S, Vlahovicek K (2009) Prediction of protein-protein interaction sites in sequences and 3D structures by random forests. PLoS Comput Biol 5: e1000278.

[38] Wang L, Yang MQ, Yang JY (2009) Prediction of DNA-binding residues from protein sequence information using random forests. BMC Genomics 10 Suppl 1: S1.

[39] Leo Breiman (1996). "BIAS, VARIANCE, AND ARCING CLASSIFIERS" (PDF). TECHNICAL REPORT. Retrieved 19 January 2015.

[40] Zhou Zhi-Hua (2012). Ensemble Methods: Foundations and Algorithms. Chapman and Hall/CRC. p. 23. ISBN 978-1439830031.

[41] Boosting, in Wikipedia http://en.wikipedia.org/wiki/Boosting.

[42] Artificial neural network, in Wikipedia http://en.wikipedia.org/wiki/Artificial_neural_network.

[43] A.A. Zamyatin, Protein Volume in Solution, Prog. Biophys. Mol. Biol. 24(1972)107-123.

[44] C. Chotia, The Nature of the Accessible and Buried Surfaces in Proteins, J. Mol. Biol., 105(1975)1-14.

[45] C. Tandford, Adv. Protein Chem. 17(1962)69-165.

[46] Cheng, J., et al., SCRATCH: a protein structure and structural feature prediction server. Nucleic Acids Res, 2005. 33(Web Server issue): p. W72-6.

[47] Magnan, C.N. and P. Baldi, SSpro/ACCpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity. Bioinformatics, 2014. 30(18): p. 2592-7.