QUALITY ASSESSMENT OF PROTEINS' COMPUTATIONAL STRUCTURES USING MACHINE LEARNING TECHNIQUES: RANDOM FOREST AND SUPPORT VECTOR MACHINE

Shokoufeh Mirzaei, College of Engineering, California State Polytechnic University, Pomona, California, 3801 W Temple Ave, Pomona, CA 91768, 909-869-2411, smirzaei@cpp.edu
Chester Carlson, California State Polytechnic University, Pomona, California, 3801 W Temple Ave, Pomona, CA 91768, 909-869-2411, cgc1204@gmail.com
Silvia Crivelli, Computational Science Division, Lawrence Berkeley National Laboratory, 1 Cyclotron Rd, Berkeley, CA 94720, Berkeley, CA, 925- 367-5900, sncrivelli@lbl.gov

ABSTRACT

In the field of Computational Structural Biology, a scoring function usually ranks the predicted protein models based on their similarity to the native structure. This paper presents scoring functions developed by two commonly used machine-learning techniques --Random Forest and Support Vector Machine-- to predict the quality of proteins structures. Both functions are compared against scoring functions proposed in the literature using four different metrics. Our results suggest that the addition of an important feature can yield results that are consistent across performance metrics whereas the performance of different methods using the same features is susceptible to the metric used.

INTRODUCTION

The human body is an extremely complex and efficient engine performing millions of microscopic operations each day. The information (the blue print) that drives these operations is encoded within our DNA. This DNA is broken down into sequences within 23 chromosome pairs, each of which contain genes that give very specific instruction for the proteins in our body. Proteins are building blocks of our tissues, muscles and bones. They increase our immunity, balance our PH value, transport nutrients, and form enzymes that help to break down foods. Proteins are created as an extended sequence of amino acids and they quickly fold into a complex structure called 3D structure or native structure. This structure ultimately dictates the function of a protein.

To date, the sequences of millions of proteins have been determined by numerous genome projects [1]. However finding the 3D structure of these sequences poses great difficulty to scientists. There are about 100,000 protein structures that have accurately been determined by means of experimental procedures such as X-Ray Crystallography and NMR Spectroscopy. The large deficit in the number of known 3D structures is due to the extreme time and cost requirements associated with the experimental methods. Therefore, it is vital to the field to find a more effective method of determining protein structures. Computational methods have been under development and progress for several years and, as more data becomes available and computers get faster, these methods have aroused hope of a more efficient way to protein structure determination. If proven to be successful, computational methods will complement or even replace current experimental ones.

The U.S. National Institute of General Medical Sciences (NIH/NIGMS) sponsors a bi-annual competition named Critical Assessment of protein Structure Prediction (CASP) whose goal is to advance the development of methods for determining a protein's structure from its sequence [2] (http://predictioncenter.org). Every other summer, the CASP competition challenges hundreds of different participants to use their methods of protein structure prediction to blindly predict the native structure of about one hundred proteins whose native structures have been or are about to be

experimentally determined but not yet published, given only their sequence of amino acids. After the competition is over and the native structures are made available to the public, CASP evaluates the quality of each contestant's work and assesses the evolution of the methods along CASP competitions.

One of the main roadblocks in protein structure prediction is that the prediction methods generally produce many models (decoys) but they are not efficient at selecting the best ones. For the past two decades, a repository of CASP data (i.e., protein models submitted to CASP by the different participating groups) has been generated which has created an opportunity for application of machine learning (ML) methods to improve protein selection and consequently protein structure prediction. Often, ML techniques are used to combine several protein features to identify the "native-like" models. Machine learning algorithms use a wide variety of approaches including Support Vector Machine (SVM) [1, 3, 4], and Random Forest (RF)[5], to combine measurable features into a single number that estimates the quality of decoys [6].

Mirzaei et. al. (2015) used SVM to develop scoring functions using data from CASP8 (2008), CASP9 (2010), and CASP10 (2012). Mirzaei et al. (2015) use a publicly available benchmark dataset of protein models and their associated features (i.e. protein characteristics) developed by Chen Keasar, which makes it possible to compare a variety of machine learning methods, or test hypothesis regarding the inclusion/exclusion of features in the dataset. In this manuscript, we use Keasar's benchmark dataset to compare scoring functions based on the Random Forest Regression (RF-Regression) and Support Vector Regression (SVM-e) algorithm using 56 and 59 of the features in the dataset.

This paper is structured as follows. First, we explain the dataset and protein features used. Next, we discuss the RF-Regression method and its parameter tuning schema. Then, we explain the SVM-e method and its parameter tuning technique. Finally, we provide the results and conclusions, respectively.

DATASET AND FEATURES

Just like we did in [1], in this paper we use Keasar's benchmark dataset (http://wefold.nersc.gov/wordpress/download). This dataset consists of 36,682 computationally predicted decoys corresponding to 302 protein prediction targets from CASP8, CASP9, and CASP10. These three competitions have already been completed and the "native" structures of target proteins are publicly available. This dataset also includes 59 features calculated for every decoy structure using the MESHI software package designed for protein modeling. The features in the 56-feature dataset consist of a combination of commonly used and knowledge-based energy terms like hydrogen bonds, hydrogen bond patterns, and energy terms. The 59 feature-dataset has three extra features: energy, optimization Score and selection Score in addition to the original 56. Energy is indeed the MESHI energy which itself is a linear combination (with manually tuned coefficients) of many other features in the list (bond, angle, etc.). Optimization Score and selection Score features are different versions of Keasar's scoring functions, which use a non-linear combination of other features. The details of these features can be found at (http://www.cs.bgu.ac.il/~frankel/TechnicalReports/2015/1506.pdf).

We use the quality assessment (QA) criterion for decoy quality called Global Distance Test Total Score (GDT_TS). GDT_TS is a measurement that shows how similar a decoy is to its native structure. This measurement is calculated based on the largest set of amino acid residues' alpha carbon atoms in the model that fall within a set of defined distance cutoff of their position in the experimental structure. GDT_TS is the primary ranking metric used by the CASP participants and assessors to determine the quality of a decoy once the experimental structure of the protein has been made available.

RANDOM FOREST

Grouping Important Features

To develop the optimal scoring function based on the Random Forest algorithm, two tasks must be accomplished. First, group significant features together. Second, randomly sample features from those groups to create the scoring function. To reduce the chance of overfitting the data with too many features, instead of using all features, grouping and sampling of features at random was implemented. Overfitting happens when a predictive model has too many parameters and becomes overly relevant to the training dataset. An over-fit model tends to have a superb performance on the training set and poor performance in predicting a new observation. To group the features, we calculated the statistical correlation for each pair of features. These measurements resulted in two matrices (a 56x56 matrix and a 59x59 matrix) with values between -1 and 1.

We used hierarchical clustering to group highly correlated features. To construct the clustering Dendrogram, we used *1-abs(correlation(i,j))* instead of using the *correlation(i,j)* terms. This way, features that are highly correlated appear at the bottom of the Dendogram, making the grouping task more intuitive. Figures 1 and 2 show the Dendrograms of *1-abs(correlation(i,j))* terms for the 56-feature and 59-feature dataset. Using figure 1, features for the dataset were grouped in eight categories and one group was allocated to features that are not correlated to any other features, called fixed features. All the fixed features were used in the development of all scoring functions because they did not have a correlation with other features in the pool. On the other hand only one feature, selected randomly, was used from each of the highly correlated groups in the development of the scoring functions. Therefore, in each run of RF-Regression the set of fixed features in combination with eight features randomly selected from the eight clusters (one from each with replacement) created the final set of features used for creating scoring functions. The same schema was used for the datasets with 59 features. This time using figure 2, 12 groups were formed including 11 clusters of correlated features and 11 fixed features.

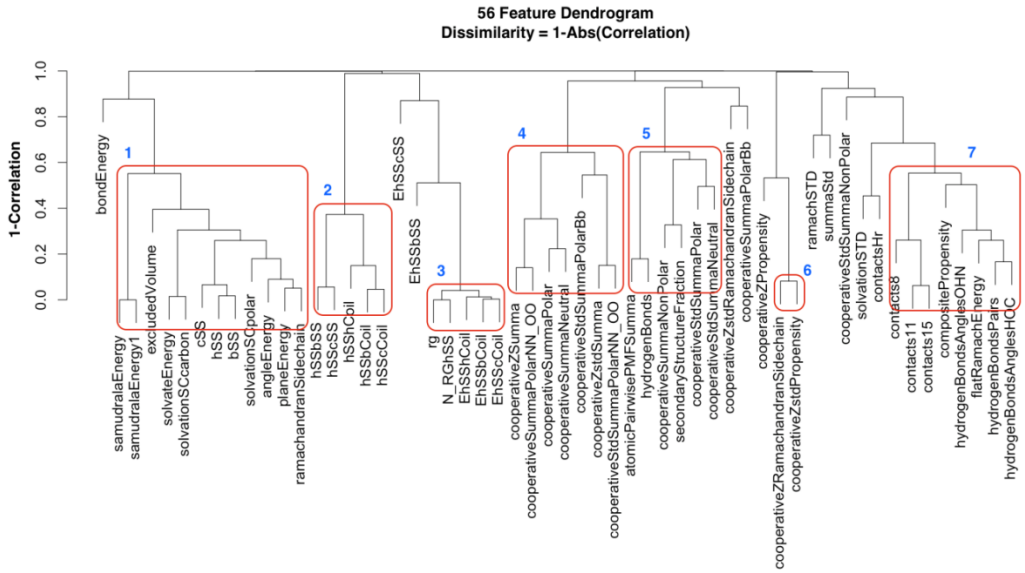

Figure 1: Dissimilarity Plot with Clusters for the 56-Feature Dataset

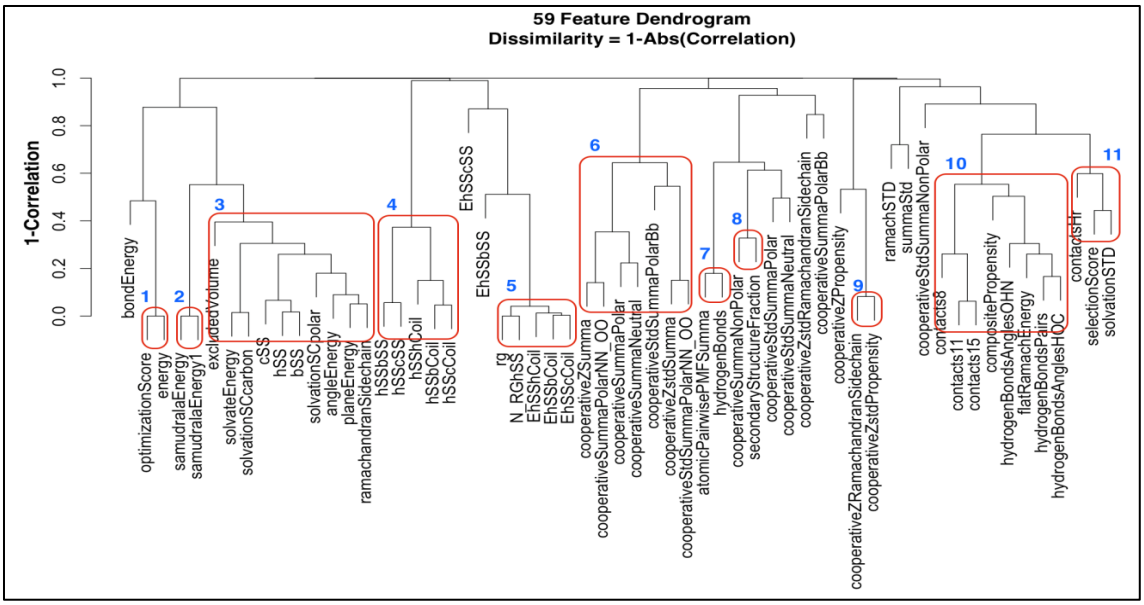

Figure 2: Dissimilarity Plot with Clusters for the 59-Feature Dataset

METHOD

Random Forest Scoring Functions

Random Forest, proposed by Brieman (2001) is an ensemble learning technique that employs thousands of decision trees to perform regression/classification to develop a strong predictive model. From the available benchmark database two datasets have been assembled to train and test the performance of a scoring function. One dataset contains 56 features and the other contains the same 56 features plus three predefined combinations: the MESHI energy – a manually tuned linear combination of those 56 features, and two different non-linear combinations from Keasar's scoring functions. Thus the second dataset tests the possible advantage of a double-layered computational model.

To be consistent with Mirzaei et. al. (2015), decoys from CASP10 are considered as test set and decoys from CASP8 and CASP9 are used for training purposes for both the 56- and 59-feature datasets. Each dataset is analyzed by the same method to determine whether the addition of the three MESHI features provides a significant improvement in the model performance. One hundred combinations of each set of features are randomly selected to develop scoring functions. The features that are randomly picked become the predictors in the RF-Regression function to predict the GDT_TS, which is a continuous response variable. Each of the scoring functions, developed by the training dataset, will be evaluated by predicting the value of GDT_TS in the testing dataset. The performance of each scoring function and the overall RF-Regression method is determined by three measurements that will be discussed in the Results section of this paper.

Parameter Tuning

There are many different parameters that can be adjusted and fine-tuned to improve the output of the RF-Regression algorithm. Because this research utilizes the regression model of the RF-Regression algorithm, two main parameters of the RF-Regression were selected for parameter tuning. These

parameters are "m_{try}", the number of predictor variables picked for splitting at each node and "n_{tree}", the number of trees grown for each run. Similar research suggests that these two parameters have the most influence on the learning of the RF-Regression algorithm [6]. A grid search was conducted to find the optimum values of m_{try} and n_{tree} over a range of values. In the grid search, m_{try} ranged between 1 and 17 for the 56-feature dataset and between 1 and 21 for the 59-feature dataset and n_{tree} ranged between 10 to 5000 in various intervals which were consistent for both datasets. Consequently, n_{tree}=1200 decision trees and m_{try} = 5 of the 56-feature dataset provided the optimal performance within the range investigated in this research. Additionally, for the 59-feature dataset, n_{tree}=400 decision trees and m_{try} = 7 features yielded the RF-Regression with the highest performance. To implement the method, we used the random Forest package in R ported by Andy Liaw and Matthew Wiener from a Fortran source code by Leo Breiman and Adele Cutler [7]. This source code can be downloaded at (http://www.stat.berkeley.edu/~breiman/RandomForests/reg_home.htm). We used three Quality Assessment metrics for the comparison of these methods' performances on the test set. These metrics are median of per target errors, median of per target loss, median of per target correlation, and enrichment, which are discussed in the next section.

Developing the SVM-e Scoring Function

From Keasar's database, two sets of data were generated to train, evaluate, and test the scoring function. One dataset includes 56 features and the other dataset includes those 56 features plus three predefined combinations thereof: the MESHI energy -- a manually tuned linear combination of those 56 features, and two different non-linear combinations from a previous version of Keasar's scoring function [1]. Thus the second dataset tests the possible advantage of a double-layered computational model.

We train and evaluate the scoring function and tune its parameters using 190 non-redundant protein targets from CASP8 and CASP9. The 190 targets include 84 targets from CASP8 and 106 targets from CASP9. The protein targets were randomly assigned to 19 folds, each fold containing 10 targets. The random assignment of proteins to fold was done without replacement, thus making sure each target contributes exactly once to the validation process. To train the model, we selected 18 folds –containing 180 targets—and used the remaining fold --containing 10 targets— to evaluate the scoring function. We repeated this process 19 times. Furthermore, a grid search was conducted to tune the SVM-e parameters γ, C, and ε. To this end, during each round of the cross-validation, a wide range of parameter values were used to train the model. The process was repeated multiple times and the final sets of parameter values to which the model was found to be sensitive were: $\gamma = \{0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10, 20, 30, 40\}$ and $c = \{0.1, 0.2, 0.3, 0.4, 0..5, 1, 2, 3, 4, 5, 6, 7\}$, with $\varepsilon = 0.1$. Also, it was found that a Radial Basis Kernel Function (RFB) provides the lowest error term. Consequently, each fold was trained with 120 different combinations of parameters, resulting in 2280 scoring functions. Figure 3 shows the average root mean square of the differences between the predicted and observed decoys GDT_TS (MRMSE) for each combination of gamma and cost parameters and for the training and validation sets.

We selected a gamma equal to 0.01 which minimized the MRMSE of the validation set. Although, this value of gamma does not minimize the training set, the errors of both, training and validation sets are reasonably low. Also, we selected C as the value that corresponds to the median of MRMSE in the validation set when gamma is 0.01. The final parameter values are γ=0.01, C=3, and ε=0.1. Applying the same procedure using both datasets with 56 and 59 features resulted in the selection of the same set of parameters.

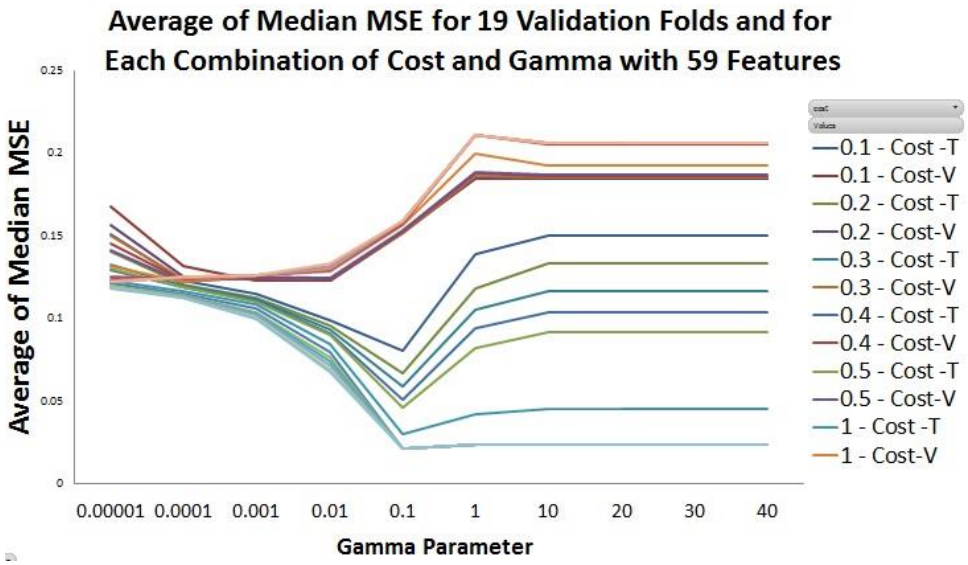

Figure 3. MRMSE for each combination of gamma and cost, and for the training and validation sets. Cost-T is the cost parameter for the test set and Cost-V is the cost parameters associated with the validation set.

We used these selected parameters and the 19 random folds used for the training sets to predict the CASP10 data as a blind test and to evaluate the model performance. This process yielded 19 MRMSE terms. Figure 3 shows the graph of MRMSE when the CASP10 dataset is tested. The 19 scoring functions have different accuracy, as they are trained with different sets of protein targets. Since only 10 targets were put away every time, the training sets have an overlap of almost 95%. The MRMSE of the scoring functions when the 59 features are included varies between 0.145 and 0.157. Figure 4 shows the histogram of MRMSE for the 19 scoring functions when they are used to predict the GDT_TS of CASP10 targets. As the histogram shows, the MRMSEs follow a t-distribution with 18 degrees of freedom. Therefore, it can be concluded that with the set of parameters introduced earlier, the MRMSE of CASP10 targets for any random selection of 180 targets from CASP8 and CASP9 will result in a MRMSE that falls between 0.149 and 0.151 with 95% level of confidence.

When 56 features were used for the training purposes, the MRMSE of the scoring functions ranges between 0.149 and 0.160. Also, the MRMSE of CASP10 for any random selection of 180 targets from CASP8 and CASP9 result in a MRMSE that falls between 0.152 and 0.155 with 95% level of confidence. This leads us to the conclusion that the inclusion of the three extra features has a significant impact on the accuracy of the predictive model using SVM-e.

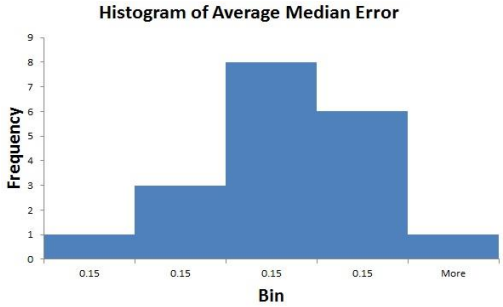

Figure 4. Histogram of median root squared MSE when 19 random folds were used to predict CASP10 data and parameters were: $\gamma=0.01$, $c=3$, and $\varepsilon=0.1$.

1. RESULTS

The performance of the proposed models was evaluated by four complementary measures. Table 1 shows the results of the 2 variants of the SVM-e method compared to the 2 variants of the RF-Regression method as well as the statistical potentials GOAP [8, 9] and QMEAN [10] using three evaluation criteria. Table 1 also shows that when using the same features the results are susceptible to the evaluation metric used. In fact, SVM-e with 59 features outperforms other methods in Median Error and Median Pearson Correlation; and it is outperformed by GOAP in Loss and Enrichment. These results show that our SVM-e method benefits from a double-layered approach that adds an energy function and two machine-learning based scoring functions to the set of 56 features. Likewise, of the RF-Regression variants, RF-Regression with 59 features outperforms the result obtained by 56 features in 3 metrics. Our results suggest that the addition of an important feature can yield results that are consistent across performance metrics whereas the performance of different methods using the same features is susceptible to the metric used.

Table 1. Comparison of the methods using different metrics. The best performance in each measure is bolded.

	SVM-e (59 features)	SVM-e (56 features)	RF-Regression (59 features)	RF-Regression (56 features)	GOAP	QMEAN
Median Error	**0.149**	0.153	0.154	0.168	N/A	0.18
Median Pearson Correlation	**0.599**	0.595	0.570	0.466	0.52	0.642
Loss	0.052	0.050	0.062	0.064	**0.036**	0.046
Enrichment	3.667	3.333	3.333	3.333	**4.46**	4.17

CONCLUSION

The main goal of this paper was to evaluate two approaches for scoring protein models and to compare them with each other and with two baseline methods, GOAP and QMEAN. To this end we use a carefully designed machine learning set-up that includes a carefully curated, non-redundant data set, and a common feature set. These methods, SVM-e and Random Forest, predict the GDT_TS score of a single decoy using a set of structural features as well as energy and Meta energy terms that can be directly calculated from that decoy. We considered two versions of the SVM-e and RF-Regression methods and we evaluated their performance using four different metrics. A secondary goal of this paper was to improve the baseline results. This goal was achieved by SVM-e with 59 features in median error. Although the improvements shown here and in the literature are minor, they underscore the difficulty of the protein scoring problem and suggest that our efforts to collaborate across disciplines are worthwhile pursuing.

ACKNOWLEDGEMENTS

This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Visiting Faculty Program (VFP) program. This work also used resources from National Energy Research Scientific Computing Center (NERSC) for providing computing resources that made this research possible.

REFERENCES

[1] Mirzaei S, Sidi T, Keasar C, Crivelli S. Purely structural protein scoring functions using support vector machine and ensemble learning. *Proceedings of BioKDD conference, Sydney, Australia, Aug*. 2015

[2] Moult J, Fidelis K, Kryshtafovych A, Schwede T, Tramontano A. Critical assessment of methods of protein structure prediction (casp)—round x. *Proteins: Structure, Function, and Bioinformatics*. 2014;82:1-6

[3] Qiu J, Sheffler W, Baker D, Noble WS. Ranking predicted protein structures with support vector regression. *Proteins: Structure, Function, and Bioinformatics*. 2008;71:1175-1182

[4] Wang Z, Tegge AN, Cheng J. Evaluating the absolute quality of a single protein model using structural features and support vector machines. *Proteins: Structure, Function, and Bioinformatics*. 2009;75:638-647

[5] Manavalan B, Lee J, Lee J. Random forest-based protein model quality assessment (rfmqa) using structural features and potential energy terms. *PloS one*. 2014;9:e106542

[6] Shi X, Zhang J, He Z, Shang Y, Xu D. A sampling-based method for ranking protein structural models by integrating multiple scores and features. *Current Protein and Peptide Science*. 2011;12:540-548

[7] Breiman L. Random forests. *Machine learning*. 2001;45:5-32

[8] Zhou H, Skolnick J. Goap: A generalized orientation-dependent, all-atom statistical potential for protein structure prediction. *Biophysical journal*. 2011;101:2043-2052

[9] Kalisman N, Levi A, Maximova T, Reshef D, Zafriri-Lynn S, Gleyzer Y, et al. Meshi: A new library of java classes for molecular modeling. *Bioinformatics*. 2005;21:3931-3932

[10] Benkert P, Tosatto SC, Schomburg D. Qmean: A comprehensive scoring function for model quality assessment. *Proteins: Structure, Function, and Bioinformatics*. 2008;71:261-277