Journal of Bioinformatics and Computational Biology
Vol. 13, No. 2 (2015) 1550005 (19 pages)

DOI: 10.1142/S0219720015500055

Imperial College Press
www.icpress.co.uk

Quality assessment of modeled protein structure using physicochemical properties

Prashant Singh Rana*, Harish Sharma, Mahua Bhattacharya
and Anupam Shukla

Department of Information Communication and Technology
ABV-Indian Institute of Information Technology and Management
Gwalior MP-474015, India
**psrana@gmail.com*

Received 4 April 2014
Revised 17 October 2014
Accepted 16 November 2014
Published 19 December 2014

Physicochemical properties of proteins always guide to determine the quality of the protein structure, therefore it has been rigorously used to distinguish native or native-like structure from other predicted structures. In this work, we explore nine machine learning methods with six physicochemical properties to predict the Root Mean Square Deviation (RMSD), Template Modeling (TM-score), and Global Distance Test (GDT_TS-score) of modeled protein structure in the absence of its true native state. Physicochemical properties namely total surface area, euclidean distance (ED), total empirical energy, secondary structure penalty (SS), sequence length (SL), and pair number (PN) are used. There are a total of 95,091 modeled structures of 4896 native targets. A real coded Self-adaptive Differential Evolution algorithm (SaDE) is used to determine the feature importance. The K-fold cross validation is used to measure the robustness of the best predictive method. Through the intensive experiments, it is found that Random Forest method outperforms over other machine learning methods. This work makes the prediction faster and inexpensive. The performance result shows the prediction of RMSD, TM-score, and GDT_TS-score on Root Mean Square Error (RMSE) as 1.20, 0.06, and 0.06 respectively; correlation scores are 0.96, 0.92, and 0.91 respectively; R^2 are 0.92, 0.85, and 0.84 respectively; and accuracy are 78.82% (with ± 1 err), 86.56% (with ± 0.1 err), and 87.37% (with ± 0.1 err) respectively on the testing data set. The data set used in the study is available as supplement at http://bit.ly/RF-PCP-DataSets.

Keywords: Physicochemical properties of protein; protein structure prediction; machine learning; random forest; SaDE; feature importance.

1. Introduction

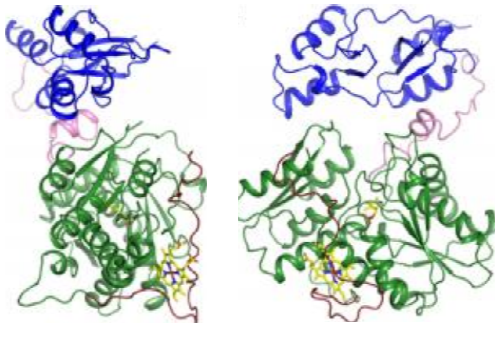
Protein sequences are translated into three-dimensional (3D) tertiary forms to carry out several biological functions. Prediction of high resolution protein structure has become one of the "grand challenge problems" in modern biology. Physicochemical

*Corresponding author.

properties of amino acids and their solvent environment are the key determinants in folding a protein sequence into its unique tertiary structure. These factors essentially generate various types of energy contributors such as electrostatic, van der Waals, salvation/desolvation which create folding pathways. *Ab initio* approaches for structure determination employ these physicochemical factors to generate a structure or an ensemble of structures from the sequence as plausible candidates for the native. In the alternative approach, called homology modeling, one uses experimentally known protein structures as templates based on sequence similarity. Due to lack of a clear understanding of the true folding pathway of proteins to the native and insufficient experimental data, several prediction methods end up with low quality structures. These low quality structures may look similar to any high resolution structure passing all the quality assessment criteria but in reality they could be 10–15 Å away from their true native states (Fig. 1). It would be highly desirable to have a predictive method which can tell how far a structure is from the native in the absence of its experimental structure.

Machine learning methods have been widely used in protein structure prediction such as two-dimensional (2D) and 3D structure prediction[1] fold recognition,[2] solvent accessibility prediction, disordered region prediction,[3] binding site prediction,[4] transmembrane helix prediction,[5] protein domain boundary prediction,[6] contact map,[7,8] functional site prediction, model generation,[9] and model evaluation.[10]

In this work, we explore nine machine learning methods with six physicochemical properties to predict the Root Mean Square Deviation (RMSD), Template Modeling (TM-score), and Global Distance Test (GDT_TS-score) of a modeled protein structure in the absence of its true native state. Physicochemical properties namely total surface area, euclidean distance (ED), total empirical energy, secondary structure penalty (SS), sequence length (SL), and pair number (PN) are calculated for each protein structure. Nine machine learning methods namely Random Forest, support vector machine (SVM), neural network (NN), linear model, M5P, cubist,



(a) Native structure (b) Predictive structure

Fig. 1. The distance of predicted structure from its native is 16.9 Å (PDB ID:2IS6).

foba, and decision stump are used for the prediction. Through the intensive experiments, it is found that Random Forest method outperforms over other machine learning methods. The K-fold cross validation is performed to measure the robustness of the best predictive model. To prove the effectiveness of the predictive model, the performance of best model is compared with top-performing ProQ2[11] and MetaMQAPII.[12]

Rest of the paper is organized as follows. A brief overview of the considered features, data set, methodology, Self-adaptive Differential Evolution algorithm (SaDE), and machine learning models are presented in Sec. 2. Model evaluation is presented in Sec. 3. Section 4 describes experiments, results, and discussion. Finally, conclusion is presented in Sec. 5.

2. Materials and Methods

2.1. *Data set and its features*

There are a total of 95,091 modeled structures of 4896 native targets. The modeled structures are taken from protein structure prediction center (CASP-5 to CASP-10 experiments), public decoys database[13] and native structure from protein data bank (RCSB). Table 1 describes the physicochemical properties used in this study. Table 2 shows the RMSD, TM-score, GDT_TS-score, and feature values of randomly selected targets from the dataset (dataset is available in the supplement). Table 3 shows the correlation between all features. There is high correlation between (i) ED and PN, (ii) SL and PN, and (iii) SL and area, where as the correlation of energy is very low with all the features. The Pearson correlation of all the features are plotted against

Table 1. Description of the features.

Feature	Information
Area	Total surface area
ED	Euclidean distance
Energy	Total empirical energy
SS	Secondary structure penalty
SL	Sequence length
PN	Pair number

Table 2. Sample dataset.

Target	RMSD	TM	GDT	Area	ED	Energy	SS	SL	PN
T0612_Atome2_CBS_TS1	8.65	0.44	0.44	6713.12	7348.12	−153.87	51	107	431
T0523_panther_TS2	7.49	0.67	0.64	6712.69	11,135.7	−1671.26	146	111	651
T0453_PS2-server_TS1	1.61	0.84	0.82	5311.88	3700.43	−925.86	118	85	269
T0453_LEE-SERVER_TS2	1.42	0.88	0.87	5309.69	3714.65	−185.26	65	85	269
NATIVE_4H89	0	1	1	8920.44	25,745	−4448	85	168	1358
NATIVE_3IBZ	0	1	1	8914.8	26,478	−3501	49	177	1471

Table 3. Correlation between all features.

	Area	ED	Energy	SS	SL	PN
Area	1.000	0.803	0.002	0.656	0.942	0.837
ED	0.803	1.000	0.001	0.514	0.838	0.953
Energy	0.002	0.001	1.000	0.003	0.002	0.001
SS	0.656	0.514	0.003	1.000	0.670	0.572
SL	0.942	0.838	0.002	0.670	1.000	0.913
PN	0.837	0.953	0.001	0.572	0.913	1.000

each other and as well as against RMSD, TM-score, and GDT_TS-score (refer to Supplement Document:S1).

2.2. *Qualitative assessment*

The RMSD, TM-score, and GDT_TS-score assess the quality of the protein structure relative to the experimental structure.

2.2.1. *RMSD*

The RMSD is calculated using the superposition between matched pairs of $C\alpha$ between two protein sequences. This superposition is computed using the Kabsch rotation matrix.[14,15] The RMSD is calculated as

$$\text{RMSD} = \sqrt{\sum_{i}^{N}(d_i * d_i)/N}, \tag{1}$$

where d_i is the distance between matched pair i, N is the number of matched pairs. RMSD is calculated using the freely available program at.[16]

2.2.2. *TM-score*

TM-score is an algorithm to calculate the structural similarity of two protein models.[17] It is used to quantitatively assess the accuracy of protein structure predictions relative to the experimental structure. TM-score weights the close atom pairs stronger than the distant matches, it is more sensitive to the topology fold than the often used RMSD since a local variation can result in a high RMSD value. TM-score has the value in (0,1] and independent on the length of the proteins. Based on statistics, the expected TM-score value for a random pair of proteins is ≤ 0.17 and for correctly aligned proteins ≥ 0.5. The TM-Score is calculated as

$$\text{TM score} = \frac{1}{L}\sum_{i=1}^{N}(1/(1 + d_i^2/d^2)), \tag{2}$$

where d_i is the distance between identical residues i, d is the distance threshold, N is the number of residue pairs, and L is the number of residues in the experimental structure. TM-Score is calculated using the freely available program at.[18]

2.2.3. *GDT_TS score*

GDT_TS-score[19,20] is another measure that quantitatively assess the accuracy of protein structure predictions relative to the experimental structure. It is a measure of similarity between two protein structures with identical amino acid sequences but different tertiary structures. GDT_TS-score has the value in (0,1]. Similar to TM-score, it is also independent on the length of the proteins. The GDT_TS-score is calculated as:

$$\text{GDT\_TS score} = (C_1 + C_2 + C_3 + C_4)/4N, \tag{3}$$

where C_1 is the number of residues superposed below (threshold/4), C_2 is the number of residues superposed below (threshold/2), C_3 is the number of residues superposed below (threshold), C_4 is the number of residues superposed below (2*threshold), N is total number of residues and threshold used for the GDT_TS-score is 4 Å. GDT_TS-score is calculated using the freely available program at.[18]

2.3. *Feature measurement*

Here, six physicochemical properties namely total surface area (Area), ED, total empirical energy (Energy), SS, SL, and PN are selected. A brief discussion on the selected properties is given below:

2.3.1. *Total surface area (Area)*

Protein folding is ruled by various driving forces, which seek towards minimization of its total surface area. Degree of these external forces depends on the surface of protein exposed to the solvent, which convey the strong dependency of free energy on solvent accessible surface area (SASA).[21] SASA has been widely used as one of the important properties to assess the quality of protein structures. Hydrophobic collapse is considered as a major factor in protein folding and this can be estimated as a loss of SASA of nonpolar residues. Each amino acid shows a different affinity to be found on the surface of the protein based on the functional groups present in its side chain.[22] Some questions arise with regard to the usage of SASA: (i) should it be the total area or is it the area of the nonpolar residues, (ii) what is the standard fixed value of SASA for a native structure, and (iii) is the rule of minimum area applicable to nonglobular proteins. Here, total SASA have been calculated using Lee and Richards[22] method as absolute value.

2.3.2. *ED*

Spatial positioning of $C\alpha$ or $C\beta$ atoms are a decisive factor in providing the 3D conformation of a protein structure. Recently, neighborhood profiles of $C\alpha$ atoms for each pair of residues have been characterized and observed to be invariant in 3618 native proteins suggesting certain universal geometrical constraints in their positioning.[23] Here, four aliphatic nonpolar residues are considered i.e. Alanine (ALA), Valine (VAL), Leucine (LEU), and Isoleucine (ILE); collectively they formed six

unique pairs among each other. Cumulative inter-atomic ED of their respective $C\beta$ atoms for aliphatic nonpolar residues were calculated for each residue pair. ED is given as

$$E_d = \sum_{i=1}^{n} \sum_{j=1}^{n} e_{ij}, \tag{4}$$

where n is the total number of aliphatic nonpolar residues; i and j are individual aliphatic nonpolar residues. e is the ED between i and j.

2.3.3. *Total empirical energy (Energy)*

The total empirical energy is the absolute sum of electrostatic force, van der Waals force and hydrophobic force.[24,25] Molecular dynamics simulation package AMBER12[26] is used to compute total empirical energy. It is computed as given below:

$$E^{ij}_{\text{elec}} = \frac{332 * qi * qj}{r_{ij}}, \tag{5}$$

$$E^{ij}_{vdW} = \frac{C^{ij}_{12}}{r^{12}_{ij}} - \frac{C^{ij}_{6}}{r^{6}_{ij}}, \tag{6}$$

$$E^{ij}_{\text{hyd}} = \frac{M^{ij}_{12}}{r^{12}_{ij}} - \frac{M^{ij}_{6}}{r^{6}_{ij}}, \tag{7}$$

where r_{ij} is the distance between pair of atoms i and j, $C^{ij}_{12} = \epsilon\sigma^{12}$, $C^{ij}_{6} = 2\epsilon\sigma^{6}$, σ is the van der Waals radii, ϵ is the well depth, $M^{ij}_{12} = \epsilon R^{12}$, $M^{ij}_{6} = \epsilon R^{6}$, R is the distance variable and ϵ is set to 1. Finally total empirical energy is given as

$$E_{\text{total}} = \sum_{i}^{n-1} \sum_{j=i+1}^{n} (E^{ij}_{\text{elec}} + E^{ij}_{vdW} + E^{ij}_{\text{hyd}}). \tag{8}$$

2.3.4. *SS*

Secondary structure prediction has reached to 82% accuracy[27–29] over the last few years. Therefore, deviation from ideal predicted secondary structures can be used as a measure to quantify the quality of a structure. SS is measured from the secondary structure sequence. It is computed as the missmatches in the helix, sheet, and coil of the STRIDE[30] and the PSIPRED[31] prediction. STRIDE get the actual number of helix, sheet, and coil present in the secondary structure sequence where as PSIPRED uses NN to predict the probability for the same secondary structure classes.

$$\begin{aligned} \text{SS} &= \sum_{i=1}^{n} q_i, \\ q_i &= \begin{cases} 1 & \text{if } S_{\text{stride}}(P_i) = S_{\text{psipred}}(P_i) \\ 0 & \text{otherwise} \end{cases}, \end{aligned} \tag{9}$$

where P is the protein secondary structure sequence; $S_{\text{stride}}(P)$ and $S_{\text{psipred}}(P)$ are the number of helix, sheet, and coil returned by STRIDE and PSIPRED respectively for each amino acid P_i. SS is calculated by counting the total number of miss-matches found. It is found that SS has lower value for native and higher for non-native structure.

2.3.5. *SL*

SL is the total number of amino acid present in the protein structure. It is calculated from actual sequence.

2.3.6. *PN*

PN is the total number of aliphatic hydrophobic residue pairs in the protein structure and it is calculated by counting the total number of pairs between the $C\beta$ carbons in the protein structure.

2.4. *Approach*

The approach is described in Fig. 2. In the first phase, the modeled protein structures are taken from protein structure prediction center (CASP-5 to CASP-10 experiments), public decoys database[13] and native structure from protein data bank (RCSB). The feature measurement, as discussed in Sec. 2.3, of protein structures were carried out in the second phase. The removal of duplicates and missing value entries from the dataset were carried out in the third phase. There are a total of 95,091 modeled structures of 4896 native targets. In the fourth phase, a real coded SaDE[32] was used to measure the importance of each feature. Feature selection makes the prediction of model efficient and accurate. In the fifth phase, the nine machine learning methods (refer to Table 5) were trained and tested on the data set with their default parameters. Figure 3 describes the prediction method. Finally, the evaluation of the method is done on Root Mean Square Error (RMSE, Eq. (14)), correlation

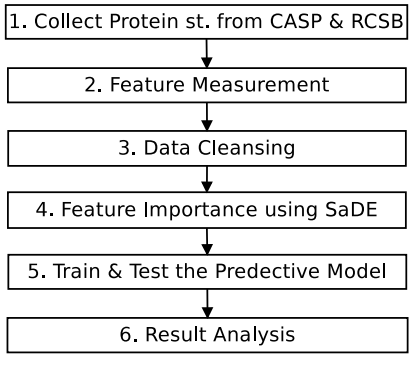


Fig. 2. Methodology used.

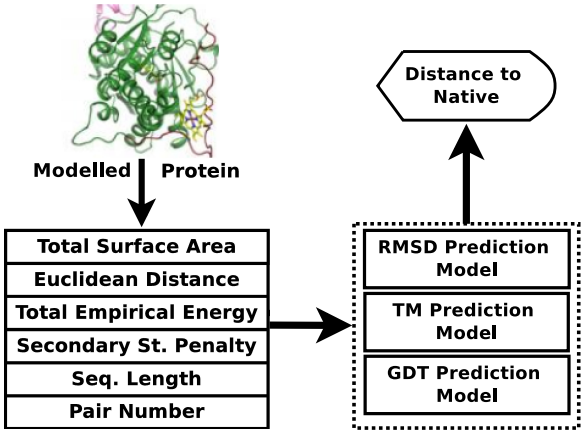


Fig. 3. Prediction method.

(Eq. (15)), R^2 (Eq. (16)), and accuracy (Eq. (17)). K-fold cross validation is used to measure the robustness of the best predictive method.

The key issue resolved through this approach is that many models have similar structures for the same target or same target may have different modeled structures. All the physicochemical properties of such structures may be the same for few cases and different for most of the cases, but removal of such duplicate entries ensures the uniqueness in the dataset.

2.4.1. *SaDE*

SaDE was proposed by Qin *et al.*,[32] where the control parameter values and the trial vector generation strategies are also gradually self-adapted by learning from their previous experiences in generating promising solutions. Consequently, it is possible to determine a more suitable generation strategy along with its parameter settings adaptively to match different phases of the search process/evolution.

In SaDE, four effective trial vector generation strategies namely the DE/rand/1/bin, DE/rand-to-best/2/bin, DE/rand/2/bin and DE/current-to-rand/1 were chosen to constitute a strategy candidate pool. The first three DE-variants are equipped with a binomial type crossover while the last one uses arithmetic recombination.

In the SaDE algorithm, for each target vector in the current population, one trial vector generation strategy is selected from the candidate pool according to the probability learned from its success rate in generating improved solutions (that can survive to the next generation) within a certain number of previous generations, called the learning period (LP). The selected strategy is subsequently applied to the corresponding target vector to generate a trial vector. More specifically, at each generation, the probabilities of choosing each strategy in the candidate pool are summed to 1. These probabilities are initially equal ($1/K$ for K

total number of strategies in the pool) and are then gradually adapted during evolution, based on the Success and Failure Rates[32] over the previous LP generations. The adaptations of the probabilities take place in such a fashion that, the larger the success rate for the kth strategy in the pool within the previous LP generations, the larger is the probability of applying it to generate trial vectors at the current generation.

2.4.2. *Feature importance using SaDE*

SaDE gives the optimum weight to each feature according to the objective function defined in Eq. (10) for RMSD, TM, and GDT_TS. The crossover rate (CR) and mutation rate (MR) are set to be 0.9 and 0.01 respectively.

$$\text{Obj fun} = \min\left(\sum_{i=1}^{T}\sqrt{\left(X_i - \sum_{j=1}^{n} w_j \cdot P_{i,j}\right)^2}\right), \tag{10}$$

where T is the total number of instances in training data set, X will be RMSD, TM or GDT_TS target, P is physicochemical properties, n is the number of properties (6 in this case) and w is the weight given to each feature defined in [0,1].

After five different runs, the average weight given to each feature is described in Table 4 for RMSD, TM, and GDT_TS and their average weight is used for ranking the features. It is found that energy has the highest ranking and ED has the lowest ranking.

2.4.3. *Machine learning methods*

In this work, we used nine machine learning methods (refer to Table 5) for prediction of near native protein structure. The methods are available in R open source software. R is licensed under GNU GPL. The brief detail of the methods are described as below:

(1) Decision Trees (rpart): This method is an extension of C4.5 classification algorithms described by Quinlan.[33]
(2) Random Forest (randomForest): It is based on a forest of trees using random inputs.[34]

Table 4. Importance of each feature using SaDE.

	Features					
Target	Energy	SS	Area	PN	ED	SL
RMSD	0.333	0.250	0.123	0.108	0.094	0.092
TM	0.242	0.160	0.194	0.125	0.121	0.158
GDT_TS	0.219	0.159	0.187	0.132	0.133	0.171
Avg.	0.265	0.190	0.168	0.122	0.116	0.140
Ranking	1	2	3	5	6	4

Table 5. Machine learning methods used.

Model	Method	Package	Tuning parameter and value(s)
Decision Tree[33]	rpart	rpart	MinSplit = 20, MaxDepth = 30, MinBucket = 7
Random Forest[34]	rf	randomForest	mtry = 500, sampling = bagging
SVM[35]	svm	e1071	nu = 10, epsilon = 0.5
LM[36]	lm	glm	None
NN[37]	neuralnet	neuralnet	hlayers = 10, MaxNWts = 10,000, maxit = 100
M5P[38]	M5P	RWeka	rules = 9, pruned = 15, smoothed = 0.6
Decision Stump[39]	Decision Stump	RWeka	rules = 6, pruned = 25, smoothed = 0.9
Cubist[40]	Cubist	Cubist	ommittees = 10, neighbors = 30
Foba[41]	Foba	Foba	lambda = 50, $k = 1000$

(3) SVM: SVMs yet represent a powerful technique for general (nonlinear) classification, regression, and outliers detection with an intuitive model representation.[35]
(4) Linear Models (GLM): It uses linear models to carry out regression, single stratum analysis of variance, and analysis of covariance.[36]
(5) NN: Training of NNs using back-propagation, resilient back-propagation with or without weight or the modified globally convergent version.[37]
(6) M5P: It is a tree learners classifier.[38]
(7) Decision Stump: It is based on one node decision trees and consider as weak learner.[39]
(8) Cubist: It is a regression modeling using rules with added instance-based corrections.[40]
(9) Foba: It is an implementation of forward, backward, and foba sparse learning algorithms for ridge regression.[41]

3. Model Evaluation

There are various ways to measure the performance of the prediction, where some are more suitable than others depending on the application considered. A brief discussion on the performance measures is explained below. Here, we created three different models to predict the three output variables (i.e. RMSD, TM-score, and GDT_TS-score) by using same number of input variables (i.e. features). The formula used for all the machine learning models is given by

$$\text{RMSD} \sim f(\text{Area},\ \text{ED},\ \text{Energy},\ \text{SS},\ \text{SL},\ \text{PN}), \tag{11}$$

$$\text{TM} \sim f(\text{Area},\ \text{ED},\ \text{Energy},\ \text{SS},\ \text{SL},\ \text{PN}), \tag{12}$$

$$\text{GDT\_TS} \sim f(\text{Area},\ \text{ED},\ \text{Energy},\ \text{SS},\ \text{SL},\ \text{PN}). \tag{13}$$

3.1. *RMSE*

RMSE is a popular formula to measure the error rate of a regression model. However, it can only be compared between models whose errors are measured in the same

units. It is calculated as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(p_i - a_i)^2}{n}}, \tag{14}$$

where a is actual target, p is predicted target, and n is the total number of instances.

3.2. *Correlation (r)*

Correlation describes the statistical relationships between actual and predicted values. It is defined as follows:

$$\text{Corr} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}, \tag{15}$$

where x is the actual value, y is the predicted value, $\bar{x}$ is the mean of the all actual values, $\bar{y}$ is the mean of the all predicted values, and n is the number of instances. Correlation lies in $[0, 1]$ and considered to be good if its value tends toward 1.

3.3. *Coefficient of determination (R^2)*

The coefficient of determination (R^2) summarizes the explanatory power of the regression model. R^2 describes the proportion of variance of the dependent variable explained by the regression model. If the regression model is perfect then R^2 is 1 and if the regression model is a total failure then R^2 is zero i.e. no variance is explained by regression. The Coefficient of Determination is computed by taking the square the r (i.e. correlation). It is defined as follows:

$$R^2 = r * r. \tag{16}$$

3.4. *Accuracy*

The accuracy is calculated as percentage deviation of predicted target with actual target with acceptable error.

$$\begin{aligned} \text{Accuracy} &= \frac{100}{n}\sum_{i=1}^{n} q_i, \\ q_i &= \begin{cases} 1 & \text{if } \text{abs}(p_i - a_i) \leq \text{err} \\ 0 & \text{otherwise} \end{cases}, \end{aligned} \tag{17}$$

where a is actual target, p is predicted target, err is the acceptable error, and n is the total number of instances.

3.5. *K-fold cross validation*

K-fold cross validation is used to measure the robustness of the predictive method. The original dataset is randomly partitioned into k equal size subsamples. Of the k

subsamples, a single subsample is retained as the validation data for testing the method, and the remaining $k-1$ subsamples are used as training data. The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data. The k results from the folds then can be averaged (or otherwise combined) to produce a single estimation. The advantage of this method over repeated random sub-sampling is that all observations are used for both training and validation, and each observation is used for validation exactly once.

3.6. *Benchmark of local model correctness*

For the benchmarking of model correctness, the performance of Random Forest with Physicochemical Properties (RF-PCP) model is compared with top-performing ProQ2[11] and MetaMQAPII.[12] Both the benchmark method are single-model method. ProQ2 is based on SVM that predict RMSD and TM-score where as MetaMQAPII is based on NN that predicts RMSD and GDT_TS-score.

4. Results

In this section, we analyze the prediction results of all the nine machine learning methods on the training–testing and validation dataset. For training–testing experiment, dataset consist of protein structures from CASP-5 to CASP-9 experiments, public decoys database and native structure from protein data bank (RCSB). All the models are run on their default parameters (refer to Table 5) and evaluated on RMSE, correlation, R^2, and accuracy. Here, the dataset is low in features and very high in observation values. The K-fold cross validation is used to measure the robustness of the best predictive method.

The regression model may suffer from overfitting issue due to the possibility of criterion used for training the model is not the same as the criterion used to judge the efficacy of a model, so the validation experiment is performed on the CASP-10 dataset using best predictive model selected from training–testing experiment.

For the benchmarking of model correctness, the performance of RF-PCP model is compared with top-performing ProQ2[11] and MetaMQAPII[12] on 24 randomly selected datasets from CASP-10.

4.1. *Training–testing experiment*

The distribution of data in training–testing experiment are set to 70% and 30% respectively for all the methods (dataset is available in the supplement). Table 6 shows the comparative performance of all the methods in the prediction of RMSD, TM-score, and GDT_TS-score on RMSE, correlation, R^2 and accuracy. The performance result shows that Random Forest method outperforms over other machine learning methods in the prediction.

The RMSE is used to measure the difference between actual and predicted values. The RMSE is calculated using Eq. (14) and Table 6 shows the RMSE of all the

Table 6. Performance comparison of machine learning methods in the prediction of RMSD, TM-score, and GDT_TS-score on testing data set.

	RMSD				TM-score				GDT_TS-score			
Model	RMSE	Corr	R^2	Acc%	RMSE	Corr	R^2	Acc%	RMSE	Corr	R^2	Acc%
D Tree	3.16	0.72	0.61	68.00	1.16	0.70	0.64	70.21	1.19	0.76	0.67	70.77
R Forest	**1.20**	**0.96**	**0.92**	**78.82**	**0.06**	**0.92**	**0.85**	**86.56**	**0.06**	**0.91**	**0.84**	**87.37**
SVM	2.66	0.80	0.72	77.36	0.55	0.78	0.72	72.36	0.58	0.82	0.77	80.36
LM	3.70	0.54	0.49	47.56	1.70	0.52	0.49	50.55	1.79	0.57	0.59	47.55
NN	3.69	0.54	0.49	47.65	1.69	0.52	0.49	51.65	1.59	0.59	0.57	50.42
M5P	2.66	0.87	0.78	75.37	0.66	0.86	0.78	83.47	0.56	0.89	0.79	80.47
D Stump	4.34	0.31	0.28	31.42	1.34	0.33	0.28	33.24	1.14	0.39	0.28	32.26
Cubist	2.68	0.91	0.82	76.15	0.68	0.99	0.82	81.15	0.58	0.90	0.81	81.52
Foba	3.72	0.53	0.48	48.22	1.72	0.55	0.48	50.14	1.52	0.53	0.58	49.98

methods. The Random Forest has the lowest RMSE of 1.20, 0.06, and 0.06 in the prediction of RMSD, TM-score, and GDT_TS-score respectively on the testing dataset.

The correlation describes the statistical relationship between actual and predicted values. The correlation is calculated using Eq. (15) and Table 6 shows the correlation of all the methods. The Random Forest has the highest correlation of 0.96, 0.92, and 0.92 in the prediction of RMSD, TM-score, and GDT_TS-score respectively on the testing dataset.

The R^2 is the sum of squares regression between actual and predicted values. The R^2 is calculated using Eq. (16) and Table 6 shows the R^2 of all the methods. The Random Forest has the highest R^2 of 0.92, 0.85, and 0.84 in the prediction of RMSD, TM-score, and GDT_TS-score respectively on the testing dataset.

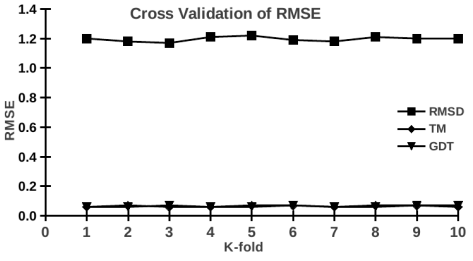
Accuracy is the degree of consistency of measured quantity to its true actual. The accuracy is calculated using Eq. (17) with some acceptable error and Table 6 shows the accuracy of all the methods. The Random Forest has the highest accuracy of 78.82% (± 1 err), 86.56% (± 0.1 err), and 87.37% (± 0.1 err) in the prediction of RMSD, TM-score, and GDT_TS-score respectively on the testing dataset.

Here, 10-fold cross validation is used to measure the robustness of the Random Forest. Figure 4 shows the RMSE, correlation, R^2, and accuracy for 10 folds in prediction of RMSD, TM-score, and GDT_TS-score. Cross validation result shows the uniform performance on all the model evaluation parameters.

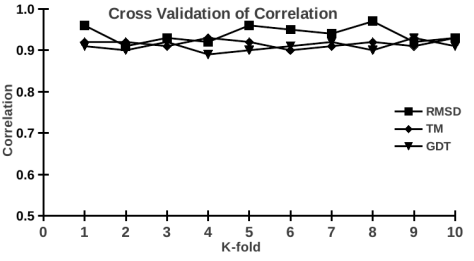
Figure 5 shows the scatter plot between actual and predicted of RMSD, TM-score, and GDT_TS-score on testing dataset using Random Forest.

4.2. *Validation experiment*

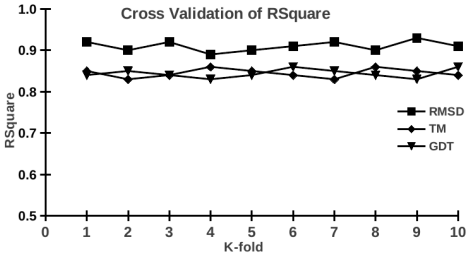
The validation experiment is performed on the CASP-10 dataset using Random Forest (best predictive model selected from training–testing experiment; CASP-10 dataset is available in the supplement). Table 7 shows the validation performance in



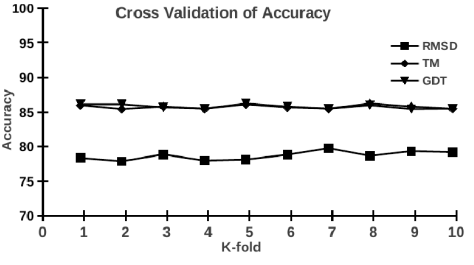
(a) RMSE



(b) Correlation

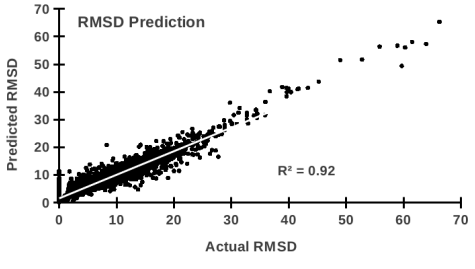


(c) R^2

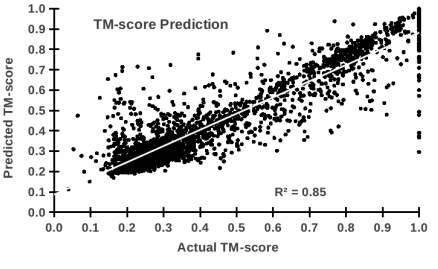


(d) Accuracy

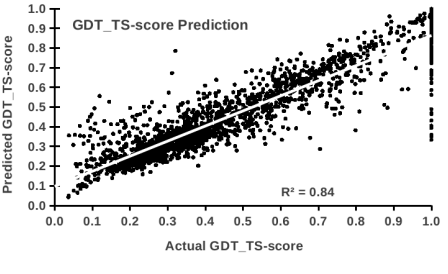
Fig. 4. 10-fold cross validation of RMSE, correlation, R^2, and accuracy on the testing data set in the prediction of RMSD, TM-score, and GDT_TS-score using Random Forest.



(a) RMSD prediction



(b) TM-score prediction



(c) GDT_TS-score prediction

Fig. 5. Scatter plot of actual versus predicted values of RMSD, TM-score, and GDT_TS-score on testing dataset using Random Forest.

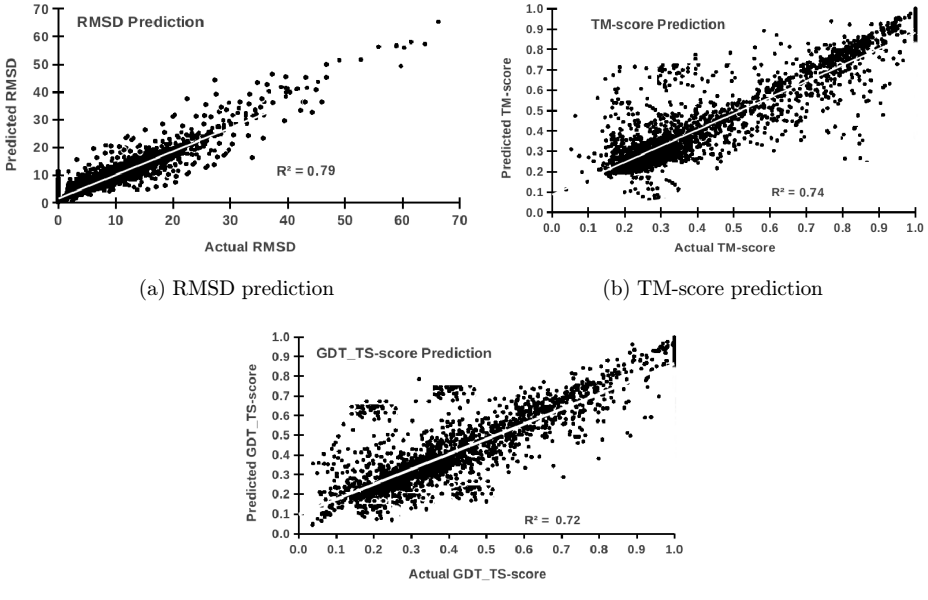
Table 7. Performance of Random Forest in the prediction of RMSD, TM-score, and GDT_TS-score on validation data set.

	RMSD				TM-score				GDT_TS-score			
Model	RMSE	Corr	R^2	Acc%	RMSE	Corr	R^2	Acc%	RMSE	Corr	R^2	Acc%
R Forest	2.68	0.89	0.79	70.82	0.07	0.86	0.74	71.22	0.07	0.85	0.72	70.13

the prediction of RMSD, TM-score, and GDT_TS-score on RMSE, correlation, R^2, and accuracy.

The RMSE are 2.68, 0.07, and 0.07 for RMSD, TM-score and GDT_TS-score respectively and a bit higher than the testing results. The correlation are 0.89, 0.86, and 0.85 for RMSD, TM-score, and GDT_TS-score respectively and a bit lower than the testing results. The R^2 are 0.79, 0.74, and 0.72 for RMSD, TM-score, and GDT_TS-score respectively and seem to be lower than the testing results. Accuracy are 70.82, 71.22, and 70.13 for RMSD, TM-score, and GDT_TS-score respectively and quite lower than the testing results. It is found that performance on validation data set is slightly lower than the training–testing experiment (Tables 6 and 7).

Figure 6 shows the scatter plot between actual and predicted of RMSD, TM-score, and GDT_TS-score on validation dataset using Random Forest.



(a) RMSD prediction

(b) TM-score prediction

(c) GDT_TS-score prediction

Fig. 6. Scatter plot of actual versus predicted values of RMSD, TM-score, and GDT_TS-score on validation dataset using Random Forest.

Table 8. Performance validation on the existing decoys sets in the prediction of RMSD, TM-score, and GDT_TS-score using Random Forest.

	RMSD				TM-score			GDT_TS-score		
CASP Target_ID	ACT[!]	PQ[+]	MM[*]	RF-PCP[#]	ACT[!]	PQ[+]	RF-PCP[#]	ACT[!]	MM[*]	RF-PCP[#]
T0649_Quark_Ts4	8.81	5.05	4.11	**11.04**	0.41	0.26	**0.51**	0.29	0.41	**0.46**
T0651_Bhageerathh_Ts5	20.14	15	8.6	**20.96**	0.21	0.04	**0.27**	0.1	**0.1**	0.13
T0654_Multicom-Construct_Ts2	3.44	**2.95**	5.48	1.67	0.7	0.51	**0.74**	0.63	0.42	**0.67**
T0671_Multicom-Construct_Ts5	13.18	7.1	4.18	**13.44**	0.4	0.15	**0.52**	0.22	0.41	**0.38**
T0674_Multicom-Cluster_Ts3	20.38	4.73	6.55	**20.28**	0.33	0.29	**0.35**	0.27	0.24	**0.26**
T0684_Zhang-Server_Ts3	18.83	**15**	3.4	11.73	0.22	**0**	0.55	0.12	0.51	**0.45**
T0688_Bilab-Enable_Ts1	3.2	2.81	**2.89**	2	0.81	0.53	**0.89**	0.69	**0.68**	0.86
T0690_Tasser-Vmt_Ts2	20.62	15	6.23	**20.57**	0.33	0	**0.34**	0.24	0.3	**0.24**
T0705_Zhou-Sparks-X_Ts1	21.57	8.2	5.45	**20.68**	0.41	0.19	**0.4**	0.19	0.3	**0.2**
T0713_Pms_Ts4	17.25	4.22	2.97	**27.52**	0.23	0.34	**0.33**	0.13	0.55	**0.18**
T0714_Multicom-Novel_Ts2	1.68	1.58	**1.64**	1.43	0.88	0.78	**0.85**	0.87	0.83	**0.84**
T0724_Chuo-Fams-Server_Ts5	26.92	15	10.54	**27.92**	0.25	0	**0.26**	0.19	0.03	**0.2**
T0735_Quark_Ts1	21.54	2.65	2.99	**17.04**	0.18	0.56	**0.42**	0.1	0.54	**0.26**
T0737_Phyre2_A_Ts3	12.87	**15**	6.53	3.17	0.46	0	**0.64**	0.4	0.21	**0.5**
T0746_Hhpreda_Ts1	7.16	9.39	**5.76**	13.22	0.64	0.09	**0.6**	0.47	0.29	**0.36**
T0651_Native	0	4.5	1.2	**0.04**	1	0.3	**0.91**	1	0.83	**0.89**
T0653_Native	0	2.99	1.3	**0.18**	1	0.5	**0.9**	1	0.84	**0.96**
T0671_Native	0	3.01	**1**	1.06	1	0.48	**0.8**	1	**0.91**	0.88
T0684_Native	0	3.45	**0.9**	1.07	1	0.43	**0.8**	1	**0.93**	0.86
T0690_Native	0	3.45	2.7	**0.01**	1	0.43	**0.99**	1	0.8	**1**
T0705_Native	0	3.41	1.6	**0.58**	1	0.44	**0.96**	1	**0.88**	0.84
T0713_Native	0	2.73	2.6	**0.42**	1	0.55	**0.98**	1	0.78	**0.95**
T0717_Native	0	2.67	3.2	**0.38**	1	**0.56**	0.53	1	0.66	**0.8**
T0724_Native	0	3.81	2.1	**1.12**	1	0.38	**0.73**	1	**0.8**	0.7

[!]Actual; [+]ProQ2; [*]MetaMQAPII; [#] RF-PCP: Random Forest with physicochemical properties.

For the benchmarking of model correctness, the performance of RF-PCP model is compared with top-performing ProQ2[11] and MetaMQAPII[12] on 24 randomly selected dataset from CASP-10 and the performance is found to be quite impressive (Table 8).

5. Conclusion

In this work, we explore nine machine learning methods with six physicochemical properties for estimating the absolute quality of a modeled protein structure in the absence of its true native state. The absolute quality of a model is expressed in terms of how well the model score agrees with the expected values from a representative set of high resolution experimental structures. The qualitative measure are RMSD, TM-score, and GDT_TS-score. Here, machine learning methods does not include any additional information from other models or alternative template structures. The dataset used in this study is low in features and very high in observation values. All the models are evaluated on RMSE, correlation, R^2, and accuracy. The K-fold cross validation is used to measure the robustness of the best predictive method. Through

the intensive experiments, it is found that Random Forest method outperforms over other machine learning methods. Random Forest is an ensemble method that uses bagging or boosting for sampling, so its performance is almost linear in K-fold validation.

Finally, for the benchmarking of model correctness, the performance of RF-PCP (Random Forest with Physicochemical Properties) model is compared with top-performing ProQ2 and MetaMQAPII. Both the benchmark methods are single-model method and it is found that the prediction accuracy of RF-PCP is higher. It is expected that optimizing the model parameters and adding more physicochemical properties may lead to better results.

Supplement Information

The data set used in the study is available as supplement at *http://bit.ly/RF-PCP-DataSets*. *RF-PCP-Training-Testing-CASP-5-9.csv* contain the dataset for training–testing experiment; *RF-PCP-Validation-CASP-10.csv* contain the data for validation experiment and *RF-PCP-dataset-Full.csv* contain the complete dataset.

References

1. Rost B, Sander C, Improved prediction of protein secondary structure by use of sequence profiles and neural networks, *Nat Acad Sci* **90**(16):7558–7562, 1993.
2. Jones DT *et al.*, GenTHREADER: An efficient and reliable protein fold recognition method for genomic sequences, *JMB* **287**(4):797–815, 1999.
3. Obradovic Z, Peng K, Vucetic S, Radivojac P, Dunker AK, Exploiting heterogeneous sequence properties improves prediction of protein disorder, *Proteins* **61**(S7):176–182, 2005.
4. Travers AA, DNA conformation and protein binding, *Ann Rev Biochem* **58**(1):427–452, 1989.
5. Krogh A, Larsson BÈ, Heijne GV, Sonnhammer ELL, Predicting transmembrane protein topology with a hidden Markov model: Application to complete genomes, *JMB* **305**(3):567–580, 2001.
6. Bryson K, Cozzetto D, Jones DT, Computer-assisted protein domain boundary prediction using the Dom-Pred server, *Curr Protein Peptide Sci* **8**(2):181–188, 2007.
7. Fariselli P, Olmea O, Valencia A, Casadio R, Prediction of contact maps with neural networks and correlated mutations, *Protein Eng* **14**(11):835–843, 2001.
8. Olmea O, Valencia A, Improving contact predictions by the combination of correlated mutations and other sources of sequence information, *Folding Des* **2**:S25–S32, 1997.
9. Simons KT, Kooperberg C, Huang E, Baker D, Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions, *JMB* **268**(1):209–225, 1997.
10. Wallner B, Elofsson A, Prediction of global and local model quality in CASP7 using Pcons and ProQ, *Proteins* **69**:184–193, 2007.
11. Ray A, Lindahl E, Wallner B, Improved model quality assessment using ProQ2, *Bioinformatics* **13**:224, 2012.
12. Pawlowski M, Gajda MJ, Matlak R, Bujnicki JM, MetaMQAP: A meta-server for the quality assessment of protein models, *BMC Bioinformatics* **9**:403, 2008.

13. Available at www.scfbio-iitd.res.in/software/pcsm/dataset/Public_Decoys.
14. Betancourt MR, Skolnick J, Universal similarity measure for comparing protein structures, *Biopolymers* **59**(5):305–309, 2001.
15. Kabsch W, A discussion of the solution for the best rotation to relate two sets of vectors, *Acta Crystallographica Section A* **34**(5):827–828, 1978.
16. Available at http://zhanglab.ccmb.med.umich.edu/TM-score/RMSD.f.
17. Zhang Y, Skolnick J, Scoring function for automated assessment of protein structure template quality, *Proteins* **57**(4):702–710, 2004.
18. Available at http://zhanglab.ccmb.med.umich.edu/TM-score/TMscore.f.
19. Zemla A, LGA: A method for finding 3D similarities in protein structures, *NAR* **31**(13):3370–3374, 2003.
20. Zemla A, Venclovas Č, Moult J, Fidelis K, Processing and analysis of CASP3 protein structure predictions, *Proteins* **37**:22–29, 1999.
21. Durham E, Dorr B, Woetzel N, Staritzbichler R, Meiler J, Solvent accessible surface area approximations for rapid and accurate protein structure prediction, *JMM* **15**(9):1093–1108, 2009.
22. Janin J, Surface and inside volumes in globular proteins, 1979.
23. Mittal A, Jayaram B, Backbones of folded proteins reveal novel invariant amino acid neighborhoods, *JBSD* **28**(4):443–454, 2011.
24. Arora N, Jayaram B, Strength of hydrogen bonds in a helices, *JCC* **18**:1245–1252, 1997.
25. Narang P, Bhushan K, Bose S, Jayaram B, Protein structure evaluation using an all-atom energy based empirical scoring function, *JBSD* **23**(4):385–406, 2006.
26. Götz AW, Williamson MJ, Xu D, Poole D, Le Grand S, Walker RC, Routine microsecond molecular dynamics simulations with AMBER on GPUs. 1. Generalized born, *J Chem Theory Comput* **8**(5):1542, 2012.
27. Biasini M, Bienert S, Waterhouse A, Arnold K, SWISS-MODEL: Modelling protein tertiary and quaternary structure using evolutionary information, *Nucleic Acids Res*, p. gku340, 2014.
28. Sen TZ, Jernigan RL, Garnier J, Kloczkowski A, GOR V server for protein secondary structure prediction, *Bioinformatics* **21**(11):2787–2788, 2005.
29. Kryshtafovych A, Moult J, Bales P, Bazan JF, Challenging the state of the art in protein structure prediction: Highlights of experiment CASP10 structures, *Proteins: Struct Funct Bioinform* **82**(S2):26–42, 2014.
30. Frishman D, Argos P, Knowledge based protein secondary structure assignment, *Proteins* **23**(4):566–579, 1995.
31. Jones DT, Protein secondary structure prediction based on position specific scoring matrices, *JMB* **292**(2):195–202, 1999.
32. Qin AK, Huang VL, Suganthan PN, Differential evolution algorithm with strategy adaptation for global numerical optimization, *IEEE Trans Evolution Comput* **13**(2):398–417, 2009.
33. Quinlan JR, Induction of decision trees, *Mach Learning* **1**(1):81–106, 1986.
34. Liaw A, Wiener M, Classification and regression by randomForest, *R News* **2**(3):18–22, 2002.
35. Keerthi SS, Gilbert EG, Convergence of a generalized SMO algorithm for SVM classifier design, *Mach Learning* **46**(1):351–360, 2002.
36. Chambers JM, Computational methods for data analysis, *Appl Stat* **1**(2):1–10, 1977.
37. Riedmiller M, Braun H, A direct adaptive method for faster backpropagation learning: The RPROP algorithm, *IEEE Int Conf Neural Nets*, pp. 586–591, 1993.
38. Frank E, Wang Y, Inglis S, Holmes G, Witten IH, Using model trees for classification, *Mach Learning* **32**(1):63–76, 1998.

39. Dettling M, Bühlmann P, Boosting for tumor classification with gene expression data, *Bioinformatics* **19**(9):1061–1069, 2003.
40. Rulequest: Data Mining with Cubist. Available at www.rulequest.com/cubist-info.html.
41. Zhang T, Adaptive forward-backward greedy algorithm for learning sparse representations, *IEEE Trans Inform Theory* **57**(7):4689–4708, 2011.

Prashant Singh Rana is a Project Scientist at IIT Delhi. He earned his PhD from ABV-Indian Institute of Information Technology and Management, Gwalior, India and his areas of research are Machine learning, Soft computing, Combinatorial problems, and Protein folding.

Harish Sharma is an Associate Professor at the Department of Computer Engineering, Rajasthan Technical University, Kota, India. He earned his PhD from ABV-Indian Institute of Information Technology and Management, Gwalior, India and his areas of research are Image processing, Computer vision, Soft computing, and Pattern recognition.

Mahua Bhattacharya is an Associate Professor at ABV-Indian Institute of Information Technology and Management, Gwalior, India and her areas of research are Image processing, Computer vision, Soft computing, and Pattern recognition.

Anupam Shukla is a Professor at ABV-Indian Institute of Information Technology and Management, Gwalior, India and his areas of research are Speech processing, Robotics, Soft computing, Artificial intelligence, and Bioinformatics.