MetaScore: A novel machine-learning based approach to improve traditional scoring functions for scoring protein-protein docking conformations

Keywords: Protein-protein Interactions, Protein-protein Docking, Scoring functions, Machine learning, Method combination

Yong Jung[1,2,5], Cunliang Geng[8], Alexandre M. J. J. Bonvin[8], Li C. Xue[8,9*], Vasant G. Honavar[1,2,3,4,5,6,7*]

[1]Bioinformatics & Genomics Graduate Program, Pennsylvania State University, University Park, PA 16802, USA;

[2]Artificial Intelligence Research Laboratory, Pennsylvania State University, University Park, PA 16802, USA;

[3]Center for Big Data Analytics and Discovery Informatics, Pennsylvania State University, University Park, PA 16823, USA;

[4]Institute for Computational and Data Sciences, Pennsylvania State University, University Park, PA 16802, USA;

[5]Huck Institutes of the Life Sciences, Pennsylvania State University, University Park, PA 16802, USA;

[6]Clinical and Translational Sciences Institute, Pennsylvania State University, University Park, PA 16802, USA;

[7]College of Information Sciences & Technology, Pennsylvania State University, University Park, PA 16802, USA;

[8]Bijvoet Centre for Biomolecular Research, Faculty of Science - Chemistry, Utrecht University, Padualaan 8, 3584 CH Utrecht, the Netherlands;

[9]Center for Molecular and Biomolecular Informatics, Radboudumc, Greet Grooteplein 26-28, 6525 GA Nijmegen, the Netherlands.

[*]Correspondence: Vasant G. Honavar (e-mail: vhonavar@psu.edu. Address: E335 Westgate Bldg. Pennsylvania State University, University Park, PA 16802-6823, USA. Phone: +1-814-865-3141), Li C. Xue (e-mail: Li.Xue@radboudumc.nl. Address: Route: 260, Radboud Institute for Molecular Life Sciences, Radboudumc, Geert Grooteplein Zuid 10, 6525 GA Nijmegen, the Netherlands. Phone : +31 6 18 59 43 90)

Abstract

Protein-protein interactions play a ubiquitous role in biological function. Knowledge of the three-dimensional (3D) structures of the complexes they form is essential for understanding the structural basis of those interactions and how they orchestrate key cellular processes. Computational docking has become an indispensable alternative to the expensive and time-consuming experimental approaches for determining 3D structures of protein complexes. Despite recent progress, identifying *near-native* models from a large set of conformations sampled by docking - the so-called scoring problem - still has considerable room for improvement.

We present here MetaScore, a new machine-learning based approach to improve the scoring of docked conformations. MetaScore utilizes a random forest (RF) classifier trained to distinguish *near-native* from *non-native* conformations using a rich set of features extracted from the respective protein-protein interfaces. These include physico-chemical properties, energy terms, interaction propensity-based features, geometric properties, interface topology features, evolutionary conservation and also scores produced by traditional scoring functions (SFs). MetaScore scores docked conformations by simply averaging of the score produced by the RF classifier with that produced by any traditional SF. We demonstrate that (i) MetaScore consistently outperforms each of nine traditional SFs included in this work in terms of success rate and hit rate evaluated over the top 10 predicted conformations; (ii) An ensemble method, MetaScore-Ensemble, that combines 10 variants of MetaScore obtained by combining the RF score with each of the traditional SFs outperforms each of the MetaScore variants. We conclude that the performance of traditional SFs can be improved upon by judiciously leveraging machine-learning.

1. Introduction

Proteins are among the most abundant, structurally diverse and functionally versatile biological macromolecules. They come in many sizes and shapes and perform a wide range of structural, enzymatic, transport, and signaling functions in cells[1]. But proteins rarely act alone as their functions are typically mediated by interactions with other molecules, including in particular, other proteins. Alterations in protein-protein interfaces leading to abnormal interactions with endogenous proteins, proteins from pathogens or both, are associated with many human diseases[2]. Protein interfaces have therefore become some of the most popular targets for rational drug design[3-5]. However, the development of effective therapeutic agents[6-9] to inhibit aberrant protein interactions requires detailed understanding of the structural, biophysical, and biochemical characteristics of protein-protein interfaces. The most reliable source of such information comes from X-ray crystallography[10] and nuclear magnetic resonance (NMR), which identify interfaces at the atomic level; alanine scanning mutagenesis, which identifies interfaces at the residue level; mass spectrometry-based approaches, e.g., chemical cross-linking and hydrogen/deuterium (H/D) exchange, which identify individual interfacial residues[11, 12]; NMR-based approaches[13], e.g., chemical shift perturbations, cross-saturation, and H/D exchange, which determine interfaces at the residue or atomic level[14] and cryo-electron microscopy (cryo-EM) which can directly image large macromolecular complexes in their native hydrated state[15]. However, because of the technical challenges and the high costs and efforts involved, there is still a large gap between the number of known protein-protein interactions and the availability of 3D structures for those[16]. Therefore, there is an urgent need for reliable computational approaches for predicting protein-protein interfaces and complexes.

Against this background, computational docking has emerged as a powerful tool for modelling 3D structures of protein–protein complexes[17]. Given 3D structures or models of putative protein-protein interaction partners, docking aims to generate 3D models of their complex. Docking involves two key steps: sampling of the interaction space between the protein molecules to generate docked models; and scoring of the docked conformations to distinguish near-native conformations from the sampled conformations. There has been much recent progress on both sampling as well as scoring[18, 19].

The scoring functions that have been developed for protein-protein docking can be broadly grouped into several categories[20]: 1) Physics-based scoring functions that typically consist of a linear combination of energy terms. Examples include those used in HADDOCK[21], pyDOCK[22], RosettaDock[23], ZRANK[24], IRAD[25], DFIRE[26], DFIRE2[27], PISA[28], and SWARMDOCK[29]; 2) Statistical potential-based scoring functions such as 3D-Dock[30], DFIRE[26, 27], DECK[31], SIPPER[32], and MJ3H[33] which typically convert distance-dependent pairwise atom-atom or residue- residue contacts distributions into potentials; 3) Complementarity e.g., of shape, energy, or physico-chemical characteristics[34-38], 4) Interface connectivity based scoring functions[39, 40]; 5) Evolutionary conservation based scoring functions, e.g., InterEvScore[41]; and 6) Machine learning based scoring functions that combine a wide range of features including residue propensity of interfaces, contact frequencies of residue pairs, evolutionary conservation, shape complementarity, energy terms, atom pair distance distributions, etc.[42-52] However, as evident from the results of recent CAPRI competitions[53], there is considerable room for improvement in both sampling and scoring[17, 54-56].

Against this background, we introduce MetaScore, an approach to scoring docking conformations that combines any existing scoring function with a random forest[57] (RF) classifier trained to discriminate between near native and non-native structures. The RF classifier utilizes a variety of features of the interface between the proteins in the docked conformation, including interaction propensity-based, physico-chemical, energy-based, geometric, connectivity-based, and evolutionary conservation features. We report results of experiments on a standard benchmark, the protein-protein docking benchmark version 5.0[58] (BM5), which show that MetaScore outperforms the original scoring function when the two are compared using the area under the curve of success rate (ASR) and area under the curve of hit rate (AHR) for the top 10 predicted conformations. We further describe an ensemble method, MetaScore-Ensemble, that combines the score produced by an RF classifier trained using features including scores of several traditional scoring methods and features of interfaces with the averaged score of the original scoring methods. This ensemble approach even outperforms MetaScore using any single original scoring method. We conclude that machine learning methods can complement traditional approaches to scoring docking conformations.

2. Materials and Methods

2.1. Training data set and preprocessing

We used the protein-protein docking benchmark version 4.0 (BM4)[59], which has both the bound and unbound structures of protein-protein complexes, for training in our experiments excluding antigen-antibody complexes and non-dimers. For each of the remaining (cases), decoy models (BM4 decoy set) were generated by HADDOCK running in ab initio mode using center of mass restraints following its standard three-stage docking protocol: rigid body docking, semi-flexible refinement, and water-refinement[60]. We then selected cases and their water-refined decoys using the following criteria: (1) A case has at least one decoy with acceptable or better quality (i.e., interface root mean squared deviation (*i*-RMSD) of the decoy is less than or equal to 4Å)[1]; (2) The number of interface residues in a conformation is greater than or equal to 10. Interfacial residues are determined using an alpha carbon-alpha carbon (CA-CA) distance of 8Å between two residues belonging to two different proteins in the conformation (a decoy or a bound form). Among the 176 cases in BM4, 63 cases with decoys HADDOCK generated and 45 cases with only bound structures remained. We labeled a decoy *near-native* if its *i*-RMSD relative to the bound form is less than or equal to 4Å. Otherwise, the decoy was labeled as *non-native*. This process yielded 1,221 *near-native* and 35,957 *non-native* conformations. We refer to this set as the BM4 decoy set. However, the proportion of *near-native* and *non-native* conformations is highly unbalanced. Hence, we further under-sampled the *non-native* conformations for each case so that the *near-native* to *non-native* ratio is 1:1 (after testing 1:1, 1:2, 1:4 and 1:8 using 10 fold case-wise cross-validation on the BM4 decoy set, *data not shown*). We chose *non-native* decoys whose *i*-RMSDs are greater than 14 Å for training a model (after searching and testing 4, 8, 14, and 18 Å as cutoffs, *data not shown*). Our final training set consists of 1,221 *near-native* models (*i*-RMSD $\leq$ 4Å) and 1,221 *non-native* models (*i*-RMSD > 14Å) for 108 cases.

2.2. Test data set and preprocessing

For independent testing, we used sets of decoys generated by HADDOCK from the 55 newly added docking cases to the BM5[58] (BM5 decoy set) and sets of decoys from CAPRI competitions between CAPRI 10 and CAPRI 30 excluding non-dimers (CAPRI score set)[53]. The CAPRI score set consists of decoys generated from different docking programs, which can represent an ideal set for validating scoring functions independently of the docking programs. The decoys and cases from the BM5 decoy set and CAPRI score set were filtered to the same process as that applied to the training data, BM4 decoy set. The resulting numbers of cases for BM5 decoy set and CAPRI score set are 9 and 17, respectively. The corresponding numbers for decoys were 216 *near-native* and 3,384 *non-native* conformations and 1,115 *near-native* and 3,485 *non-native* conformations for BM5 decoy set and CAPRI score set, respectively.

2.3. Comparison with State-of-the-Art Scoring Methods

We used 10 different state-of-the-art scoring functions to test the MetaScore approach: HADDOCK[21], iScore[52], DFIRE[26], DFIRE2[27], MJ3H[33], PISA[28], pyDOCK[22],

[1] Even if there is no acceptable decoy for the case, the bound structure of the case is used for training. But such a case cannot be used for evaluation of the scoring method.

SIPPER[32], SWARMDOCK[29], and TOBI's method (TOBI)[61]. Among them, HADDOCK, DFIRE2, PISA, pyDock, SWARMDOCK, and TOBI are physicochemical energy-based scoring functions. SIPPER and MJ3H are statistical potential-based functions. DFIRE is a function based on both physicochemical energy and statistical potential. iScore is a machine learning-based scoring function using a random walk graph kernel.

Both iScore and MetaScore rely on machine learning. However, unlike MetaScore which uses various features of interfaces of native and non-native protein-protein conformations to train classifiers that discriminate between native and non-native conformations, iScore utilizes node labeled graphs to incorporate the details of interfaces. Furthermore, MetaScore is an ensemble technique which can be applied to any combination of scoring functions, including iScore.

2.4. Evaluation Metrics

The performance of a scoring method to correctly rank decoys based on *i*-RMSD was evaluated using two metrics: The success rate (the percentage of cases that have at least one near-native conformation among the top *N* conformations) and the hit rate (the overall percentage of near-native conformations that are included among the top *N* conformations). Both were calculated for an increasing number of predictions *N* varying between 1 and 400. For easier comparisons, area under the curve of success rate (ASR) and area under the curve of hit rate (AHR) were computed from the plots of corresponding success rate and hit rate respectively for *N* between 1 and 400 predictions. We focus on curves of ASRs and AHRs for the top 10 and top 400 predictions because top 10 decoys are considered for further analysis in the biologists' perspective[44] and CAPRI[56] competitions also allow them to be submitted for the next evaluation, and because 400 are the total number of decoys HADDOCK generally generates at its final stage for a case. All metrics are normalized between 0 and 1.

2.5. MetaScore, a novel approach combining scores from machine learning classifier based scoring function with scores from a traditional scoring function

MetaScore is an approach that combines the random forest (RF) based score produced from our RF classifier trained using several features with the score from a traditional scoring function.

2.5.1. The RF classifier. We trained an RF classifier using a diverse set of features of the interfaces between the interacting partners in decoys of our training data set to discriminate between *near-native* and *non-native* conformations. Random forest (RF) is an ensemble tree-structured classifier which is used for a data set with a large number of training data points and input features[57]. A random forest has two hyperparameters, *ntrees* (the number of trees to grow) and *mtry* (the number of features randomly selected as candidates at each split in a tree). They were optimized using a grid search approach; the value of *ntrees* was set from 10 to 500 with a step length of 10 and the value of *mtry* was set from 1 to 28 with a step length of 3. The hyperparameter optimization accompanies every RF model trained in different situations such as training with different feature sets, combining with different traditional scoring methods, and so on. The trained RF classifier outputs a probability for a decoy being *non-native*. The lower an RF score for a decoy, the more likely it to be *near-native* according to the RF classifier.

2.5.2. The Min-Max normalization within each case. Before combining the scores from different scoring functions including the RF score, we normalized the scores of decoys for each

case from each scoring function using the Min-Max normalization method. Min-Max normalization scales a list of data from 0 to 1. The minimum value in the data is mapped to 0 and the maximum one in the data is mapped to 1. The strength of this method is that all relationships among the data values can be preserved exactly and that any potential bias is not introduced into the data[62]. However, the Min-Max normalization is vulnerable to outliers in the original data, e.g., scores of decoys which have clashes. The resulting normalized values may fluctuate with existence of outliers in the data set. Before applying the Min-Max normalization, we defined values that fall outside two standard deviations of the mean in the data (here, scores of decoys within a case from a scoring method) as outliers. We forced outliers in the upper side of the data to be assigned 1 and those in the lower side to be assigned 0 as a normalized value. Then, we applied the Min-Max normalization into the remaining original data.

A normalized value (z) for x in a set of decoy scores for a case, X, using this method is calculated as follows:

$$z = \frac{x - min\ (X)}{max(X) - min(X)}$$

where $min(X)$ and $max(X)$ are the minimum and maximum values in the X given its range excluding outliers.

2.5.3. The final score of MetaScore. The final score is obtained by simply averaging the normalized scores of a decoy from the different scoring methods.

2.6. Features of MetaScore

We used seven types of features to encode protein-protein interfaces, each of which has been shown to be useful for characterizing properties of protein-protein interface residues[63, 64]. We extracted the following features for the binding site formed by the interacting partners in each decoy: i) Raw and normalized scores from each scoring function (Score features), ii) Evolutionary features, iii) Interaction propensity based features (Statistical features), iv) Hydrophobicity (Physicochemical feature), v) Energy-based features, vi) Geometric features, and vii) Connectivity features (see below for detail). A decoy is represented by a feature vector formed by its corresponding features.

2.6.1. Raw and normalized scores from each scoring function (Score features)

We included the raw scores and the normalized scores from each scoring function as part of MetaScore features, which are called Score features. Because different methods produce scores in different ranges, and even the scores assigned by a single method to decoys from different docking cases are in general incomparable, there is a need to normalize the scores. We applied Min-Max normalization method to normalize scores of decoys in each case for each scoring method. Contrary to the normalized score, the original scores from a classical scoring function also contain valuable information such as the size of interface region[65], the scoring function's expertise on how to combine its own multiple features related to binding process and so on. Therefore, it is expected that a combination of original scores and normalized scores can play roles as complementing each other on training a model. We therefore decided to use both original scores and normalized scores.

2.6.2. Evolutionary features

Binding sites tend to be highly conserved across species[64, 66, 67]. A scoring function that ranks decoys based on the degree to which their binding sites match the known or predicted binding sites of the target complex produces rankings that tend to place *near-native* conformations above *non-native* ones[52, 68]. Therefore, evolutionary conservation scores of interfacial residues in the binding sites are expected to contribute to classifying decoys into *near-native* decoys or *non-native* models.

We used Position-Specific Scoring Matrix Information Contents (PSSM-ICs) of interfacial residues as conservation scores. PSSM-IC is a measure of the information content for a residue in a PSSM based on Shannon's uncertainty using prior residue probability and relative frequency of the residue at a specific protein sequence position[69]. The higher a value of PSSM-IC of a residue, the more conserved the residue is. The PSSM-ICs are calculated from a result of multiple sequence alignment using PSI-BLAST[70]. We ran PSI-BLAST of BLAST 2.7.1+ against NCBI nr database (as of February 4, 2018) to retrieve the sequence homologs of each protein sequence using 3 iterations of PSI-BLAST with an e-value cutoff of 0.0001. Based on the length of the protein sequence, we automatically set "query length-specific" parameters, e.g., BLAST substitution matrix, word size, gap open cost and gap extend cost, according to a guideline provided in NCBI BLAST user manual (https://www.ncbi.nlm.nih.gov/books/NBK279684/) (see **Supplementary Table S1**). We collected PSSM-ICs for only interfacial residues between the interacting partners for each decoy and aggregated the PSSM-ICs into three types of representative values: average, minimum, and maximum of the PSSM-ICs for each and both of two proteins in a decoy. In total, 9 features were generated.

2.6.3. Interaction propensity-based features (Statistical features)

Previous studies[30, 31, 71-73] have shown that pair-wise amino acid interaction propensities provide useful information about interaction patterns of amino acids in complexes. We utilized interaction propensities of amino acid pairs in interfacial regions of protein-protein complexes, which were precomputed by InterEvScore[41]. The pre-calculated interaction propensities can be found in a supplementary table in the InterEvScore paper[41]. The interaction propensity of residue x and y, $IP(x, y)$, was defined as the ratio of the observed frequency in the protein-protein complexes and the expected frequency derived as the random probability to pick the interaction pair of x and y.

Also, we assumed that interaction propensities weighted by conservation scores and/or distances between interfacial residue pairs can be promising features by reflecting evolutionary closeness and geometrical tightness into the interaction propensity. We generated two additional interaction propensity-based features weighted by only conservation scores (IP_{PSSM}) and both conservation scores and distances between interfacial residue pairs ($IP_{PSSM,Dist}$). For each interfacial residue pair (x, y) in a decoy (D_i) which consists of protein A and B, IP_{PSSM} and $IP_{PSSM,Dist}$ are defined as:

$$IP_{PSSM}(x,y) = \sum_{m=1}^{20}\sum_{n=1}^{20} IP(m,n) \times PSSM_A(x,m) \times PSSM_B(y,n)$$

$$IP_{PSSM,Dist}(x,y) = \frac{IP_{PSSM}(x,y)}{Dist(x,y)}$$

where $Dist(x, y)$ represents the distance between residue x in protein A and residue y in protein B, $IP(x, y)$ represents the interaction propensity value for a pair of residue x and y that InterEvScore provides, and $PSSM_A(x, m)$ is the position-specific score corresponding to the value of the m-th amino acid in the 20-element vector for interfacial residue x in the PSSM profile from the sequence of protein A. All PSSM values were normalized by the sigmoid function.

Because the sizes of interfaces of different decoys are various, we summarized a list of values for each type of interaction propensity-based values (IP, IP_{PSSM}, and $IP_{PSSM,Dist}$) from interfacial residue pairs in a decoy by summation and averaging, which results in 6 features.

2.6.4. Hydrophobicity (Physicochemical feature)

Macromolecules' physicochemical properties play important roles for the forces of attraction or repulsion among them. Among various physicochemical properties, hydrophobicity has been widely used in not only scoring of docked conformations but also predicting binding sites[74-77]. Additionally, the role of hydrophobicity in protein folding/unfolding and interactions has been well known[78-80]. We assigned hydrophobicity values of amino acids from the AAIndex[81] database into all interfacial residues of both proteins in a decoy and average them to use as a feature.

2.6.5. Energy-based features

We used the Van der Waals, electrostatic, and empirical desolvation energies calculated by HADDOCK for a decoy[82]. We adopted both normalized and raw values of the energy-based features. Using only raw values for training the RF model is unfair because the values assigned to decoys from different docking cases are incomparable. On the other hand, using only normalized values can cause loss of valuable information implied such as the size and the true net energy produced in the interface of each decoy. For each normalized energy feature, we applied the same Min-Max normalization method.

2.6.6. Geometric features

2.6.6.1. Shortest distances of interfacial residue pairs. We assumed that a *near-native* decoy should be a tightly bound form of the proteins and that decoys would have short and uniform distances of interfacial residues between two different proteins if the two proteins form a tight complex. Hence, we used the shortest distances of interfacial residue pairs as features to reflect principle of shape complementarity for a decoy. Distances between alpha carbon atoms of the two interfacial residue pairs in a decoy were computed and we selected the top 10 shortest distances. The lower the values are, the more compact the decoy.

2.6.6.2. Convexity-to-concavity ratio. The CX value measures the ratio of the volume of atoms that occupy within a sphere with a radius of 10Å to the volume of empty space in the sphere[83]. It has been widely used in previous studies as a protrusion index[63, 84]. The smaller a CX value, the more protrude the atom and its 10 Å neighborhood are. We assumed that if the alpha-carbon atoms of interfacial residues in a protein of a decoy protrude, the ones in their partner interfacial residues in another protein of the decoy would be dented in a compact decoy, and *vice versa*. In

this light, higher convexity-to-concavity ratios using CX values for a pair of interfacial residues can indicate that either residue protrudes and the other one is dented. Keeping this in mind, we generated a feature, $CX_{ratio}(x, y)$, modifying the equation to calculate the ratio of CX values of alpha-carbon atoms of each interfacial residue pair (x, y).

Let I_{A0}, I_{A1}, ..., I_{An} denote a set of interfacial residues in a protein A of a decoy. Here, I_{Ai} where $1 \leq i \leq n$ is an interfacial residue in protein A, where n denotes the number of interfacial residues in the protein A. For each interfacial residue pair (I_{Ai}, I_{Bj}) of a decoy which consists of protein A and B, $CX_{ratio}(I_{Ai}, I_{Bj})$ is defined as:

$$\mathrm{CX}_{\mathrm{ratio}}\left(\mathrm{I}_{\mathrm{A}i}, \mathrm{I}_{\mathrm{B}j}\right) = \frac{max(\mathrm{CX}_{\mathrm{A}i}, \mathrm{CX}_{\mathrm{B}j})^2 + 1}{min(\mathrm{CX}_{\mathrm{A}i}, \mathrm{CX}_{\mathrm{B}j})^2 + 1}$$

where CX_{Ai} and CX_{Bj} represent CX values calculated by centering the 10Å sphere on alpha-carbon atoms of I_{Ai} and I_{Bj}, respectively.

$CX_{ratio}(I_{Ai}, I_{Bj})$ is larger than or equal to 1. The higher value of $CX_{ratio}(I_{Ai}, I_{Bj})$ can be regarded that the alpha-carbon atom of I_{Ai} or I_{Bj} protrudes and the alpha-carbon atom of another one is dented. The lower values of $CX_{ratio}(I_{Ai}, I_{Bj})$ can be considered that the both alpha-carbon atom of I_{Ai} and I_{Bj} protrude or are dented. Those CX-related values are obtained as many as the number of interfacial residue pairs in the decoy. We summarize them as forms of average and standard deviation, which ends up making a couple of features.

2.6.6.3. Buried surface area. The buried surface area (BSA)[82] is one of the HADDOCK-derived features. The BSA estimates the size of the interface between two proteins in a protein-protein complex. It can be obtained by calculating the difference between then entire solvent accessible surface area of two unbound proteins and that of a decoy. We used this value as one of the geometric features for training our model. Because the ranges of BSA differ by cases, we normalized BSA values by apply the Min-Max normalization method described above, excluding outliers.

2.6.6.4. Relative accessible surface area. The relative accessible surface area (rASA) of each interfacial residue was calculated using both its solvent accessible area obtained using STRIDE[85] and the known surface area of the residue[86]. The average of rASA values of the interfacial residues was used as a feature for a decoy.

2.6.6.5. Secondary structure. It is well known that particular secondary structures are preferred at protein interfaces[87, 88]. To capture the tendency of protein secondary structures to occur in the interface regions, we counted how many times different secondary structures appear in interfacial residues of a decoy structure. We used 7 secondary structure categories; Alpha Helix, 3-10 Helix, PI-Helix, Extended Conformation, Isolated Bridge, Turn, and Coil. Using STRIDE[85], we counted the occurrence of each secondary structure and normalized the occurrence by dividing it by the number of interfacial residues. In total, 7 features of secondary structures for a decoy were generated.

2.6.7. Connectivity features

To capture the connectivity of interfacial residues and the size of interface, we added three features: The number of interfacial residue pairs, the total number of interfacial residues and the

link density. The link density feature was implemented as defined in Basu et al.[89], which is a weighted number of interfacial residue pairs by the maximum number of possible links of interfacial residues between the two different proteins.

3. Results

3.1. Combination of scores from the RF classifier and scores from HADDOCK can improve the performance of HADDOCK scoring

To test our hypothesis that combining a machine learning model trained using potent interaction features with an existing scoring function can improve the performance of the original scoring function, we chose HADDOCK firstly as a representative of traditional scoring methods. We compared three scoring methods, HADDOCK, our RF classifier, and our MetaScore approach combining scores from HADDOCK and the RF classifier (MetaScore-HADDOCK) using 10 fold case-wise cross-validation with training set derived from BM4[59] (BM4 decoy set) and independent test procedures with sets of decoys from the newly added cases from BM5[58] (BM5 decoy set) and the CAPRI score set[53]. In the 10 fold case-wise cross-validation, a set of cases is randomly partitioned into 10 subsets. Of the 10 subsets, all decoys for cases in a single subset are retained as the test data and scored by a scoring method trained with decoys of cases from the remaining subsets. This process is repeated for all single subsets for testing in the cross-validation.

Table 1 and **Figs. 1-3** show that MetaScore-HADDOCK has better or at least comparable performance than the original method, HADDOCK, for all four performance metrics across all data sets we tested. The RF classifier itself, however, does not outperform HADDOCK for every data set and every evaluation method. Based on the observations, we conclude that the combination of scores from the RF classifier and HADDOCK could improve the scoring performance.

Table 1. Performance comparison of three methods, a classical scoring method (HADDOCK), machine learning-based scoring method using RF (RF classifier), and the combined method of the two methods (MetaScore-HADDOCK) using the BM4 decoy training set, BM5 decoy set, which is a set of decoys generated by HADDOCK from the newly added docking cases to the protein-protein docking benchmark version 5.0, and CAPRI score set[53].

Data sets	Method	ASR for top 10	AHR for top 10	ASR for top 400	AHR for top 400
BM4 decoy set	HADDOCK	0.29	0.036	0.85	0.64
	RF classifier	0.36	0.04	0.89	0.65
	MetaScore-HADDOCK	0.33	0.044	0.87	0.66
BM5 decoy set	HADDOCK	0.29	0.048	0.86	0.72
	RF classifier	0.38	0.032	0.89	0.65
	MetaScore-HADDOCK	0.44	0.056	0.9	0.72
CAPRI score set	HADDOCK	0.8	0.044	0.97	0.68
	RF classifier	0.72	0.032	0.97	0.65
	MetaScore-HADDOCK	0.8	0.044	0.97	0.68

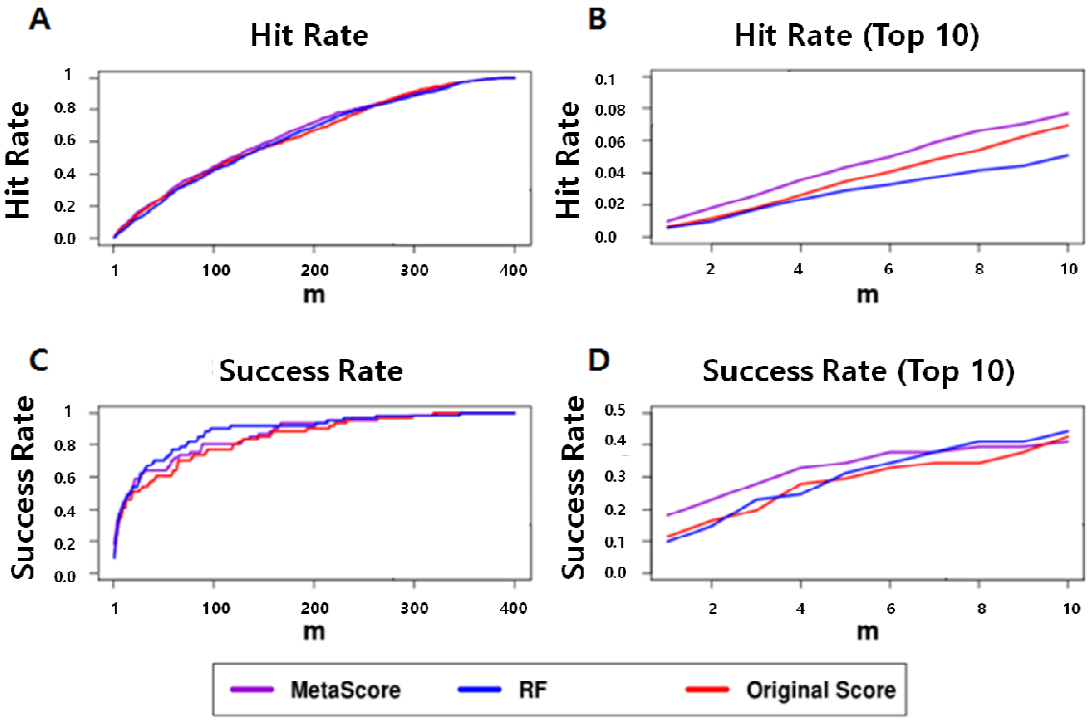


Figure 1. Success rates and hit rates plotted against the top m conformations for a classical scoring method (HADDOCK), machine learning-based method using RF (RF), and the combined method of the two methods (MetaScore) using the BM4 decoy training set. There are four panels. (A) Hit rates for conformations of top m ranging from 1 to 400; (B) Hit rates for conformations of top m ranging from 1 to 10; (C) Success rates for conformations of top m ranging from 1 to 400; (D) Success rates for conformations of top m ranging from 1 to 10.

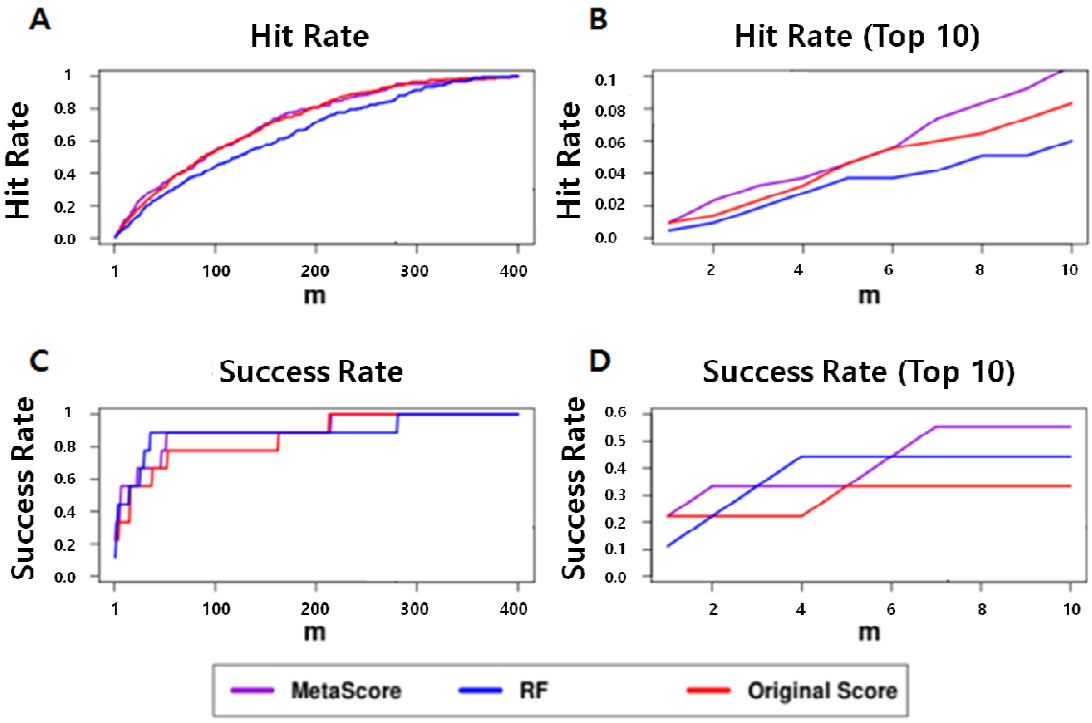


Figure 2. Success rates and hit rates plotted against the top m conformations for a classical scoring method (HADDOCK), machine learning-based method using RF (RF), and the combined method of the two methods (MetaScore) using BM5 decoy set. There are four panels. (A) Hit rates for conformations of top m ranging from 1 to 400; (B) Hit rates for conformations of top m ranging from 1 to 10; (C) Success rates for conformations of top m ranging from 1 to 400; (D) Success rates for conformations of top m ranging from 1 to 10.

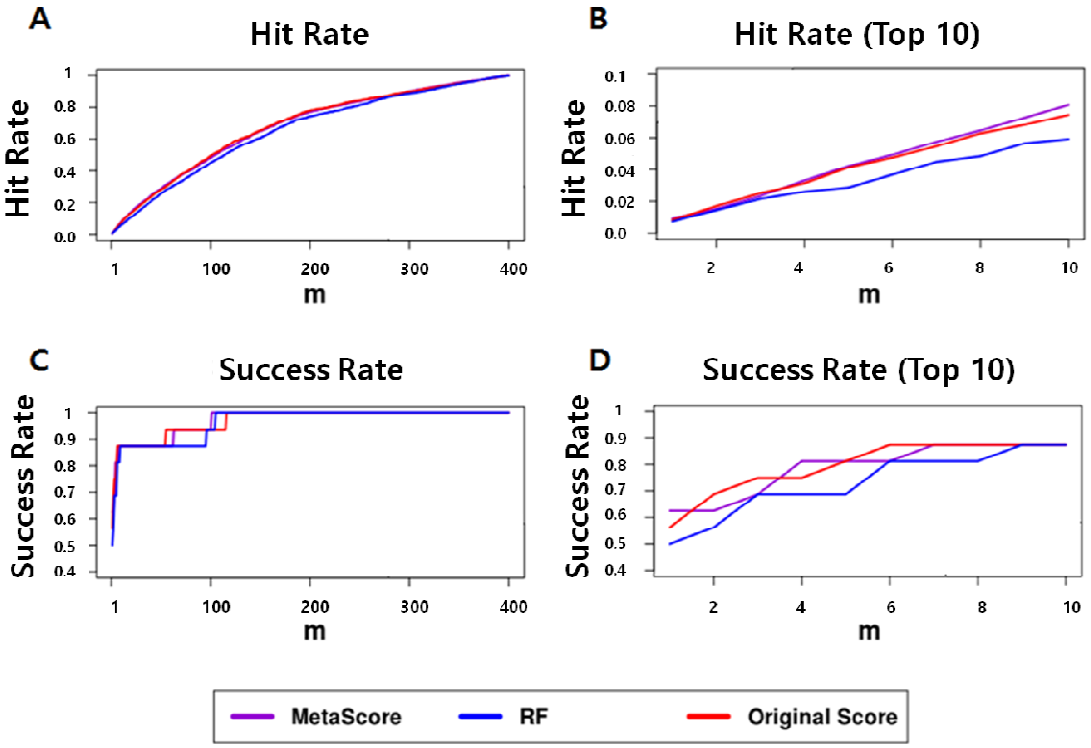


Figure 3. Success rates and hit rates plotted against the top m conformations for a classical scoring method (HADDOCK), machine learning-based method using RF (RF), and the combined method of the two methods (MetaScore) using CAPRI set. There are four panels. (A) Hit rates for conformations of top m ranging from 1 to 400; (B) Hit rates for conformations of top m ranging from 1 to 10; (C) Success rates for conformations of top m ranging from 1 to 400; (D) Success rates for conformations of top m ranging from 1 to 10.

3.2. Evaluation of feature importance

To train our RF classifier, we used various types of features of protein-protein interfaces that describe the interaction characteristics between a pair of proteins. We evaluated their impact on the performance of the RF classifier using 10-fold case-wise cross-validation and excluding in turn each of the seven feature types (**Table 2**).

Table 2. Scoring results by subtracting each feature type.

Method	Excluded feature type	ASR for top 10[1]	AHR for top 10	ASR for top 400	AHR for top 400
RF classifier	Connectivity features	0.28	0.044	0.87	0.65
	Statistical features	0.29	0.044	0.87	0.66
	Geometric features	0.3	0.044	0.87	0.66
	Score features	0.31	0.044	0.85	0.65
	Energy-based features	0.32	0.036	0.88	0.62
	Physicochemical feature	0.32	0.044	0.88	0.64
	Evolutionary features	0.34	0.048	0.87	0.65
	None	0.36	0.04	0.89	0.65
MetaScore-HADDOCK	Connectivity features	0.32	0.048	0.86	0.68
	Statistical features	0.34	0.044	0.86	0.67
	Geometric features	0.31	0.048	0.85	0.67
	Score features	0.32	0.048	0.86	0.67
	Energy-based features	0.34	0.048	0.86	0.66
	Physicochemical feature	0.31	0.048	0.85	0.65
	Evolutionary features	0.34	0.052	0.86	0.68
	None	0.33	0.044	0.87	0.66

[1]The results are ordered by decreased amount of ASR for top 10 in the RF classifier for each exclusion of feature types.

In the RF classifier, we found that the ASRs for top 10 and 400 predictions decreased for each feature type removed. Based on the ASR for top 10 predictions, which is a more focused evaluation metric for scoring methods, all feature types contribute to the performance of the RF classifier. Among the various types, Connectivity features is the feature type which contributes the best to the RF classifier but Evolutionary features is the least contributing feature type. Although the AHRs for top 10 and 400 predictions are not the best in the RF classifier using all features, the differences of the AHRs across most of the exclusion tests are insignificant in consideration of their standard deviation. We therefore determined to use the RF classifier using all features as our machine learning based model.

To see if feature combinations on training a machine learning model also affects the MetaScore-HADDOCK's performance, we evaluated MetaScore-HADDOCK by excluding each type of features individually in the part of training the machine learning based model. **Table 2** shows that the change of feature combinations has relatively little impact on the performance compared to the RF classifier based on the observation that standard deviations of the four performance measures in the MetaScore-HADDOCK are less than those in the RF classifier. Based on these results, we conjecture that combining scores from the two different scoring methods, the RF classifier and HADDOCK, helps reduce the change of the performance subject to changes among subsets of the entire feature set in the RF classifier. Although MetaScore-HADDOCK using all features does not show the best performance, we choose it as a final model because 1) difference of performance between the best performing MetaScore-HADDOCK which is trained without Evolutionary features and MetaScore-HADDOCK using all features is not statistically

significant within standard deviation and 2) the RF classifier trained with all features has the best performance in terms of ASR, which is the more relevant evaluation metric for scoring functions. This is because we conjecture that the best performing RF classifier has higher chance of resulting in better MetaScore.

3.3. Combination of RF classifier scores and scores from other scoring methods can improve the performance of each method

To test if MetaScore approach can be applicable to other methods, not only HADDOCK, we performed the same procedure using 9 previously published scoring functions, iScore[52], DFIRE[26], DFIRE2[27], MJ3H[33], PISA[28], pyDOCK[22], SIPPER[32], SWARMDOCK[29], and TOBI's method[61], respectively. We obtained scores from the 9 methods for decoys in our two data sets, BM4 decoy set and BM5 decoy set. For each scoring method, we replaced the normalized HADDOCK scores and the raw HADDOCK scores with the normalized scores and raw scores of the respective scoring methods, respectively and retrained our model with each set of scores. The resulting combined methods are called MetaScore-iScore, MetaScore-DFIRE, MetaScore-DFIRE2, MetaScore-MJ3H, MetaScore-PISA, MetaScore-pyDOCK, MetaScore-SIPPER, MetaScore-SWARMDOCK, and MetaScore-TOBI, respectively.

The results in **Table 3** show that our MetaScore approach for most original scoring methods improves their performance for both the BM4 decoy set, our training set, using 10-fold case-wise cross-validation and the BM5 decoy set, the test set, in terms of ASR and AHR evaluated over the decoys ranked among the top 10 predictions except for AHR of DFIRE using BM5 decoy set. Also, even though results of three methods (iScore, PISA and MJ3H) using BM4 decoy set and five methods (HADDOCK, DFIRE, DFIRE2, MJ3H, and PISA) using BM5 decoy set do not show the improvement in MetaScore in terms of ASR and AHR evaluated for the top 400 decoys ranked, the performances of MetaScore and the original methods are comparable or the decrease in performance is marginal (less than 2.56%) in the independent testing procedure using BM5 decoy set. (**Supplementary Figures S1-18**).

Table 3. Performance comparison of before and after combining classical scoring methods with each of their corresponding RF classifiers using the BM4 decoy training set and BM5 decoy set, which is a set of decoys generated by HADDOCK from the newly added docking cases to the protein-protein docking benchmark version 5.0. Our MetaScore approach improved the performance of all scoring functions we evaluated. Numbers in parentheses indicate percentages of increase from original methods. Values with no increase are highlighted in bold.

Data sets	Method	MetaScore Method				Original Method			
		ASR for top 10	AHR for top 10	ASR for top 400	AHR for top 400	ASR for top 10	AHR for top 10	ASR for top 400	AHR for top 400
BM4 decoy set	**HADDOCK**	0.33 (15.28%)	0.044 (22.22%)	0.87 (2.35%)	0.66 (3.13%)	0.29	0.036	0.85	0.64
	iScore	0.48 (4.34%)	**0.074 (-10.84%)**	**0.89 (0.00%)**	**0.71 (-2.74%)**	0.46	0.083	0.89	0.73
	DFIRE	0.32 (29.03%)	0.052 (44.44%)	0.84 (6.33%)	0.67 (3.08%)	0.25	0.036	0.79	0.65
	DFIRE2	0.27 (51.11%)	0.044 (57.14%)	0.85 (4.94%)	0.67 (6.35%)	0.18	0.028	0.81	0.63
	MJ3H	0.38 (9.20%)	0.056 (7.69%)	0.87 (2.35%)	**0.68 (-1.45%)**	0.35	0.052	0.85	0.69
	PISA	0.42 (6.12%)	0.064 (14.29%)	**0.89 (0.00%)**	0.71 (1.43%)	0.39	0.056	0.89	0.70
	pyDOCK	0.23 (42.50%)	0.028 (75.00%)	0.81 (8.00%)	0.63 (8.62%)	0.16	0.016	0.75	0.58
	SIPPER	0.26 (128.57%)	0.024 (100.00)	0.88 (7.32%)	0.62 (10.71%)	0.11	0.012	0.82	0.56
	SWARMDOCK	0.27 (103.03%)	0.028 (133.33%)	0.86 (3.61%)	0.61 (8.93%)	0.13	0.012	0.83	0.56
	TOBI	0.14 (133.33%)	0.012 (200.00%)	0.84 (12.00%)	0.54 (22.73%)	0.06	0.004	0.75	0.44
BM5 decoy set	**HADDOCK**	0.44 (52.79%)	0.056 (16.67%)	0.90 (4.65%)	**0.72 (0.00%)**	0.29	0.048	0.86	0.72
	iScore	0.42 (27.27%)	0.059 (47.5%)	0.86 (13.16%)	0.72 (1.41%)	0.33	0.040	0.76	0.71
	DFIRE	0.49	**0.064**	0.92	**0.77**	0.48	0.064	0.86	0.78

	(2.52%)	**(0.00%)**	(6.98%)	**(-1.28%)**				
DFIRE2	0.57 (8.40%)	0.072 (12.50%)	0.93 (6.90%)	**0.76 (0.00%)**	0.52	0.064	0.87	0.76
MJ3H	0.53 (70.51%)	0.056 (55.56%)	**0.91 (0.00%)**	0.57 (9.62%)	0.31	0.036	0.91	0.52
PISA	0.43 (1.89%)	0.064 (6.67%)	0.95 (3.26%)	**0.76 (-2.56%)**	0.42	0.060	0.92	0.78
pyDOCK	0.56 (31.13%)	0.068 (21.43%)	0.92 (10.84%)	0.77 (2.67%)	0.42	0.056	0.83	0.75
SIPPER	0.43 (68.75%)	0.060 (25.00%)	0.90 (4.65%)	0.72 (4.35%)	0.26	0.048	0.86	0.69
SWARMDOCK	0.22 (154.55%)	0.016 (300.00%)	0.86 (8.86%)	0.5 (35.14%)	0.09	0.004	0.79	0.37
TOBI	0.23 (163.64%)	0.028 (250.00%)	0.78 (16.42%)	0.47 (20.51%)	0.09	0.008	0.67	0.39

These results indicate that our proposed method, MetaScore, using a combination of an RF classifier and an existing original scoring method is likely to improve the performance of the original method.

3.4. Many heads are better than one

Ensembles of multiple predictive models are known to often outperform individual models[90-92]. We incorporated the ensemble approach into MetaScore to obtain MetaScore-Ensemble which combines several previously published methods: HADDOCK[21], iScore[52], DFIRE[26], DFIRE2[27], MJ3H[33], PISA[28], pyDOCK[22], SIPPER[32], SWARMDOCK[29], and TOBI's method[61], which is called "Expert Committee." To examine how the performance of MetaScore-Ensemble varies as a function of the performance of members in the ensemble, we used three scoring method groups (Groups in **Table 4)**: the higher performing group (ExpertsHigh), the lower performing group (ExpertsLow), and the members in the Expert Committee (Experts). ExpertsHigh and ExpertsLow were chosen based on the ASR and AHR for top 10 predictions obtained by 10 fold case-wise cross-validation using the BM4 decoy set, our training set. ExpertsHigh consists of HADDOCK, iScore, DFIRE, MJ3H, and PISA, and the ExpertsLow consists of the others. In addition, we used five ways of aggregating multiple scores (Approaches in **Table 4)**, to see the combination effect on MetaScore-Ensemble against each scoring method group:

1) **RF(Group)**, which is the RF classifier trained using only the raw scores and the normalized scores of members in a scoring method group (Group),
2) **RF(Group + Features)**, which is the RF classifier trained using our feature set of the protein-protein interfaces including the raw scores and the normalized scores of members in a Group,
3) **Avg(Group)**, which is a method averaging the normalized scores of members in a Group,
4) **Semi-MetaScore-Group**, which is a method combining the score from the RF classifier trained using only the raw scores and the normalized scores of members in a Group with the averaged score of the normalized scores of members in the Group,
5) **MetaScore-Group**, which is to combine the score from the RF classifier trained using our feature set of the protein-protein interfaces including the raw scores and the normalized scores of members in a Group with the averaged score of the normalized scores of members in the Group.

We tested fifteen MetaScore-Ensemble methods in total using combinations of three Groups and five Approaches (**Table 4**). For example, MetaScore-ExpertsHigh represents the one of MetaScore-Ensemble methods, which combines the score of the RF classifier trained using interaction features extracted from the protein-protein interfaces, the raw scores and the normalized scores of members in the ExpertsHigh Group with the averaged score of the normalized scores of the members in the ExpertsHigh Group.

Table 4. Category of scoring method groups and combination approaches for testing MetaScore-Ensemble methods.

Groups of Scoring Functions (Group)	Combination approaches (Approach)
ExpertsHigh	**RF(Group)[2]**
	RF(Group + Features[7])[3]
ExpertsLow	**Avg(Group)[4]**
Experts[1]	**Semi-MetaScore-Group[5]**
	MetaScore-Group[6]

[1]**Experts**: Ten published method including HADDOCK, iScore, DFIRE, DFIRE2, MJ3H, PISA, pyDOCK, SIPPER, SWARMDOCK, and TOBI's method; **ExpertsHigh**: HADDOCK, iScore, DFIRE, MJ3H, and PISA; **ExpertsLow**: DFIRE2, pyDOCK, SIPPER, SWARMDOCK, and TOBI's method.

[2]**RF(Group)**: The RF classifier trained using only the raw scores and the normalized scores of members in a Group; [3]**RF(Group + Features)**: The RF classifier trained using our feature set of the protein-protein interfaces including the raw scores and the normalized scores of members in a Group; [4]**Avg(Group)**: A method averaging the normalized scores of members in a Group; [5]**Semi-MetaScore-Group**: A method combining the score from the RF classifier trained using only the raw scores and the normalized scores of members in a Group with the averaged score of the normalized scores of members in the Group; [6]**MetaScore-Group**: A method combining the score from the RF classifier trained using our feature set of the protein-protein interfaces including the raw scores and the normalized scores of members in a Group with the averaged score of the normalized scores of members in the Group

[7]**Features**: Interaction features including Score features, Evolutionary features, Statistical features, Physicochemical feature, Energy-based features, Geometric features, and Connectivity features.

The comparison results using ASR and AHR on our independent test set, BM5 decoy sets, are shown in **Table 5.** The curves of success rates and hit rates are shown in **Fig. 4**. We can observe that most of the MetaScore-Ensemble methods perform better than other scoring functions including single traditional methods and MetaScore variants, and that the MetaScore-Experts**,** which is the MetaScore-Ensemble method using MetaScore-Group Approach applied to the Experts Group, has the best performance in both ASR and AHR for top 10 predictions.

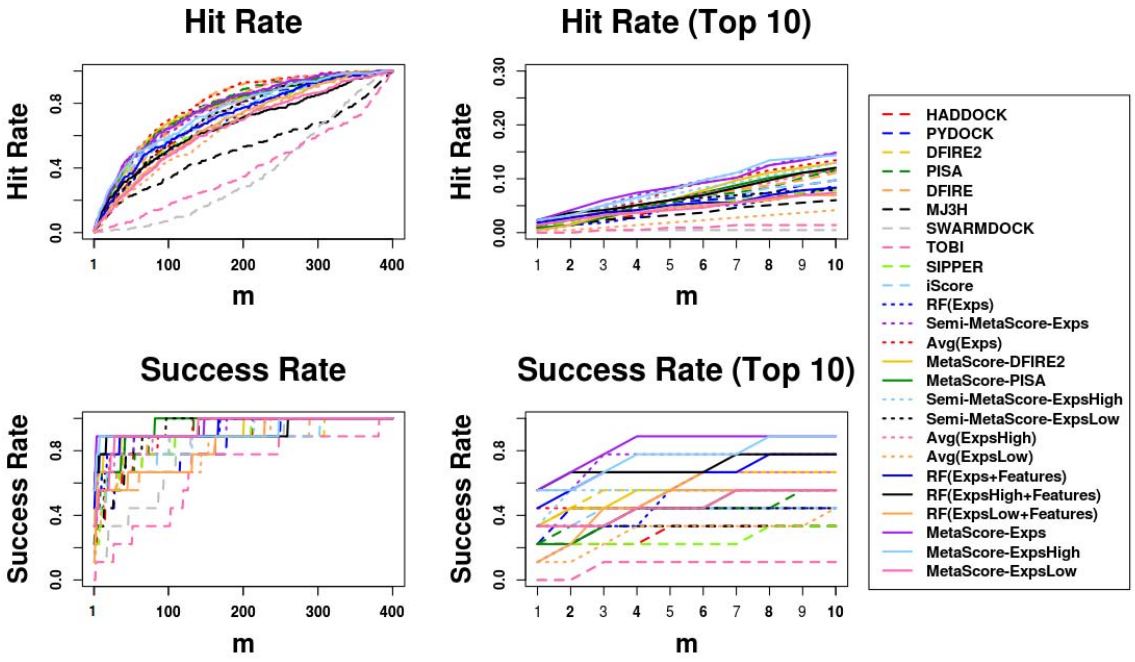


Figure 4. Success rates and hit rates plotted against the top m conformations for original methods, machine learning-based scoring methods combined with each original method, averaging method of Expert committee's scores, and machine learning-based scoring method using the Expert committee's scores combined with the averaging method of their scores using BM5 decoy set. The Expert committee has three groups, the high-ranked group (ExpsHigh), the low-ranked group (ExpsLow), and the group of entire members (Exps). There are four panels. (A) Hit rates for conformations of top m ranging from 1 to 400; (B) Hit rates for conformations of top m ranging from 1 to 10; (C) Success rates for conformations of top m ranging from 1 to 400; (D) Success rates for conformations of top m ranging from 1 to 10.

Table 5. Performance comparison of scoring methods including original methods, RF classifier variants, averaging method variants, MetaScore variants, Semi-MetaScore variants using BM5 decoy set, which is a set of decoys generated by HADDOCK from the newly added docking cases to the protein-protein docking benchmark version 5.0.

Methods	ASR for top 10[1]	AHR for top 10	ASR for top 400	AHR for top 400
MetaScore-Experts	0.82	0.088	0.96	0.77
MetaScore-ExpertsHigh	0.76	0.088	0.93	0.75
Semi-MetaScore-Experts	0.73	0.088	0.94	0.76
RF(ExpertsHigh + Features)	0.70	0.068	0.92	0.66
RF(Experts + Features)	0.67	0.052	0.94	0.71
MetaScore-DFIRE2	0.57	0.072	0.93	0.76
Avg(Experts)	0.57	0.080	0.93	0.8
MetaScore-pyDOCK	0.56	0.068	0.92	0.77
RF(ExpertsHigh)	0.54	0.060	0.89	0.72
MetaScore-MJ3H	0.53	0.056	0.91	0.57
Semi-MetaScore-ExpertsHigh	0.53	0.076	0.9	0.69
Avg(ExpertsHigh)	0.53	0.064	0.9	0.78
DFIRE2	0.52	0.064	0.87	0.76
MetaScore-DFIRE	0.49	0.064	0.92	0.77
DFIRE	0.48	0.064	0.86	0.78
MetaScore-ExpertsLow	0.46	0.044	0.94	0.65
RF(ExpertsLow + Features)	0.46	0.044	0.84	0.68
RF(Experts)	0.45	0.044	0.93	0.67
MetaScore-HADDOCK	0.44	0.056	0.9	0.72
MetaScore-PISA	0.43	0.064	0.95	0.76
MetaScore-SIPPER	0.43	0.060	0.9	0.72
pyDOCK	0.42	0.056	0.83	0.75
PISA	0.42	0.060	0.92	0.78
MetaScore-iScore	0.42	0.059	0.86	0.72
Semi-MetaScore-ExpertsLow	0.41	0.056	0.94	0.72
RF(Features)	0.38	0.032	0.89	0.65
iScore	0.33	0.040	0.76	0.71
MJ3H	0.31	0.036	0.91	0.52
RF(ExpertsLow)	0.31	0.024	0.85	0.64
HADDOCK	0.29	0.048	0.86	0.72
Avg(ExpertsLow)	0.29	0.020	0.84	0.65
SIPPER	0.26	0.048	0.86	0.69
MetaScore-TOBI	0.23	0.028	0.78	0.47
MetaScore-SWARMDOCK	0.22	0.016	0.86	0.5
TOBI	0.09	0.008	0.67	0.39
SWARMDOCK	0.09	0.004	0.79	0.37

[1]The results are ordered by ASR for top 10 predictions.

Note: **Features, Group, RF(Group), RF(Group + Features), Avg(Group), Semi-MetaScore-Group,** and **MetaScore-Group** are defined in **Table 4**.

Moreover, **Avg(Group)** applied to three Groups (ExpertsHigh, ExpertsLow, and Experts) outperforms each members in each group. Regardless of which Group is used, the **Avg(Group)** is outperformed by **RF(Group + Features)**, **Semi-MetaScore-Group**, and **MetaScore-Group**. Moreover, **MetaScore-Group** outperforms not only **Semi-MetaScore-Group** in every Group but also each of the MetaScore variants using each members in the corresponding Group. In addition, **RF(Group + Features)** which incorporates the features of interfaces for training the RF classifier outperforms **RF(Group)** which does not. Taken together, we can conclude that combining methods using any Approaches we tested except **RF(Group)** outperform individual methods, and that a machine learning model trained with additional features of interfacial regions outperforms a simple averaging method and a machine learning model not using features for interfacial regions in decoys.

Additionally, regardless of which one in the five Approaches is used, Approaches using the ExpertsHigh Group outperform ones using the ExpertsLow Group. In **Avg(Group)**, **Semi-MetaScore-Group** and **MetaScore-Group** Approaches**,** use of the Experts Group outperforms use of either ExpertsHigh or ExpertsLow Group. Except for **RF(Group)** and **Avg(Group)**, the Approaches using all members in the Experts Group were ranked in the top 5 methods in **Table 5**. As we expected, we observed that MetaScore-Ensemble methods which use better performing members can outperform ones that use less performing members, and that MetaScore-Ensemble methods using more members can perform better than using less members, except for MetaScore-Ensemble methods using only an RF classifier

4. Discussion

We have proposed a new approach, MetaScore, to rank docking models. The approach takes advantage of a machine learning-based classifier trained with widely used interaction features of interfacial regions to distinguish *near-native* conformations from *non-native* decoys. By simply averaging the score from the machine learning-based classifier trained using RF and the score from a traditional scoring method, we re-score the given models. By testing our approach on previously published scoring methods, we showed that the performance of the traditional scoring methods are improved.

When combining scores from two scoring methods, three scenarios in total can take place. First, by combining the scores from two scoring methods which both have good performance on scoring decoys, there is higher possibility of improving the ranking. Second, when the scores from a good and poor performing scoring methods are combined, incorrect ranking positions assigned by the latter one can be shifted closer toward correct positions by the better one. Third, if the two scores from two scoring methods are incorrect for ordering decoys, the combined score is still incorrect. Out of three cases, the first two are beneficial. Therefore, we can conclude that the main improvement of our approach comes from the synergistic/complementary effect.

Still, there is room for improvement of MetaScore in multiple perspectives. (1) Better performing machine learning-based classifiers could help MetaScore perform better. Better classifiers might be trained by using better combinations of different machine learning algorithms and/or different feature sets. Moreover, one of the attractive capabilities in machine learning algorithms is that they can manage a growing set of training data efficiently. Even

though the set of training data contains low-quality data, several algorithms are able to handle the noise associated with the low quality of the data. Also, because a larger training set tends to improve the prediction power of a model, MetaScore is expected to easily evolve with the increasing size of data from a variety of docking softwares. (2) The use of more effective combining methods can be helpful for further improvement. We showed here that already the simple averaging method for combining scores from the RF classifier and a traditional scoring method in MetaScore improved the performance compared to the traditional scoring method. More sophisticated combination methods such as a linear combination using weighted terms might further improve the results. Various combination methods developed in other research fields could be applied to the scoring problem for protein-protein docking[90, 92].

5. Conclusions

We have shown that MetaScore, a combination strategy of an RF classifier and an original scoring method, leads to the improvement of the original scoring method. We conducted experiments 1) to establish feature sets for training an RF classifier, 2) to confirm that MetaScore can improve the performance of the original scoring method, 3) to see if the strategy of MetaScore applied with a group of several published scoring methods can lead to significant improvement. Our results highlight that MetaScore consistently outperforms each of the traditional scoring functions we tested, and that the consensus model built by MetaScore-Ensemble can always perform better than not only each of original scoring methods but also MetaScore in combination with any single method in terms of success rate and hit rate evaluated over the conformations ranked among the top 10 predictions. We believe that our approach will be useful not only to boost the performance of an existing single scoring method but also to develop a powerful scoring method by applying our strategy into a group of best performing scoring methods.

CRediT authorship contribution statement

Yong Jung: Investigation, Methodology, Formal analysis, Validation, Visualization, Writing original draft. **Cunliang Geng**: Investigation, Data curation, Writing - review & editing, **Alexandre M. J. J. Bonvin**: Funding acquisition, Resources, Writing - review & editing, **Li C. Xue**: Funding acquisition, Supervision, Writing - review & editing, **Vasant G. Honavar**: Funding acquisition, Resources, Project administration, Writing - review & editing, Supervision

Declarations of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to affect the study reported in this paper.

Acknowledgements

Y.J. was supported in part by a research assistantship funded by the Center for Big Data Analytics and Discovery Informatics at Pennsylvania State University. The work of V.H. was supported in part by the National Center for Advancing Translational Sciences, National Institutes of Health through the grant UL1 TR000127 and TR002014, by the National Science Foundation, through the grants 1518732, 1640834 and 1636795, the Pennsylvania State University's Institute for Cyberscience and the Center for Big Data Analytics and Discovery Informatics, the Edward Frymoyer Endowed Professorship in Information Sciences and Technology at Pennsylvania State University and the Sudha Murty Distinguished Visiting Chair in Neurocomputing and Data Science funded by the Pratiksha Trust at the Indian Institute of Science. This work was also supported in part by the European H2020 e-Infrastructure grant BioExcel (grant no. 675728 and 823830) (A.M.J.J.B.). Financial support from the Netherlands Organisation for Scientific Research through an Accelerating Scientific Discovery (ASDI) from the Netherlands eScience Center (grant no. 027016G04) (L.X. and A.M.J.J.B.) and a Veni grant (grant no. 722.014.005) (L.X.) are acknowledged.

References

[1] P. Larrañaga *et al.*, "Machine learning in bioinformatics," (in eng), *Brief Bioinform,* vol. 7, no. 1, pp. 86-112, Mar 2006.

[2] D. P. Ryan and J. M. Matthews, "Protein-protein interactions in human disease," (in eng), *Curr Opin Struct Biol,* vol. 15, no. 4, pp. 441-6, Aug 2005, doi: 10.1016/j.sbi.2005.06.001.

[3] A. Metz, E. Ciglia, and H. Gohlke, "Modulating protein-protein interactions: from structural determinants of binding to druggability prediction to application," (in eng), *Curr Pharm Des,* vol. 18, no. 30, pp. 4630-47, 2012.

[4] D. González-Ruiz and H. Gohlke, "Targeting protein-protein interactions with small molecules: challenges and perspectives for computational binding epitope detection and ligand finding," (in eng), *Curr Med Chem,* vol. 13, no. 22, pp. 2607-25, 2006.

[5] B. Nisius, F. Sha, and H. Gohlke, "Structure-based computational analysis of protein binding sites for function and druggability prediction," (in eng), *J Biotechnol,* vol. 159, no. 3, pp. 123-34, Jun 2012, doi: 10.1016/j.jbiotec.2011.12.005.

[6] P. Zhou, C. Wang, Y. Ren, C. Yang, and F. Tian, "Computational peptidology: a new and promising approach to therapeutic peptide design," (in eng), *Curr Med Chem,* vol. 20, no. 15, pp. 1985-96, 2013.

[7] D. E. Szymkowski, "Creating the next generation of protein therapeutics through rational drug design," (in eng), *Curr Opin Drug Discov Devel,* vol. 8, no. 5, pp. 590-600, Sep 2005.

[8] J. Wanner, D. C. Fry, Z. Peng, and J. Roberts, "Druggability assessment of protein-protein interfaces," (in eng), *Future Med Chem,* vol. 3, no. 16, pp. 2021-38, Dec 2011, doi: 10.4155/fmc.11.156.

[9] Y. Jung *et al.*, "Identification of prognostic biomarkers for glioblastomas using protein expression profiling," (in eng), *Int J Oncol,* vol. 40, no. 4, pp. 1122-32, Apr 2012, doi: 10.3892/ijo.2011.1302.

[10] Y. Shi, "A glimpse of structural biology through X-ray crystallography," (in eng), *Cell,* vol. 159, no. 5, pp. 995-1014, Nov 2014, doi: 10.1016/j.cell.2014.10.051.

[11] A. N. Hoofnagle, K. A. Resing, and N. G. Ahn, "Protein analysis by hydrogen exchange mass spectrometry," (in eng), *Annu Rev Biophys Biomol Struct,* vol. 32, pp. 1-25, 2003, doi: 10.1146/annurev.biophys.32.110601.142417.

[12] S. Kaveti and J. R. Engen, "Protein interactions probed with mass spectrometry," (in eng), *Methods Mol Biol*, vol. 316, pp. 179-97, 2006, doi: 10.1385/1-59259-964-8:179.

[13] H. van Ingen and A. M. Bonvin, "Information-driven modeling of large macromolecular assemblies using NMR data," (in eng), *J Magn Reson*, vol. 241, pp. 103-14, Apr 2014, doi: 10.1016/j.jmr.2013.10.021.

[14] J. P. Rodrigues, E. Karaca, and A. M. Bonvin, "Information-driven structural modelling of protein-protein interactions," (in eng), *Methods Mol Biol*, vol. 1215, pp. 399-424, 2015, doi: 10.1007/978-1-4939-1465-4_18.

[15] P. I. Koukos and A. M. J. J. Bonvin, "Integrative Modelling of Biomolecular Complexes," (in eng), *J Mol Biol*, Nov 2019, doi: 10.1016/j.jmb.2019.11.009.

[16] R. Mosca, A. Céol, and P. Aloy, "Interactome3D: adding structural details to protein networks," (in eng), *Nat Methods*, vol. 10, no. 1, pp. 47-53, Jan 2013, doi: 10.1038/nmeth.2289.

[17] I. A. Vakser, "Protein-protein docking: from interaction to interactome," (in eng), *Biophys J*, vol. 107, no. 8, pp. 1785-1793, Oct 2014, doi: 10.1016/j.bpj.2014.08.033.

[18] H. Park, H. Lee, and C. Seok, "High-resolution protein-protein docking by global optimization: recent advances and future challenges," (in eng), *Curr Opin Struct Biol*, vol. 35, pp. 24-31, Dec 2015, doi: 10.1016/j.sbi.2015.08.001.

[19] M. M. Gromiha, K. Yugandhar, and S. Jemimah, "Protein-protein interactions: scoring schemes and binding affinity," (in eng), *Curr Opin Struct Biol*, vol. 44, pp. 31-38, 06 2017, doi: 10.1016/j.sbi.2016.10.016.

[20] C. Geng, L. C. Xue, J. Roel-Touris, and A. M. Bonvin, "Finding the ΔΔG spot: Are predictors of binding affinity changes upon mutations in protein–protein interactions ready for it?," *Wiley Interdisciplinary Reviews: Computational Molecular Science*, vol. 9, no. 5, p. e1410, 2019.

[21] C. Dominguez, R. Boelens, and A. M. Bonvin, "HADDOCK: a protein-protein docking approach based on biochemical or biophysical information," (in eng), *J Am Chem Soc*, vol. 125, no. 7, pp. 1731-7, Feb 2003, doi: 10.1021/ja026939x.

[22] T. M. Cheng, T. L. Blundell, and J. Fernandez-Recio, "pyDock: electrostatics and desolvation for effective scoring of rigid-body protein-protein docking," (in eng), *Proteins*, vol. 68, no. 2, pp. 503-15, Aug 2007, doi: 10.1002/prot.21419.

[23] S. Lyskov and J. J. Gray, "The RosettaDock server for local protein-protein docking," (in eng), *Nucleic Acids Res*, vol. 36, no. Web Server issue, pp. W233-8, Jul 2008, doi: 10.1093/nar/gkn216.

[24] B. Pierce and Z. Weng, "ZRANK: reranking protein docking predictions with an optimized energy function," (in eng), *Proteins*, vol. 67, no. 4, pp. 1078-86, Jun 2007, doi: 10.1002/prot.21373.

[25] T. Vreven, H. Hwang, and Z. Weng, "Integrating atom-based and residue-based scoring functions for protein-protein docking," (in eng), *Protein Sci*, vol. 20, no. 9, pp. 1576-86, Sep 2011, doi: 10.1002/pro.687.

[26] Y. Yang and Y. Zhou, "Specific interactions for ab initio folding of protein terminal regions with secondary structures," (in eng), *Proteins*, vol. 72, no. 2, pp. 793-803, Aug 2008, doi: 10.1002/prot.21968.

[27] Y. Yang and Y. Zhou, "Ab initio folding of terminal segments with secondary structures reveals the fine difference between two closely related all-atom statistical energy functions," (in eng), *Protein Sci*, vol. 17, no. 7, pp. 1212-9, Jul 2008, doi: 10.1110/ps.033480.107.

[28] S. Viswanath, D. V. Ravikant, and R. Elber, "Improving ranking of models for protein complexes with side chain modeling and atomic potentials," (in eng), *Proteins*, vol. 81, no. 4, pp. 592-606, Apr 2013, doi: 10.1002/prot.24214.

[29] I. H. Moal and P. A. Bates, "SwarmDock and the use of normal modes in protein-protein docking," (in eng), *Int J Mol Sci*, vol. 11, no. 10, pp. 3623-48, Sep 2010, doi: 10.3390/ijms11103623.

[30] G. Moont, H. A. Gabb, and M. J. Sternberg, "Use of pair potentials across protein interfaces in screening predicted docked complexes," (in eng), *Proteins*, vol. 35, no. 3, pp. 364-73, May 1999.

[31] S. Liu and I. A. Vakser, "DECK: Distance and environment-dependent, coarse-grained, knowledge-based potentials for protein-protein docking," (in eng), *BMC Bioinformatics*, vol. 12, p. 280, Jul 2011, doi: 10.1186/1471-2105-12-280.

[32] C. Pons, D. Talavera, X. de la Cruz, M. Orozco, and J. Fernandez-Recio, "Scoring by intermolecular pairwise propensities of exposed residues (SIPPER): a new efficient potential for protein-protein docking," (in eng), *J Chem Inf Model*, vol. 51, no. 2, pp. 370-7, Feb 2011, doi: 10.1021/ci100353e.

[33] S. Miyazawa and R. L. Jernigan, "Self-consistent estimation of inter-residue protein contact energies based on an equilibrium mixture approximation of residues," (in eng), *Proteins*, vol. 34, no. 1, pp. 49-68, Jan 1999.

[34] T. Geppert, E. Proschak, and G. Schneider, "Protein-protein docking by shape-complementarity and property matching," (in eng), *J Comput Chem*, vol. 31, no. 9, pp. 1919-28, Jul 2010, doi: 10.1002/jcc.21479.

[35] P. Mitra and D. Pal, "New measures for estimating surface complementarity and packing at protein-protein interfaces," (in eng), *FEBS Lett*, vol. 584, no. 6, pp. 1163-8, Mar 2010, doi: 10.1016/j.febslet.2010.02.021.

[36] H. A. Gabb, R. M. Jackson, and M. J. Sternberg, "Modelling protein docking using shape complementarity, electrostatics and biochemical information," (in eng), *J Mol Biol*, vol. 272, no. 1, pp. 106-20, Sep 1997, doi: 10.1006/jmbi.1997.1203.

[37] M. C. Lawrence and P. M. Colman, "Shape complementarity at protein/protein interfaces," (in eng), *J Mol Biol*, vol. 234, no. 4, pp. 946-50, Dec 1993, doi: 10.1006/jmbi.1993.1648.

[38] A. J. McCoy, V. Chandana Epa, and P. M. Colman, "Electrostatic complementarity at protein/protein interfaces," (in eng), *J Mol Biol*, vol. 268, no. 2, pp. 570-84, May 1997, doi: 10.1006/jmbi.1997.0987.

[39] S. Chang, X. Jiao, C. H. Li, X. Q. Gong, W. Z. Chen, and C. X. Wang, "Amino acid network and its scoring application in protein-protein docking," (in eng), *Biophys Chem*, vol. 134, no. 3, pp. 111-8, May 2008, doi: 10.1016/j.bpc.2007.12.005.

[40] R. Khashan, W. Zheng, and A. Tropsha, "Scoring protein interaction decoys using exposed residues (SPIDER): a novel multibody interaction scoring function based on frequent geometric patterns of interfacial residues," (in eng), *Proteins*, vol. 80, no. 9, pp. 2207-17, Aug 2012, doi: 10.1002/prot.24110.

[41] J. Andreani, G. Faure, and R. Guerois, "InterEvScore: a novel coarse-grained interface scoring function using a multi-body statistical potential coupled to evolution," (in eng), *Bioinformatics*, vol. 29, no. 14, pp. 1742-9, Jul 2013, doi: 10.1093/bioinformatics/btt260.

[42] A. J. Bordner and A. A. Gorin, "Protein docking using surface matching and supervised machine learning," (in eng), *Proteins*, vol. 68, no. 2, pp. 488-502, Aug 2007, doi: 10.1002/prot.21406.

[43] M. H. Chae, F. Krull, S. Lorenzen, and E. W. Knapp, "Predicting protein complex geometries with a neural network," (in eng), *Proteins*, vol. 78, no. 4, pp. 1026-39, Mar 2010, doi: 10.1002/prot.22626.

[44] T. Bourquard, J. Bernauer, J. Azé, and A. Poupon, "A collaborative filtering approach for protein-protein docking scoring functions," (in eng), *PLoS One*, vol. 6, no. 4, p. e18541, Apr 2011, doi: 10.1371/journal.pone.0018541.

[45] J. Azé, T. Bourquard, S. Hamel, A. Poupon, and D. W. Ritchie, "Using Kendall-τ meta-bagging to improve protein-protein docking predictions," in *IAPR International Conference on Pattern Recognition in Bioinformatics*, 2011: Springer, pp. 284-295.

[46] F. Fink, J. Hochrein, V. Wolowski, R. Merkl, and W. Gronwald, "PROCOS: computational analysis of protein-protein complexes," (in eng), *J Comput Chem*, vol. 32, no. 12, pp. 2575-86, Sep 2011, doi: 10.1002/jcc.21837.

[47] S. Basu and B. Wallner, "Finding correct protein-protein docking models using ProQDock," (in eng), *Bioinformatics*, vol. 32, no. 12, pp. i262-i270, 06 2016, doi: 10.1093/bioinformatics/btw257.

[48] H. Li, K. S. Leung, M. H. Wong, and P. J. Ballester, "Substituting random forest for multiple linear regression improves binding affinity prediction of scoring functions: Cyscore as a case study," (in eng), *BMC Bioinformatics*, vol. 15, p. 291, Aug 2014, doi: 10.1186/1471-2105-15-291.

[49] H. M. Ashtawy and N. R. Mahapatra, "A Comparative Assessment of Predictive Accuracies of Conventional and Machine Learning Scoring Functions for Protein-Ligand Binding Affinity Prediction," (in eng), *IEEE/ACM Trans Comput Biol Bioinform*, vol. 12, no. 2, pp. 335-47, 2015 Mar-Apr 2015, doi: 10.1109/TCBB.2014.2351824.

[50] B. Jiménez-García, J. Roel-Touris, M. Romero-Durana, M. Vidal, D. Jiménez-González, and J. Fernández-Recio, "LightDock: a new multi-scale approach to protein-protein docking," (in eng), *Bioinformatics*, vol. 34, no. 1, pp. 49-55, 01 2018, doi: 10.1093/bioinformatics/btx555.

[51] I. H. Moal *et al.*, "IRaPPA: information retrieval based integration of biophysical models for protein assembly selection," (in eng), *Bioinformatics*, vol. 33, no. 12, pp. 1806-1813, Jun 2017, doi: 10.1093/bioinformatics/btx068.

[52] C. Geng, Y. Jung, N. Renaud, V. Honavar, A. M. J. J. Bonvin, and L. C. Xue, "iScore: a novel graph kernel-based function for scoring protein-protein docking models," (in eng), *Bioinformatics*, vol. 36, no. 1, pp. 112-121, Jan 2020, doi: 10.1093/bioinformatics/btz496.

[53] M. F. Lensink and S. J. Wodak, "Score_set: a CAPRI benchmark for scoring protein complexes," (in eng), *Proteins*, vol. 82, no. 11, pp. 3163-9, Nov 2014, doi: 10.1002/prot.24678.

[54] M. F. Lensink and S. J. Wodak, "Docking, scoring, and affinity prediction in CAPRI," (in eng), *Proteins*, vol. 81, no. 12, pp. 2082-95, Dec 2013, doi: 10.1002/prot.24428.

[55] M. F. Lensink, S. Velankar, and S. J. Wodak, "Modeling protein-protein and protein-peptide complexes: CAPRI 6th edition," (in eng), *Proteins*, vol. 85, no. 3, pp. 359-377, 03 2017, doi: 10.1002/prot.25215.

[56] M. F. Lensink, S. Velankar, M. Baek, L. Heo, C. Seok, and S. J. Wodak, "The challenge of modeling protein assemblies: the CASP12-CAPRI experiment," (in eng), *Proteins*, vol. 86 Suppl 1, pp. 257-273, 03 2018, doi: 10.1002/prot.25419.

[57] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5-32, 2001.

[58] T. Vreven *et al.*, "Updates to the Integrated Protein-Protein Interaction Benchmarks: Docking Benchmark Version 5 and Affinity Benchmark Version 2," (in eng), *J Mol Biol*, vol. 427, no. 19, pp. 3031-41, Sep 2015, doi: 10.1016/j.jmb.2015.07.016.

[59] H. Hwang, T. Vreven, J. Janin, and Z. Weng, "Protein-protein docking benchmark version 4.0," (in eng), *Proteins*, vol. 78, no. 15, pp. 3111-4, Nov 2010, doi: 10.1002/prot.22830.

[60] S. J. de Vries, M. van Dijk, and A. M. Bonvin, "The HADDOCK web server for data-driven biomolecular docking," (in eng), *Nat Protoc*, vol. 5, no. 5, pp. 883-97, May 2010, doi: 10.1038/nprot.2010.32.

[61] D. Tobi, "Designing coarse grained-and atom based-potentials for protein-protein docking," (in eng), *BMC Struct Biol*, vol. 10, p. 40, Nov 2010, doi: 10.1186/1472-6807-10-40.

[62] Z. Liu, "A method of SVM with normalization in intrusion detection," *Procedia Environmental Sciences*, vol. 11, pp. 256-262, 2011.

[63] F. Minhas, B. J. Geiss, and A. Ben-Hur, "PAIRpred: partner-specific prediction of interacting residues from sequence and structure," (in eng), *Proteins*, vol. 82, no. 7, pp. 1142-55, Jul 2014, doi: 10.1002/prot.24479.

[64] L. C. Xue, D. Dobbs, A. M. Bonvin, and V. Honavar, "Computational prediction of protein interfaces: A review of data driven methods," (in eng), *FEBS Lett*, vol. 589, no. 23, pp. 3516-26, Nov 2015, doi: 10.1016/j.febslet.2015.10.003.

[65] A. Berchanski, B. Shapira, and M. Eisenstein, "Hydrophobic complementarity in protein-protein docking," (in eng), *Proteins*, vol. 56, no. 1, pp. 130-42, Jul 2004, doi: 10.1002/prot.20145.

[66] H. Geng, T. Lu, X. Lin, Y. Liu, and F. Yan, "Prediction of Protein-Protein Interaction Sites Based on Naive Bayes Classifier," (in eng), *Biochem Res Int,* vol. 2015, p. 978193, 2015, doi: 10.1155/2015/978193.

[67] Y. Jung, Y. El-Manzalawy, D. Dobbs, and V. G. Honavar, "Partner-specific prediction of RNA-binding residues in proteins: A critical assessment," (in eng), *Proteins,* vol. 87, no. 3, pp. 198-211, 03 2019, doi: 10.1002/prot.25639.

[68] L. C. Xue, R. A. Jordan, Y. El-Manzalawy, D. Dobbs, and V. Honavar, "DockRank: ranking docked conformations using partner-specific sequence homology-based protein interface prediction," (in eng), *Proteins,* vol. 82, no. 2, pp. 250-67, Feb 2014, doi: 10.1002/prot.24370.

[69] T. D. Schneider, G. D. Stormo, L. Gold, and A. Ehrenfeucht, "Information content of binding sites on nucleotide sequences," *Journal of molecular biology,* vol. 188, no. 3, pp. 415-431, 1986.

[70] S. F. Altschul *et al.*, "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs," (in eng), *Nucleic Acids Res,* vol. 25, no. 17, pp. 3389-402, Sep 1997.

[71] H. Lu, L. Lu, and J. Skolnick, "Development of unified statistical potentials describing protein-protein interactions," (in eng), *Biophys J,* vol. 84, no. 3, pp. 1895-901, Mar 2003, doi: 10.1016/S0006-3495(03)74997-2.

[72] S. Y. Huang and X. Zou, "An iterative knowledge-based scoring function for protein-protein recognition," (in eng), *Proteins,* vol. 72, no. 2, pp. 557-79, Aug 2008, doi: 10.1002/prot.21949.

[73] F. Nadalin and A. Carbone, "Protein–protein interaction specificity is captured by contact preferences and interface composition," *Bioinformatics,* vol. 34, no. 3, pp. 459-468, 2017.

[74] A. Axenopoulos, P. Daras, G. E. Papadopoulos, and E. N. Houstis, "SP-Dock: Protein-protein docking using shape and physicochemical complementarity," *IEEE/ACM transactions on computational biology and bioinformatics,* vol. 10, no. 1, pp. 135-150, 2013.

[75] R. Sanchez-Garcia, C. O. S. Sorzano, J. M. Carazo, and J. Segura, "BIPSPI: a method for the prediction of Partner-Specific Protein-Protein Interfaces," (in eng), *Bioinformatics,* Jul 2018, doi: 10.1093/bioinformatics/bty647.

[76] R. Esmaielbeiki, K. Krawczyk, B. Knapp, J. C. Nebel, and C. M. Deane, "Progress and challenges in predicting protein interfaces," (in eng), *Brief Bioinform,* vol. 17, no. 1, pp. 117-31, Jan 2016, doi: 10.1093/bib/bbv027.

[77] S. Malhotra, O. K. Mathew, and R. Sowdhamini, "DOCKSCORE: a webserver for ranking protein-protein docked poses," (in eng), *BMC Bioinformatics,* vol. 16, p. 127, Apr 2015, doi: 10.1186/s12859-015-0572-6.

[78] P. Chanphai, L. Bekale, and H. Tajmir-Riahi, "Effect of hydrophobicity on protein–protein interactions," *European Polymer Journal,* vol. 67, pp. 224-231, 2015.

[79] H. J. Dyson, P. E. Wright, and H. A. Scheraga, "The role of hydrophobic interactions in initiation and propagation of protein folding," (in eng), *Proc Natl Acad Sci U S A,* vol. 103, no. 35, pp. 13057-61, Aug 2006, doi: 10.1073/pnas.0605504103.

[80] L. S. Jasti, N. W. Fadnavis, U. Addepally, S. Daniels, S. Deokar, and S. Ponrathnam, "Comparison of polymer induced and solvent induced trypsin denaturation: the role of hydrophobicity," (in eng), *Colloids Surf B Biointerfaces,* vol. 116, pp. 201-5, Apr 2014, doi: 10.1016/j.colsurfb.2014.01.002.

[81] S. Kawashima, P. Pokarowski, M. Pokarowska, A. Kolinski, T. Katayama, and M. Kanehisa, "AAindex: amino acid index database, progress report 2008," (in eng), *Nucleic Acids Res,* vol. 36, no. Database issue, pp. D202-5, Jan 2008, doi: 10.1093/nar/gkm998.

[82] S. J. de Vries *et al.*, "HADDOCK versus HADDOCK: new features and performance of HADDOCK2.0 on the CAPRI targets," (in eng), *Proteins,* vol. 69, no. 4, pp. 726-33, Dec 2007, doi: 10.1002/prot.21723.

[83] A. Pintar, O. Carugo, and S. Pongor, "CX, an algorithm that identifies protruding atoms in proteins," (in eng), *Bioinformatics,* vol. 18, no. 7, pp. 980-4, Jul 2002.

[84] F. Towfic, C. Caragea, D. C. Gemperline, D. Dobbs, and V. Honavar, "Struct-NB: predicting protein-RNA binding sites using structural features," (in eng), *Int J Data Min Bioinform*, vol. 4, no. 1, pp. 21-43, 2010.

[85] M. Heinig and D. Frishman, "STRIDE: a web server for secondary structure assignment from known atomic coordinates of proteins," (in eng), *Nucleic Acids Res*, vol. 32, no. Web Server issue, pp. W500-2, Jul 2004, doi: 10.1093/nar/gkh429.

[86] C. Chothia, "The nature of the accessible and buried surfaces in proteins," (in eng), *J Mol Biol*, vol. 105, no. 1, pp. 1-12, Jul 1976.

[87] D. Chakravarty, J. Janin, C. H. Robert, and P. Chakrabarti, "Changes in protein structure at the interface accompanying complex formation," (in eng), *IUCrJ*, vol. 2, no. Pt 6, pp. 643-52, Nov 2015, doi: 10.1107/S2052252515015250.

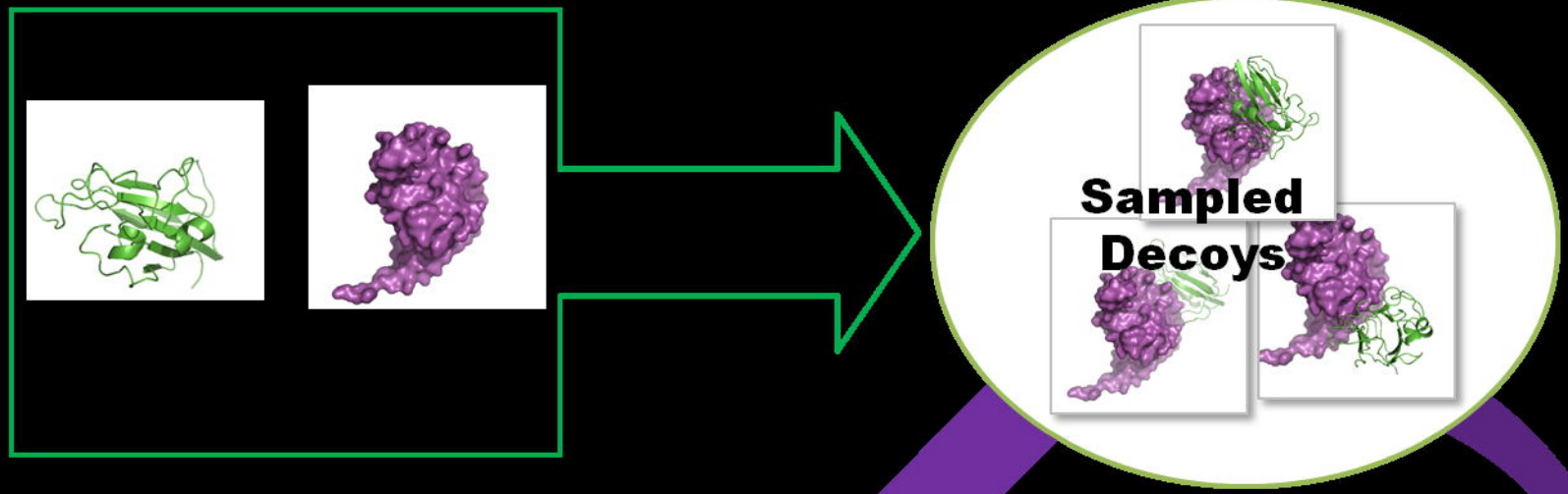
[88] J. Luo, L. Liu, S. Venkateswaran, Q. Song, and X. Zhou, "RPI-Bind: a structure-based method for accurate identification of RNA-protein binding sites," (in eng), *Sci Rep*, vol. 7, no. 1, p. 614, 04 2017, doi: 10.1038/s41598-017-00795-4.

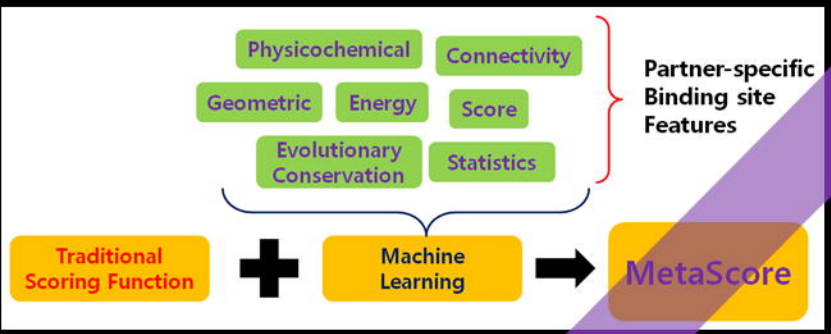
[89] S. Basu, D. Bhattacharyya, and R. Banerjee, "Mapping the distribution of packing topologies within protein interiors shows predominant preference for specific packing motifs," (in eng), *BMC Bioinformatics*, vol. 12, p. 195, May 2011, doi: 10.1186/1471-2105-12-195.

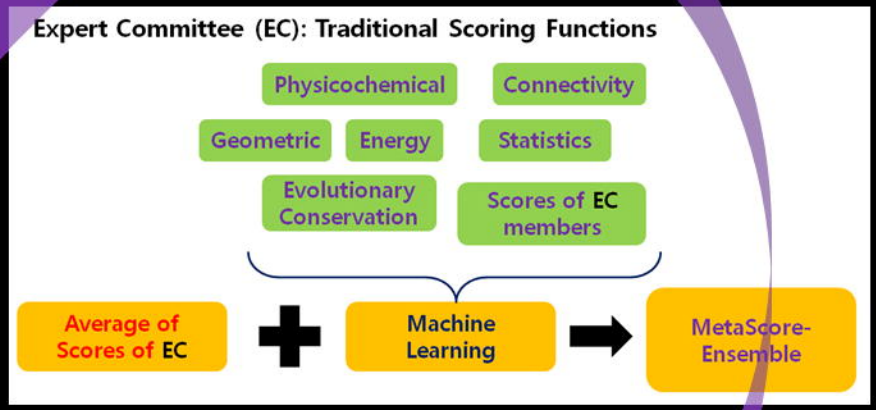
[90] P. Yang, Y. Hwa Yang, B. B Zhou, and A. Y Zomaya, "A review of ensemble methods in bioinformatics," *Current Bioinformatics*, vol. 5, no. 4, pp. 296-308, 2010.

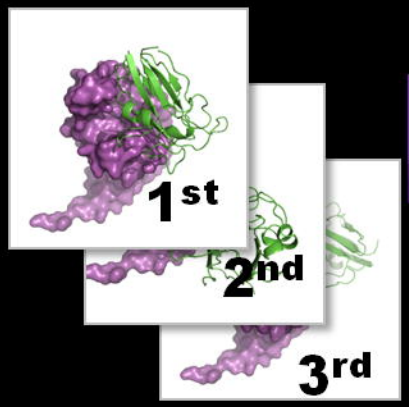
[91] L. Rokach, "Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography," *Computational Statistics & Data Analysis*, vol. 53, no. 12, pp. 4046-4072, 2009.

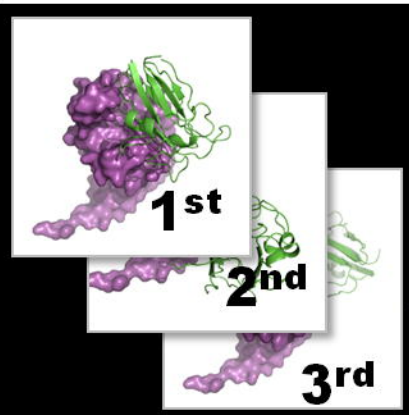
[92] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*, 2000: Springer, pp. 1-15.


Sampled
Decoys


Physicochemical
Connectivity
Geometric
Energy
Score
Evolutionary Conservation
Statistics
Partner-specific Binding site Features
Traditional Scoring Function
Machine Learning
MetaScore


Expert Committee (EC): Traditional Scoring Functions
Physicochemical
Connectivity
Geometric
Energy
Statistics
Evolutionary Conservation
Scores of EC members
Average of Scores of EC
Machine Learning
MetaScore-Ensemble


1st
2nd
3rd


1st
2nd
3rd