Briefings in Bioinformatics, 00(00), 2020, 1–18

doi: 10.1093/bib/bbz173
Problem solving protocol

Can machine learning consistently improve the scoring power of classical scoring functions? Insights into the role of machine learning in scoring functions

Chao Shen, Ye Hu, Zhe Wang, Xujun Zhang, Haiyang Zhong, Gaoang Wang, Xiaojun Yao, Lei Xu, Dongsheng Cao and Tingjun Hou

Corresponding authors. Tingjun Hou, Hangzhou Institute of Innovative Medicine, College of Pharmaceutical Sciences, Zhejiang University, Hangzhou 310058, Zhejiang, P. R. China. E-mail: tingjunhou@zju.edu.cn; Dongsheng Cao, Xiangya School of Pharmaceutical Sciences, Central South University, Changsha 410013, Hunan, P. R. China. E-mail: oriental-cds@163.com

Abstract

How to accurately estimate protein–ligand binding affinity remains a key challenge in computer-aided drug design (CADD). In many cases, it has been shown that the binding affinities predicted by classical scoring functions (SFs) cannot correlate well with experimentally measured biological activities. In the past few years, machine learning (ML)-based SFs have

Chao Shen is currently a PhD student in the College of Pharmaceutical Sciences, Zhejiang University, China. His research interests lie in the area of computer-aided drug design, including the development of structure-based virtual screening methodologies and the design of small molecule inhibitors of several important targets. He is mainly focusing on the development of AI-based scoring functions for protein-ligand docking.
Ye Hu received his PhD degree in 2011 from University of Bonn, Germany, and she is currently an independent researcher. Her current research interests include large-scale mining of ligand-target interaction data and structure-activity relationship analysis.
Zhe Wang is currently a postdoctoral fellow in the College of Pharmaceutical Sciences, Zhejiang University, China. His research interests lie in the area of computer-aided drug design, especially in virtual screening and free energy calculation. He is mainly focusing on the assessment and development of combined virtual screening strategies.
Xujun Zhang is now a postgraduate in the College of Pharmaceutical Sciences, Zhejiang University, China. He mainly engaged in the development of customized scoring functions based on artificial intelligence technologies.
Haiyang Zhong is currently a PhD student in the College of Pharmaceutical Sciences, Zhejiang University, China, and jointly cultivated by the College of Chemistry and Chemical Engineering, Lanzhou University, China. His research interests lie in area of computer-aided drug design, especially the design of small molecule inhibitors of several important targets.
Gaoang Wang is an undergraduate in the College of Pharmaceutical Sciences, Zhejiang University, China. He mainly devotes himself to the development of high-precision scoring function based on artificial intelligence technologies.
Xiaojun Yao is a professor in the State Key Laboratory of Quality Research in Chinese Medicine, Macau Institute for Applied Research in Medicine and Health, Macau University of Science and Technology, China, and the College of Chemistry and Chemical Engineering, Lanzhou University, China. His research interests lie in area of computer-aided drug design and molecular modeling.
Lei Xu received his PhD degree in 2013 from the Institute of Functional Nano & Soft Materials, Soochow University, Suzhou, China, and now he is an associate professor in the Institute of Bioinformatics and Medical Engineering, Jiangsu University of Technology, Changzhou, China. His research interests lie on the methodology development and application of computer aided drug design (CADD) and design and discovery of novel drug candidates for important targets.
Tingjun Hou received his PhD degree in 2002 from Peking University, China. He is currently a professor in the Hangzhou Institute of Innovative Medicine, College of Pharmaceutical Sciences, Zhejiang University, China. His research focuses on molecular modeling and computer-aided drug design (CADD), including development of structure-based virtual screening methodologies, theoretical predictions of ADMET and drug-likeness, discovery of small molecule inhibitors toward important drug targets and multiscale simulations of target-ligand recognition. More information can be found at the website of his group: http://cadd.zju.edu.cn.
Dongsheng Cao received his PhD degree in 2013 from Central South University, China. He is currently a professor in the Xiangya School of Pharmaceutical Sciences, Central South University, China. His research interests include (i) artificial intelligent systems for drug discovery and disease diagnosis, (ii) the development of software, web service and database in systems biology and drug discovery and (iii) design and discovery of small molecule inhibitors of important protein targets. More information can be found at the website of his group: http://www.scbdd.com.
Submitted: 1 October 2019; **Received (in revised form):** 10 December 2019

gradually emerged as potential alternatives and outperformed classical SFs in a series of studies. In this study, to better recognize the potential of classical SFs, we have conducted a comparative assessment of 25 commonly used SFs. Accordingly, the scoring power was systematically estimated by using the state-of-the-art ML methods that replaced the original multiple linear regression method to refit individual energy terms. The results show that the newly-developed ML-based SFs consistently performed better than classical ones. In particular, gradient boosting decision tree (GBDT) and random forest (RF) achieved the best predictions in most cases. The newly-developed ML-based SFs were also tested on another benchmark modified from PDBbind v2007, and the impacts of structural and sequence similarities were evaluated. The results indicated that the superiority of the ML-based SFs could be fully guaranteed when sufficient similar targets were contained in the training set. Moreover, the effect of the combinations of features from multiple SFs was explored, and the results indicated that combining NNscore2.0 with one to four other classical SFs could yield the best scoring power. However, it was not applicable to derive a generic target-specific SF or SF combination.

Key words: scoring function (SF); machine learning (ML); scoring power; binding affinity; ML-based SF

Introduction

How to accurately evaluate protein–ligand binding affinity remains a key challenge in the field of computational biology and computational chemistry. The performance of a wide range of campaigns involved in computer-aided drug design (CADD), including structure-based virtual screening (SBVS), lead optimization and drug repurposing, can be hardly guaranteed without the reliability of scoring functions (SFs) [1–3]. In the past few decades, extensive efforts have been made to develop novel SFs or improve the existing ones. However, there is still a long way to go to further develop effective SFs that could best mimic the real physiological scenario [4].

Classical SFs can be typically divided into three main categories: force field-based, knowledge-based and empirical [5]. Force field-based SFs can be usually described as the sum of two types of energies generated from a force field, namely protein–ligand interaction energy and internal ligand energy, and the former can be further divided into several non-bonded interaction terms such as van der Waals and electrostatic terms. But actually, some tough but important terms such as desolvation and entropic contributions are hardly considered or just simplified. Examples of force field-based SFs include DOCK [6], GoldScore [7] and LigandFit [8]. Empirical SFs adopt similar functional forms as force-field methods but introduce addition empirical terms to represent the protein–ligand interactions, and the weight of each term can be obtained by linearly fitting them to experimentally-determined binding affinities. Notable examples are Autodock Vina [9], X-Score [10], GlideScore [11, 12] and ChemScore [13]. Knowledge-based SFs are derived from statistical analysis of known protein structures, and a typical example is using the occurrence frequency of atom–atom pairwise distances to characterize the complexes. Popular implementations of knowledge-based SFs include PMF [14] and DrugScore [15]. Besides, some SFs are also developed to incorporate the advantages of multiple types of SFs and use a mixture of descriptors, such as SMoG2016 (empirical and knowledge-based) [16] and GalaxyDock-BP2-score (force-field based, empirical and knowledge-based) [17].

A common feature of classical SFs is that they usually assume a predetermined additive functional form representing the relationship between experimentally determined binding affinities and the features that characterize protein–ligand complexes. However, in reality, such linear correlation may not always exist [19, 20]. Rather than the predetermined functional form, ML-based SFs can automatically learn both generalized nonlinear functional forms and feature information from training data. Thus, they have gradually emerged as potential alternatives and shown higher flexibility and expressiveness than classical ones in terms of scoring power (i.e. the capability to rank compounds by binding affinities) [21–28], docking power (the capacity to distinguish near-native poses from decoys) [29, 30] and/or screening power (the capacity to discriminate active compounds from decoys) [31–36]. During the last few years, significant progress on ML-based SFs has been made. Especially, a wide range of ML algorithms, including random forest (RF) [21–23, 30, 31, 36], support vector machine (SVM) [25, 32, 33, 37, 38], artificial neural network [39–43], gradient boosting decision tree (GBDT) [26, 44–47] and convolutional neural network (CNN) [27–29, 34, 35, 48], have been applied and greatly contributed to the development of SFs.

Energy terms from classical SFs are considered as an important source of features to develop novel SFs. For example, based on the SFCscore descriptors, Zilian and Sotriffer [24] proposed a method named SFCscore^{RF} by using RF to replace the original linear fitting method. Other well-known examples include RFScore-v3 [23], NNscore2.0 [40], BgN(BsN)-Score [43] and $\Delta_{vina}RF_{20}$ [30], where energy terms from classical SFs such as Vina, X-Score and Smina are to a certain extent incorporated as features. Ballesters group developed a series of RFScore-related approaches, by which they found that the combinations of RFScore and other SFs such as Vina and Cyscore, the use of simple elemental atom types rather than more complicated sybyl atom types and structural interaction fingerprints (SIFts) generated from CREDO, and the replacement of eXtreme Gradient Boosting (XGBoost) to RF, can effectively improve the scoring power [21–23, 49]. However, the features they used were only limited to RFScore and some generated from academic software. In addition, the corresponding ML methods were also mainly confined to RF and original multiple linear regression (MLR). In 2015, Ashtawy and Mahapatra [50] conducted a comprehensive comparative assessment of the scoring power of classical and ML-based SFs for the prediction of protein–ligand binding affinity. They developed several ML-based SFs by using six ML algorithms based on a variety of descriptors calculated by X-Score, AffiScore and RFScore. Their assessment was carried out from multiple aspects, including the performance on a diverse test set, the performance on homogeneous test sets and novel targets, and even the impacts of training set size and feature selection. However, only the features from the above mentioned three SFs were explored.

In this study, to better recognize the potential of classical SFs, a comparative assessment in terms of scoring power was conducted toward 25 commonly-used SFs, and their individual

energy terms were computed by each original SF to construct new ML-based SFs. Based on our analysis, we attempt to solve the following puzzles: (1) For a single SF, can ML methods consistently improve the scoring power by replacing original MLR and which ML approach would perform best? (2) Existing well-performed ML-based SFs such as RFScore and NNscore possess their specifically designed features that are mainly composed of atom type pair counts. Can our newly-developed ML-based SFs featured by energy terms from classical SFs outperform those from existing ML-based methods? (3) Structural and sequence similarities between two proteins have been proven to exert a significant impact on the performance of ML-based methods [51]. How would our newly-constructed SFs perform as the structural or sequence similarity between training and test sets changes? (4) The combination of different features has been reported as an effective strategy to improve the final performance [23]. How would it behave when energy terms from different SFs are combined? (5) The scoring power tested on diverse test sets and homogeneous test sets usually shows large difference [50]. How would our newly-developed ML-SFs perform on some representative targets?

Materials and methods

Data sets

The PDBbind [52] database provides a consolidated repository of the bioactivity data for biomolecular complexes extracted from the Protein Data Bank (PDB) [53], and is considered as a standard benchmark widely employed for SF development. It is annually updated with the newest version of v2018 that contains a total of 16 151 protein–ligand complexes. Taking v2007 as an example, 1300 complexes with high-resolution X-ray crystal structures and high-quality experimentally determined binding affinity data are selected as the 'refined set' [54]. This set is further clustered into 65 clusters at 90% sequence similarity. Then three complexes, if available, with the highest, median and lowest binding affinity extracted from each cluster are chosen to form the 'core' set, resulting in a total of 195 diverse complexes. As the compositions of the refined and core sets of each version might vary over different versions, three most commonly used versions, i.e. v2007 [54] (containing 1300 and 195 complexes in the refined and core sets, respectively), v2013 [55] (containing 2959 and 195 complexes in the refined and core sets, respectively) and v2016 [56] (containing 4057 and 290 complexes in the refined and core sets, respectively), were used for our analysis. In all benchmark calculations, the core set was used as the test set, whereas the refined set excluding the complexes from the test set was used as the training set.

To further explore the impacts of structural and sequence similarities between proteins from the training and test sets on the overall performance, another data set (dataset II) proposed by Li *et al.* [51] was also utilized. This set was modified from PDBbind v2007, from which the pairwise structural similarity and sequence identity between each protein in the training set and each protein in the test set were calculated by TM-Score [57] and NW-align [58], respectively. To make a better comparison with the results from different settings, the test set remains unchanged, while the training set was split into several subsets based on predefined similarity cutoffs. In total, 26 structural similarity cutoffs ranging from 0.4 to 1.0 and 28 sequence identity cutoffs ranging from 0.3 to 1.0 were applied, as reported in the Supplementary Table S1. Subsequently, four groups of data sets containing 106 training subsets were obtained. Among all four groups, the two groups gradually remove increasingly similar proteins from the training sets, whereas the other two eliminate increasingly dissimilar ones.

All resulting complex structures were further standardized using the Protein Preparation Wizard [59] module in Schrödinger 2018, including removing waters and redundant chains, assigning bond orders, adding hydrogen atoms, filling in missing side chains and optimizing H-bond network. To retain the original structures, only hydrogen atoms were minimized with the OPLS2005 force field [60]. The protonation states of residues at pH = 7.0 were determined by PROPKA [61]. The most suitable ionized state or tautomer of each ligand at pH = 7.0 was generated by using Epik [62]. Metal ions were retained at first because some SFs could recognize the interactions with them. However, they were removed later if they could not be recognized by any SF. Besides, if the assessed SF or docking program based on the given SF had its intrinsic protein preparation function, then such built-in function would be used.

Scoring functions

In this study, 25 author-available and term-decomposable SFs were tested, as listed in Table 1 and further detailed in Supplementary Material. To avoid possibly large difference to the ligand conformation, all the calculations were performed without local minimization. If the scores of some complexes could not be calculated by a given SF or the scores significantly differed from the normal predicted values, these complexes were removed for this SF. The numbers of the complexes in each group of training and test sets for the construction of each SF are listed in Supplementary Table S3. When different SFs were combined, their common complexes were used. All the other parameters were set without tuning of the optional parameters, unless otherwise noted as followed.

Glide [11, 12]

The receptor grid generation utility of glide was used to generate the receptor grid, and the binding box was defined with the size of $10 \times 10 \times 10$ Å centered on the co-crystallized ligand. Then, with the selection of the score in place only option, glide scoring calculations with standard precision (SP) and extra precision (XP) were carried out.

GOLD [7]

Proteins were prepared by the built-in protein preparation module including adding hydrogen atoms and deleting unnecessary waters. Then, the four SFs implemented in GOLD were used for scoring, including piecewise linear potential (ChemPLP), GoldScore, ChemScore and Astex Statistical Potential (ASP).

PLANTS [63]

Three built-in SFs, namely ChemPLP, PLP and PLP95, were employed for scoring by setting rescore_mode to no simplex.

AutoDock [18]

Proteins and ligands were firstly converted into the pdbqt format by mgltools 1.5.6, along with the addition of hydrogen atoms, the assignment of Gasteiger charges, and cleanup of unwanted elements. Then, the compute_AutoDock41_score.py script in mgltools was used for scoring.

Table 1. Basic information of the SFs assessed

SF	Software	Classification	Number of features	SF	Software	Classification	Number of features
GlideScore-SP [7]	Schrodinger (v2018)	Empirical	8	Affinity-dG	MOE [10] (v2018)	Empirical	4
GlideScore-XP [8]		Empirical	15	Alpha-HB		Empirical	3
ChemPLP	GOLD [9] (v 5.7.1)	Empirical	10	GBVIWSA-dG		Force field	4
GoldScore		Force field	5	London-dG		Empirical	11
ChemScore		Empirical	9	AffiScore [58]	Slide (v3.4)	Empirical	10
ASP		Knowledge empirical	4	X-Score [56]	X-Score (v1.3)	Empirical	6
ChemPLP	PLANTS [53] (v1.2)	Empirical	16	Cyscore	Cyscore (v2.0.1)	Empirical	4
PLP		Empirical	19	SMoG2016 [60]	SMoG2016	knowledge, empirical	4
PLP95		Empirical	26	GalaxyDock-BP2-score [57]	GalaxyDock-BP2-score	Empirical	10
AutoDock [6]	AutoDock (v4.1)	Empirical, force field	5	RFScore-v2::elem	RF-Score-v2 [13]	ML	486
Vina [54]	Vina (v1.1.2)	Empirical	6	RFscore-v2::credo		ML	72
Smina [55]	Smina	Empirical	58	NNscore1.0 [29]	NNScore1.0	ML	194
				NNscore2.0 [30]	NNScore2.0	ML	348

Vina [9] and Smina [64]

All proteins and ligands were preprocessed in the same way as applied in AutoDock. The calculations were performed with the score_only option. As for Smina, all 58 available energy terms were extracted.

MOE [65]

Proteins were preprocessed using the built-in structure preparation module. Four SFs available in MOE, i.e. Affinity-dG, Alpha-HB, GBVIWSA-dG and London-dG, were utilized for scoring (another SF ASE could not be decomposable). The total scores were further decomposed into individual energy terms using the in-house scientific vector language scripts.

X-Score [10]

Proteins and ligands were prepared with the FixPDB and FixMol2 options. Three SFs that utilize different approaches to predict hydrophobic interactions were applied, including X-Score_HP, X-Score_HM and X-Score_HS. In addition to the individual calculation of these three SFs, their average values were also computed as X-Score_average. Thus, a total of four SFs were used.

GalaxyDock-BP2-score [17], Slide [66], Cyscore [67] and SMoG2016 [16]

The energy terms in these four SFs were directly generated with the built-in calc_energy.py, slide_score, Cyscore and SMoG2016.exe utilities, respectively.

RFScore [22]

The second version of RFScore uses the counts of atom type pairs within a predefined distance to represent protein–ligand complexes. Two atom type schemes were utilized herein, i.e. Element and credo, which applied element symbols of the interacting atoms and SIFts [68] to encode protein–ligand interactions, respectively. Distance cutoff was set to 5.0 Å with binsize of 1.0 Å. Thus, the atom type pairs within 5.0 Å around the ligand were counted every 1 Å, leading to 486 and 72 features for RFScore-v2::elem and RFscore-v2::credo, respectively.

NNscore [39, 40]

As described in AutoDock, Vina and Smina, the pdbqt formats were prerequisites for proteins and ligands. Then, the features characterizing protein–ligand complexes were directly obtained by modifying the source code of NNscore. NNscore1.0 contains a total of 194 features that are mostly knowledge-based and used to represent the close contacts, semi-close contacts, electrostatic-interaction energy, number of ligand atom types and number of ligand rotatable bonds. Compared with the first version, NNscore2.0 not only expands the basic atom types but also incorporates SIFts and the energy terms from Vina, resulting in a total of 348 features.

Machine learning algorithms

A total of eight popular ML methods, including RF [69], ExtraTrees [70], GBDT [71], XGBoost [72], support vector regression (SVR) [73], *k*-nearest neighbor (*k*NN) [74], deep neural network (DNN) [75] and MLR were applied for the generation of regression models. XGBoost and DNN were executed with the xgboost and

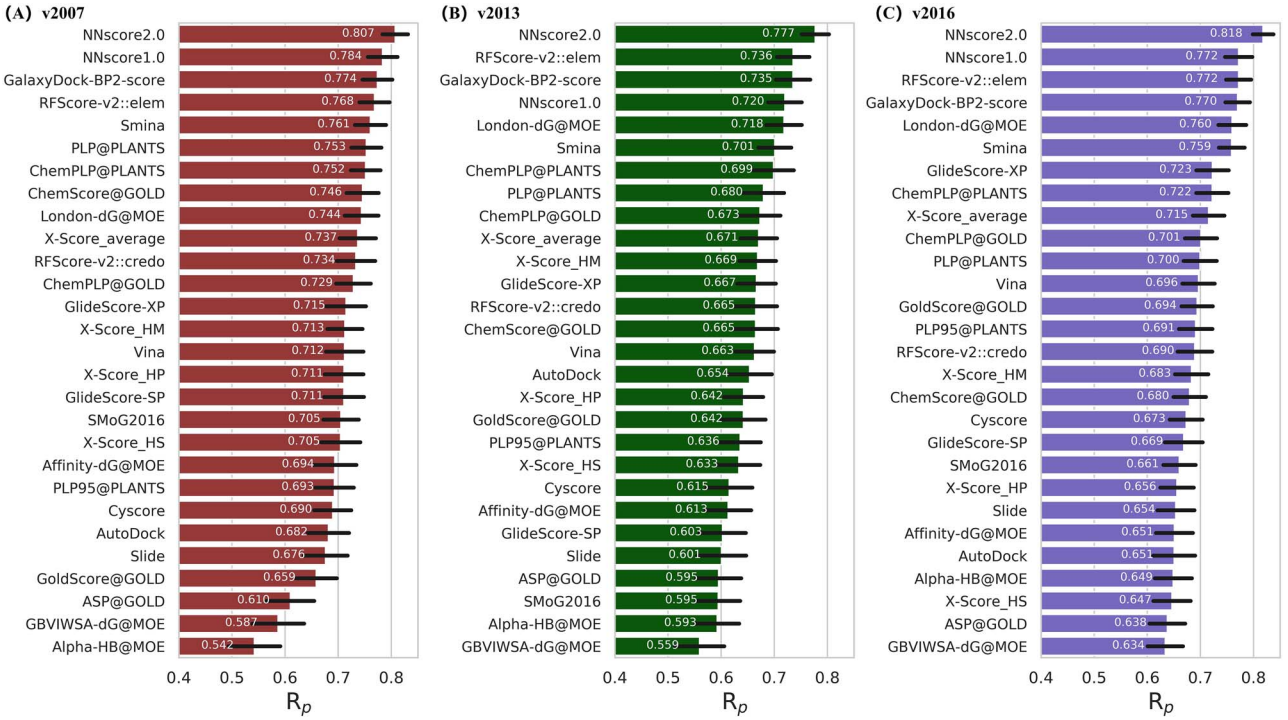

Figure 2. Comparison of the scoring powers (R_ps) of several ML-based SFs toward the core sets of (A) PDBbind v2007, (B) PDBbind v2013 and (C) PDBbind v2016. Only the best-performed ML-based SF toward a certain set of features is retained. The SFs are ranked in descending order of the scoring power.

keras [76] modules in python, respectively, whereas the others were conducted by using the scikit-learn [77] package. As the training of DNN is too time-consuming and its performance is not significantly better than the others, it was only used in the construction of the single SFs. The features extracted from each SF were firstly standardized with the sklearn.preprocessing module, as the performance of several ML methods such as MLR, *k*NN and SVR largely rely on the normalization of input data. To ensure the integrity of each SF, no additional preprocessing step was taken to the input data. However, when energy terms were combined as features, two filtering criteria were applied to avoid the influence of redundant features and accelerate the training process. First, the descriptors that had all zero values or zero variance were removed. Second, when the correlation between two descriptors is higher than the predefined threshold (0.90), the first one was remained, and the other was discarded.

The Bayesian optimization algorithm (BOA) has been proven to be an effective approach to perform hyperparameter tuning [78]. Unlike random search, it can incorporate prior beliefs about solutions to the problem. In particular, suitable priors can significantly improve the search speed by directing to the most likely configurations. In this study, the hyperparameters of each model were automatically tuned with BOA implemented in the hyeropt [79] module. Five-fold cross-validation was utilized to evaluate different hyperparameter combinations, and root-mean-squared error (RMSE) was employed as the object function. Tree Parzen estimator [80] was used as the optimization algorithm, and the maximal iteration was set to 500 (as for DNN, the maximal iteration was set to 50 because it was too time-consuming and only the coarse-tuning of the hyperparameter was conducted). The brief description of each ML approach and the corresponding hyperparameters are summarized in Supplementary Table S4.

Evaluation metrics

Three commonly used metrics for model evaluation were calculated to estimate the scoring power of each constructed SF, i.e. the Pearson correlation coefficient (R_p), the Spearman correlation coefficient (R_s) and the RMSE between the predicted and experimental binding affinities. Scores with higher R_p and R_s and those lower RMSE indicate better scoring power. The bootstrap sampling method was used to further evaluate each newly-constructed SF. To be specific, random sampling of 1000 redundant copies with replacements was conducted on the test set, where each copy was in the same size as the original test set. Then, its performance was re-evaluated on each redundant copy of the test set in terms of R_p, R_s or RMSE. Finally, a total of 1000 bootstrap samples were obtained for each performance metric, and the average score over all calculations was then used as the final result for each complex. The *post hoc* Nemernyi test [81] implemented in the scikit-posthocs [82] package was chosen to determine if the difference between any two of the compared SFs was statistically significant. If the computed *P*-value was higher than 0.10, the performances of two tested methods were considered to have no significant difference. Besides, in most instances, it was observed that R_p, R_s and RMSE showed comparable trends, and therefore R_p was considered as the major metric and discussed in the following. However, the results using the other two metrics were also reported in the Supplementary Materials.

Results and discussion

Comparison of scoring powers on PDBbind core sets

Each newly-developed ML-based SF was firstly trained on the training sets of PDBbind v2007, v2013 and v2016, and then tested on their corresponding test sets. The scores predicted by their

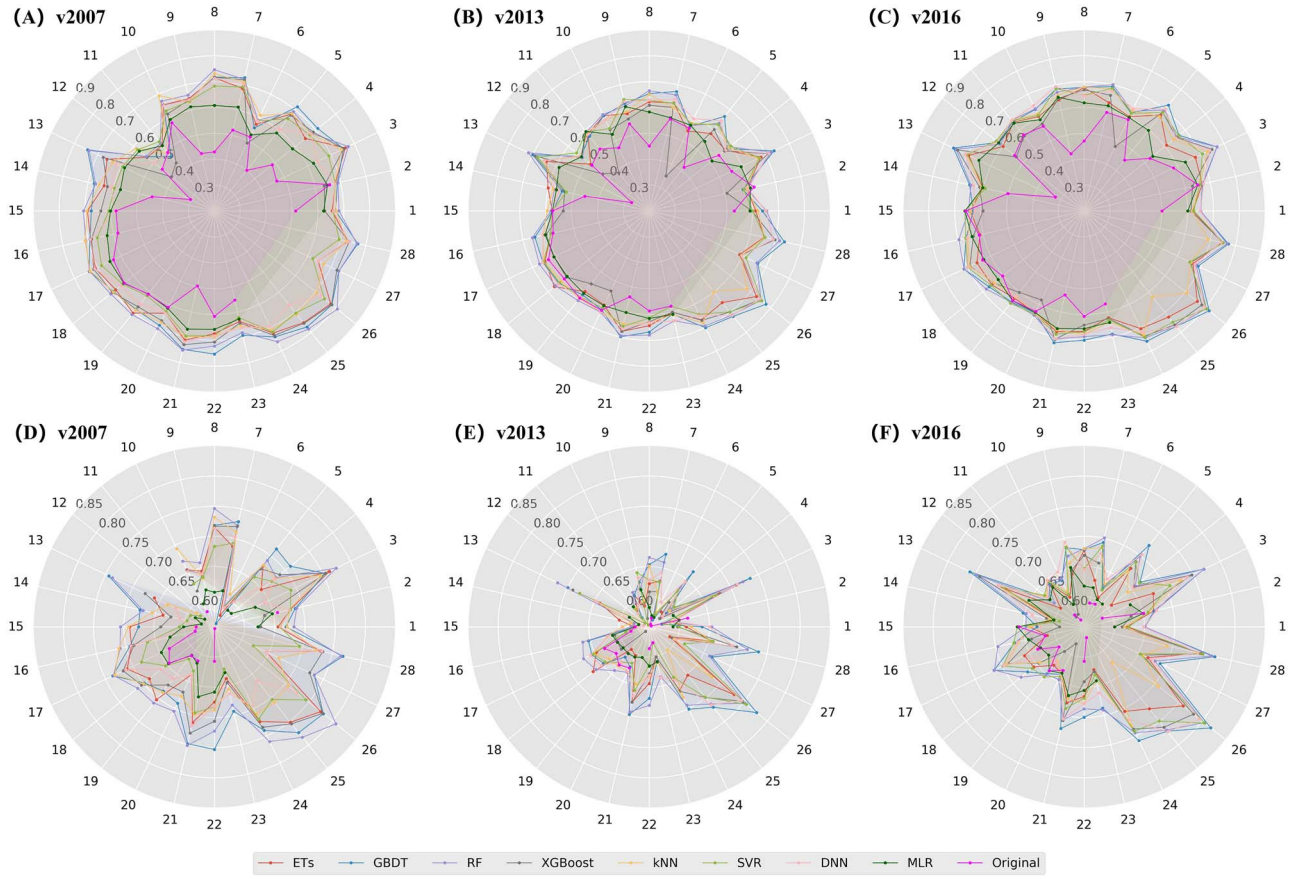

Figure 1. Comparison of the scoring powers (R_ps) of several ML-based SFs based on different ML approaches toward the core sets of (A, D) PDBbind v2007, (B, E) v2013 and (C, F) v2016. (For labels: 1 — AutoDock; 2 — Cyscore; 3 — GalaxyDock-BP2-score; 4 — GlideScore-SP; 5 — GlideScore-XP; 6 — ASP@GOLD; 7 — ChemPLP@GOLD; 8 — ChemScore@GOLD; 9 — GoldScore@GOLD; 10 — Affinity-dG@MOE; 11 — Alpha-HB@MOE; 12 — GBVIWSA-dG@MOE; 13 — London-dG@MOE; 14 — Slide; 15 — SMoG2016; 16 — Vina; 17 — X-Score_average; 18 — X-Score_HM; 19 — X-Score_HP; 20 — X-Score_HS; 21 — ChemPLP@PLANTS; 22 — PLP@PLANTS; 23 — PLP95@PLANTS; 24 — Smina; 25 — NNscore1.0; 26 — NNscore2.0; 27 — RFScore-v2::credo; 28 — RFScore-v2::elem). Standard deviations are not shown in the figure because they are too dense to be displayed clearly.

original classical SFs and the results re-fitted by MLR were utilized as the references. No reference was considered for Smina, RFScore and NNscore, since on the one hand, only the features were extracted from these three SFs, and on the other hand, MLR could not handle the matrices with too many sparse features. The resulting R_p, R_s and RMSE based on the 5-fold cross-validation for the training sets are illustrated using radar charts, as shown in Supplementary Figures S1–S3, respectively. Similarly, the results for the core sets are shown in Figures 1, S4 and S5. The *P*-values of any two compared ML methods toward a certain set of features are shown as the heatmap plots in Figures S6–S14.

As for R_p being the major metric, different ML-based SFs perform similarly for the training sets assembled from different versions of PDBbind (Figure S1). In contrast, for the test sets (Figure 1), the performance of v2013 is slightly worse than those of the other two versions. In general, the test sets show comparable results as the training sets, suggesting that the developed ML-based SFs could yield stable predictions for the test sets.

In Figure 1, the majority of ML-based SFs perform better than the original SFs by MLR, as exemplified by a number of empirical SFs including GalaxyDock-BP2-score, GlideScore-SP, GlideScore-XP, ChemPLP@GOLD, ChemScore@GOLD, London-dG@MOE and ChemPLP@PLANTS. For these SFs with relatively more features available, significant improvements are observed when advanced ML methods are applied. A striking example is found for London-dG@MOE that obtains the lowest R_p (0.323) for v2016 using the original MLR. However, the performance improves dramatically when other ML technologies are involved (Figure 1C). In addition, two force field-based SFs (i.e. AutoDock and GoldScore@GOLD) also show clear improvements, despite that only five features are employed. Such observation indicates that the parameters mainly determined by force field could be effectively re-fitted by other ML methods, reflecting that these features are more likely to display a non-linear relationship. However, the application of more advanced ML methods could not ensure considerable increase of the predictive power for several SFs, such as ASP@GOLD, SMoG2016, Alpha-HB@MOE and GBVIWISA-dG@MOE and four SFs of X-Score. In these cases, only marginal improvements are observed comparing with MLR (Figure 1C). One main reason could be that too few features are utilized in these SFs, so that little valid information could be learned from the training samples. As for the first two partially knowledge-based SFs (ASP@GOLD and SMoG2016), their poor performance might also be attributed to the difficulty with extracting individual knowledge-based terms that represent all possible pairwise atomic interactions. These inter-atomic terms can hardly be directly described for different proteins. In contrast, RFScore and

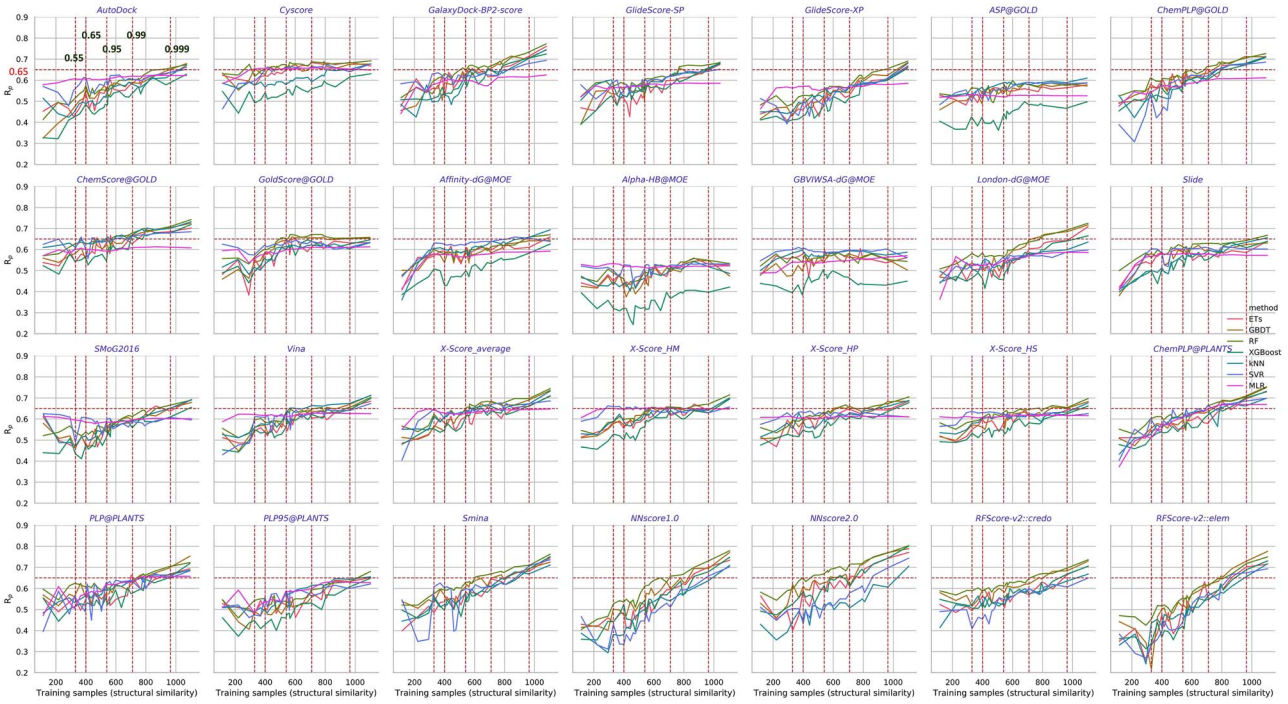

Figure 3. Test set performance (R_ps) of several ML-based SFs with increasingly similar training samples based on the structural similarity cutoff. Structural similarities of 0.55, 0.65, 0.95, 0.99 and 0.999 are marked in the figure. Standard deviations are not shown in the figure because they are too dense to be displayed clearly.

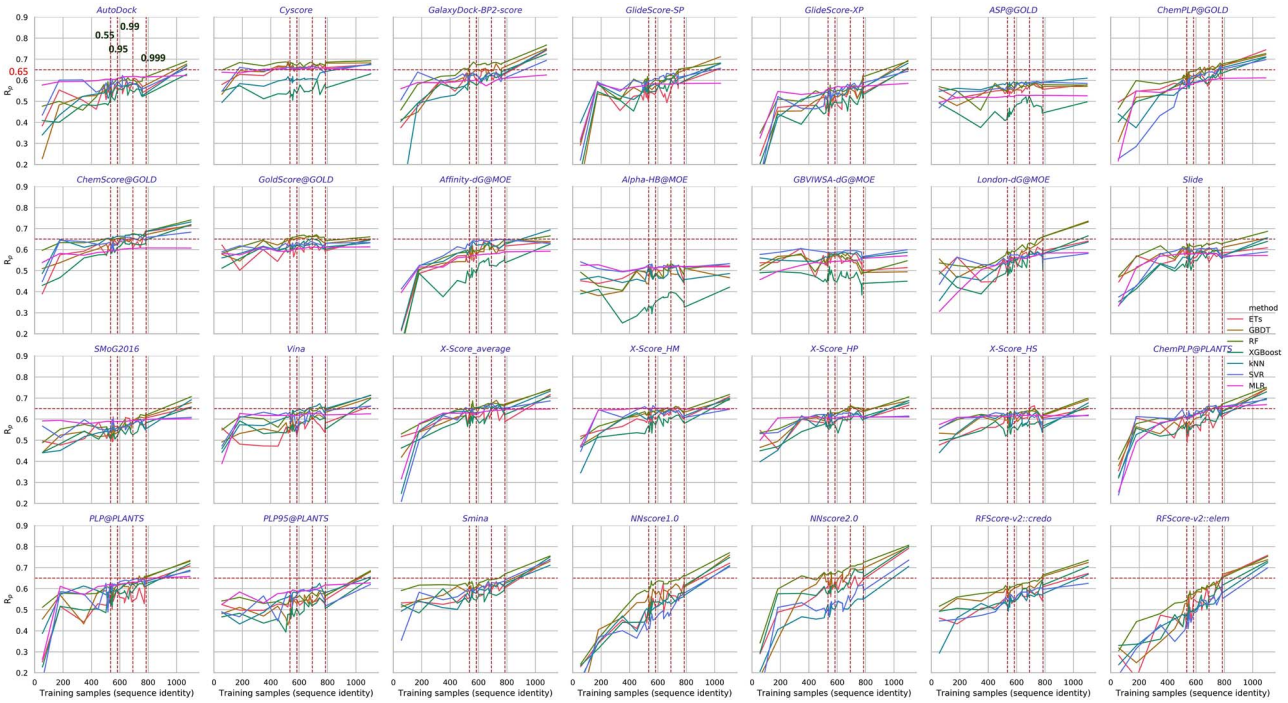

Figure 4. Test set performance (R_ps) of several ML-based SFs with increasingly similar training samples based on the sequence identity cutoff. Sequence identities of 0.55, 0.95, 0.99 and 0.999 are marked in the figure. Standard deviations are not shown in the figure because they are too dense to be displayed clearly.

NNscore represent atomic interactions by the form of simple atomic pair counts, thus leading to their evidently superior performance.

When comparing the performance of different ML approaches, four ensemble learning methods, especially GBDT and RF, are found obtaining the best scoring power for most of the SFs (Figure 1C). However, the difference between these ensemble ML methods and the others (i.e. SVR, DNN and kNN) is relatively small. For inherent ML-based SFs (i.e. NNscore and RFScore), relatively simple kNN performs significantly worse than the others, whereas ensemble ML methods, SVR and DNN could gain more improvements when sufficient features are

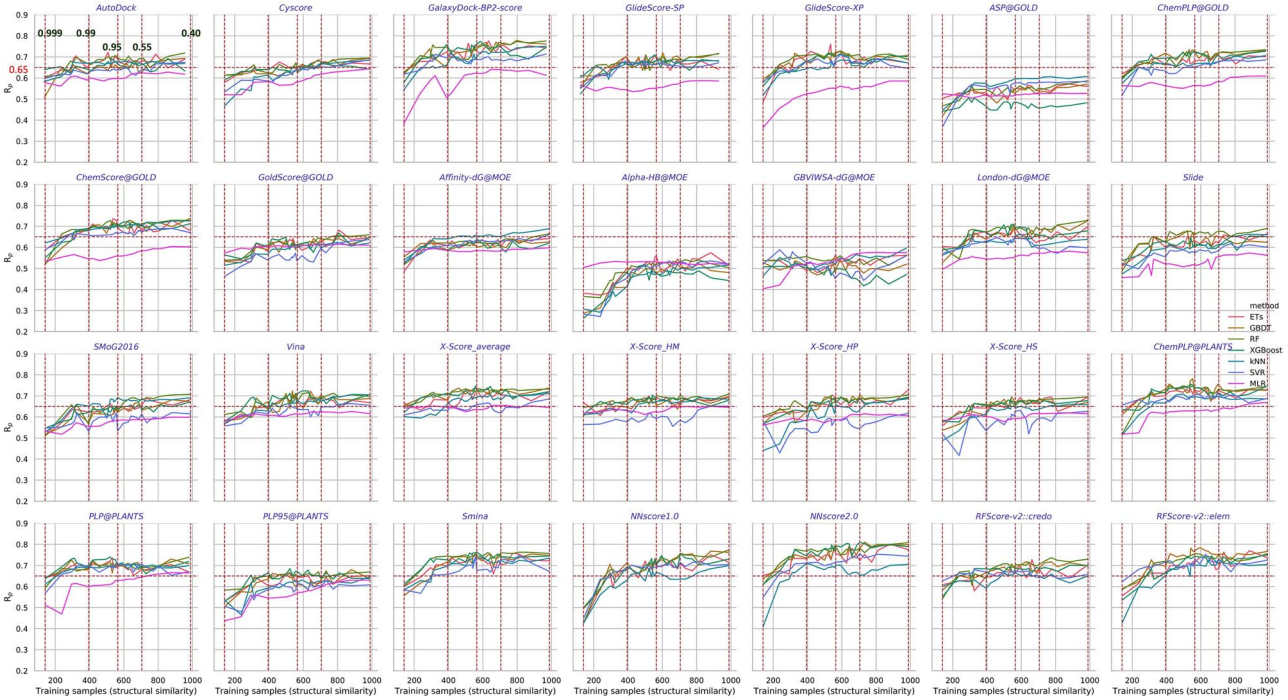

Figure 5. Test set performance (R_ps) of several ML-based SFs with increasingly dissimilar training samples based on the structural similarity cutoff. Structural similarities of 0.40, 0.55, 0.95, 0.99 and 0.999 are marked in the figure. Standard deviations are not shown in the figure because they are too dense to be displayed clearly.

included. When it comes to DNN, it is regretful to find that the improvement of the predictive power cannot well offset its high computational cost. Thus, when constructing a SF just based on energy terms, pure DNN may not be highly recommended.

Next, the best R_p values were collected for individual SFs in order to further compare the general scoring power of the 28 newly-constructed ML-based SFs, as reported in Figure 2. Accordingly, the R_s and RMSE values are provided in Supplementary Figures S15 and S16, respectively, and the corresponding *P*-values between any two of the compared SFs are shown in Figures S17–S19. Taking PDBbind v2016 as an example, the best R_p scores for 28 SFs range from 0.634 to 0.818. Among them, six SFs result in R_p values of higher than 0.75, i.e. NNscore2.0, NNscore1.0, RFScore-v2::elem, GalaxyDock-BP2-score, London-dG@MOE and Smina in descending order. Existing well-performed ML-based SFs, such as RFScore and NNscore, display the highest scoring power, which can be expected. However, GalaxyDock-BP2-score, a hybrid empirical SF that extracts multiple components from other classical SFs, i.e. AutoDock, PLP, X-Score and DrugScore [83], performs surprisingly well. This might suggest that the integration of multiple types of SFs could improve the scoring power. In addition, these six SFs are found with more features available, i.e. 348, 194, 486, 10, 11 and 58, respectively. The majority of SFs (18) yield R_p values between 0.65 and 0.75. Four SFs show relatively worse performance with R_p values lower than 0.65, i.e. Alpha-HB@MOE, Affinity-dG@MOE, GBVIWSA-dG@MOE and ASP@GOLD, which consist of 3, 4, 4 and 4 features, respectively. Therefore, SFs with fewer features tend to display worse predictive power. The performance is also shown in descending order of feature numbers (Figures S20–S22), which further validates our viewpoints mentioned above.

Impacts of structural and sequence similarity on scoring power

In any ML calculation, how to optimize the composition of the training and test sets remains a puzzle. Using the standard training and test sets, as described above, has proven that the presence of high structural overlap between the training and test sets might over-estimate the performance of ML-based SFs [49, 51, 84–86]. Thus, to further explore the impacts of structural and sequence similarity on the scoring power of the newly-developed SFs, dataset II firstly proposed by Li *et al.* [51] was also employed in this study. On the basis of 28 predefined similarity/identity cutoffs, four groups of training subsets with varying sizes were obtained, as listed in Supplementary Table S1. Accordingly, the test set performance of all SFs for individual groups assessed by R_p, R_s and RMSE is reported in Figures 3–6, S23–S26 and S27–S30, respectively.

Overall, with the increase of the structural or sequence similarity, the performance of most ML-based SFs increases gradually, whereas that of classical SFs remains almost unchanged (Figures 3 and 4). This suggests that structural and sequence similarity between the proteins in the training and test sets can surely exert a notable influence on the final performance, and the performance of ML-based SFs is indeed over-estimated to some extent. However, in most cases, ML-based SFs can still outperform classical ones because they can keep learning with the addition of increasingly similar training data. It is likely that ML-based SFs cannot give reliable predictions for completely novel targets, but the prediction accuracy of ML-based SFs can still be guaranteed when enough diverse proteins are included in the training set. As for individual SFs, the impacts of structural similarity and sequence identity are different. For most ML-based SFs, nearly 800 training samples are needed to achieve $R_p \geq 0.65$, which means that the structural similarity

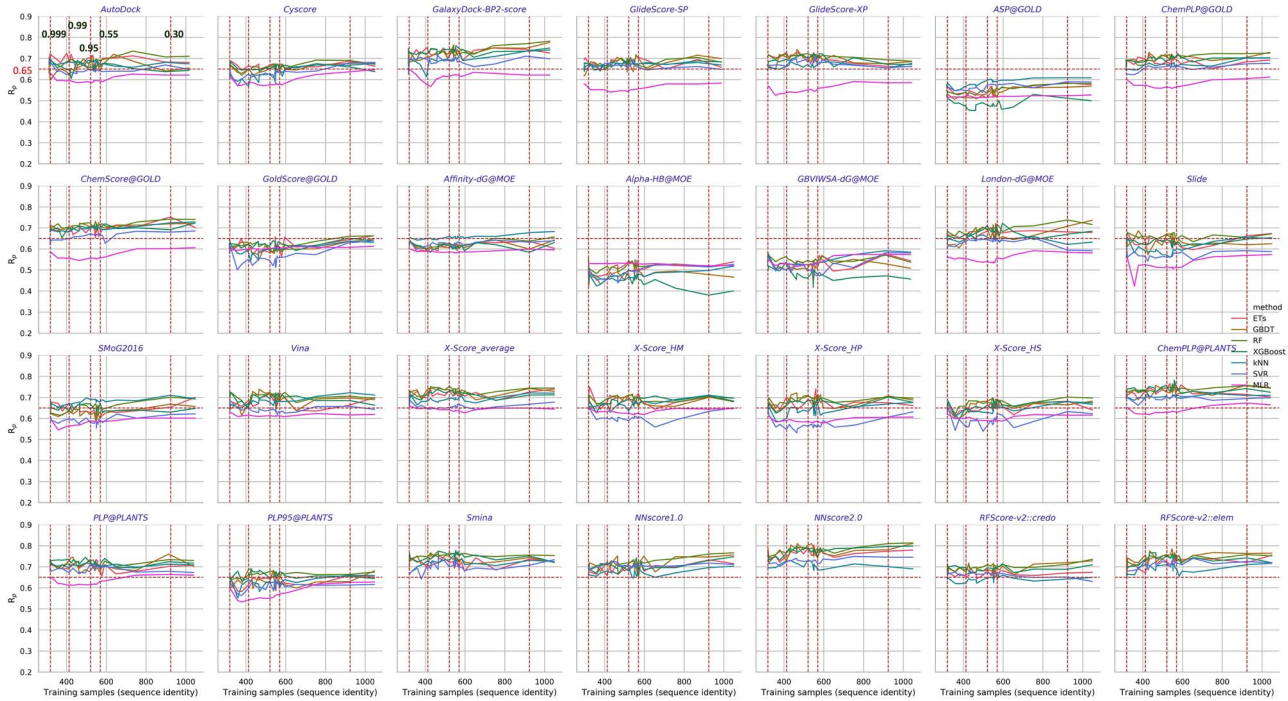

Figure 6. Test set performance (R_ps) of several ML-based SFs with increasingly dissimilar training samples based on the sequence identity cutoff. Sequence identities of 0.30, 0.55, 0.95, 0.99 and 0.999 are marked in the figure. Standard deviations are not shown in the figure because they are too dense to be displayed clearly.

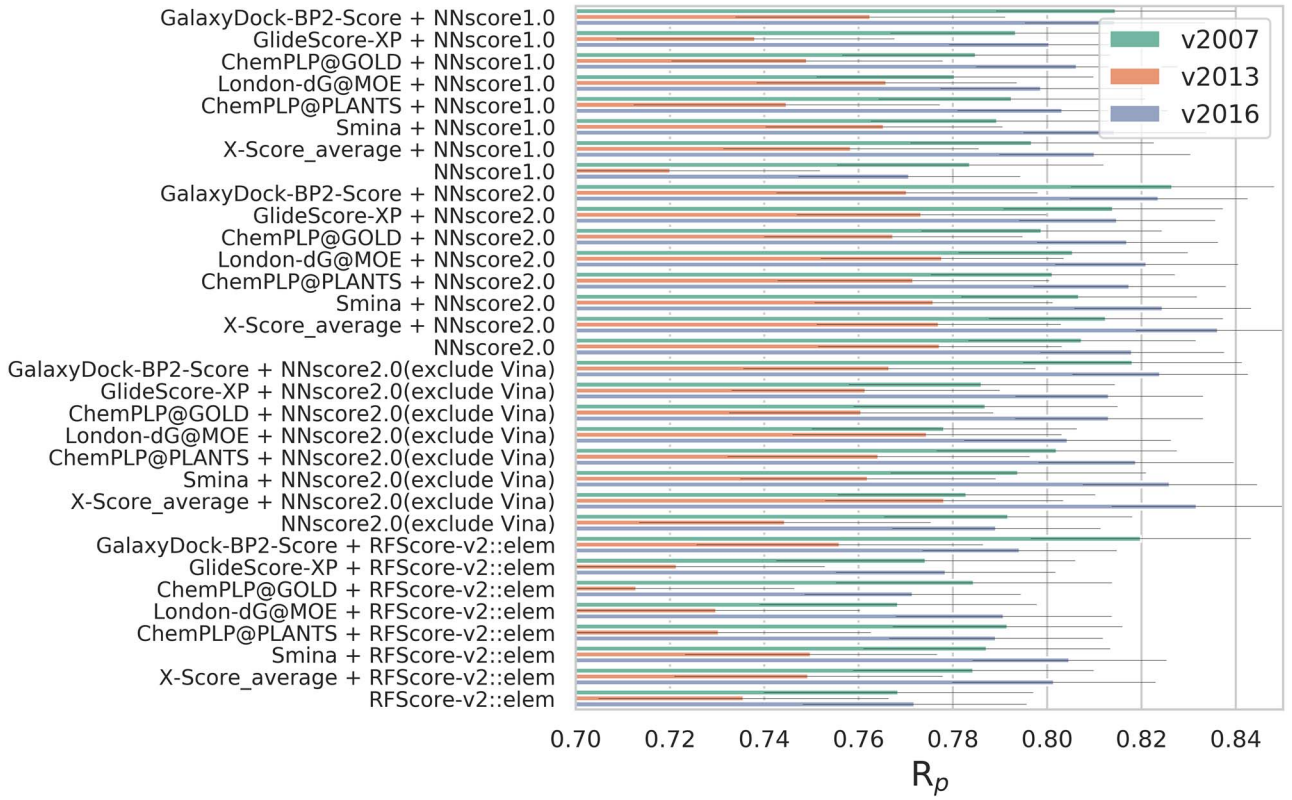

Figure 7. Comparison of the scoring powers (R_ps) of several hybrid ML-based SFs toward the core sets of PDBbind v2007, v2013 and v2016. Each SF is characterized by features from one inherent ML-based SF and one classical SF. Only the best-performed ML-based SF toward a certain set of features is retained.

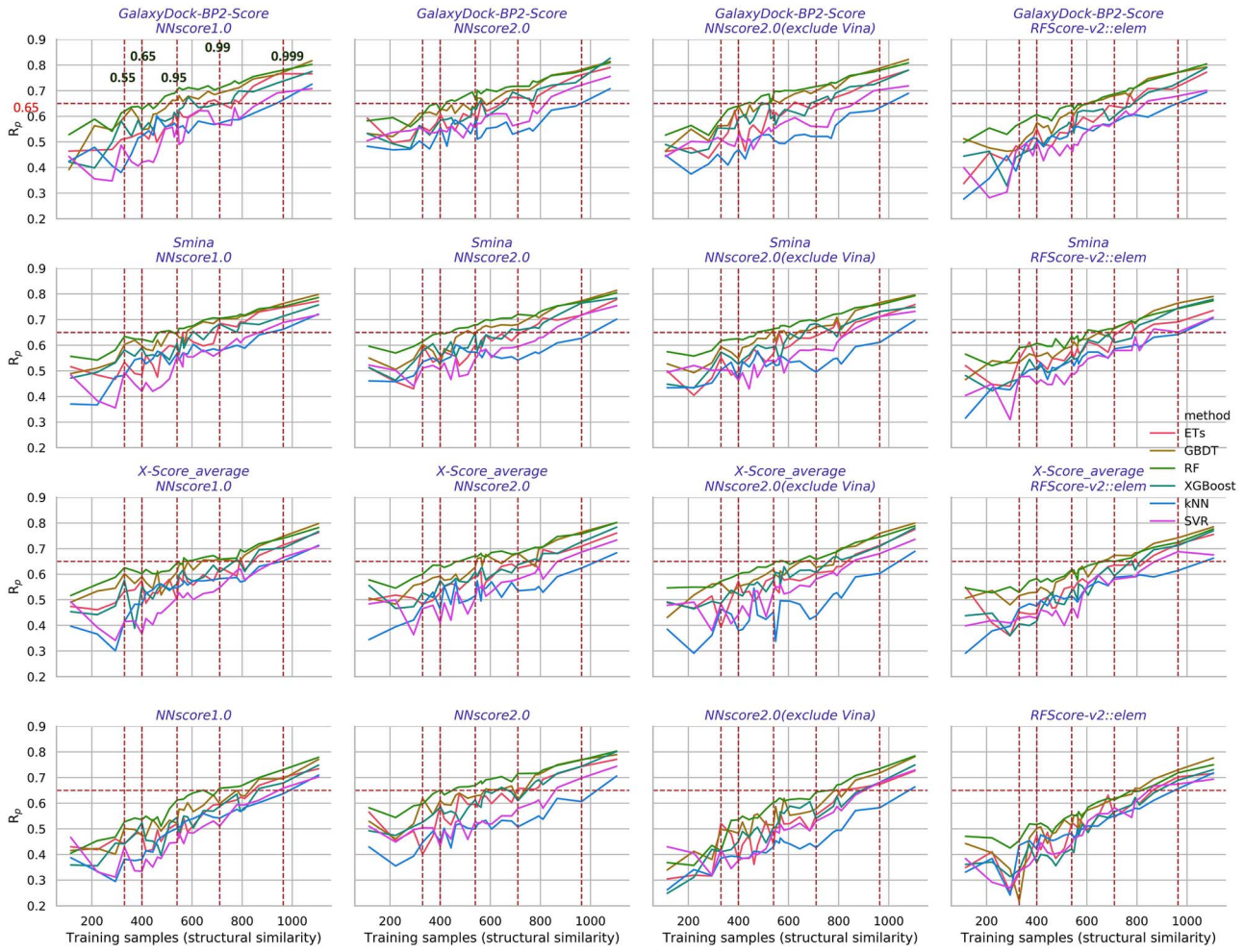

Figure 8. Test set performance (R_ps) of several hybrid ML-based SFs with increasingly similar training samples based on the structural similarity cutoff. Structural similarities of 0.55, 0.65, 0.95, 0.99 and 0.999 are marked in the figure. Standard deviations are not shown in the figure because they are too dense to be displayed clearly.

of approximately 0.996 and the sequence similarity of around 0.999 are needed to guarantee their performance. But for some SFs, such as GalaxyDock-BP2-score and NNscore 2.0, only about 500 training samples are needed to achieve $R_p \geq 0.65$. In other words, as long as there exist training samples with structural similarity higher than around 0.850 and sequence identity larger than 0.550, they can always outperform ordinary classical SFs. These observations suggest that most ML-based SFs can learn not only from highly similar samples but also from dissimilar samples with varying magnitude. Interestingly, the performance of several SFs, including Cyscore, ASP@GOLD, Alpha-HB@MOE and GBVIWSA-dG@MOE, does not change much with increasing similarity. This may be attributed to their inherent poor performance. NNscore1.0 and RFScore-v2::elem perform well on the original PDBbind core sets. However, predictive power decreases based on training samples with increasing structural or sequence dissimilarity. These phenomena illustrate that the performance of ML-based SFs characterized by simple atom type pair counts might largely rely on the composition of the training set, and they might be notably over-estimated. Thus, to better evaluate the performance of a ML-based SF, a more systematical assessment becomes necessary. Another interesting observation is that, compared with NNscore1.0, NNscore2.0 shows higher stability to the change of the structural or sequence similarity within the training sets. By incorporating the six energy terms from the Vina SF, NNscore2.0 may gain additional information even though only few similar proteins are contained in the training set.

In the previous section, it is found that RF and GBDT perform the best, and in most cases the performance of GBDT is slightly better. But according to the evaluation toward dataset II, it is obvious that RF is more stable than GBDT to handle the decrease of similar training samples. Therefore, when we want to predict the binding affinity of ligands for a target with various reported similar proteins, a GBDT-based SF may yield the best scoring power; when there are no so many similar proteins, a RF-based SF may achieve better performance; and when the target is completely novel, a classical SF may be more suitable.

Next, we further explore the impact of structural and sequence similarity of the training sets by first using the smallest set of samples with highest similarity to the test proteins and gradually involving increasingly dissimilar ones. As shown in Figures 5 and 6, S25, S26 and S29–S30, most ML-based SFs also show a remarkably better performance than their corresponding classical ones. When there are enough samples

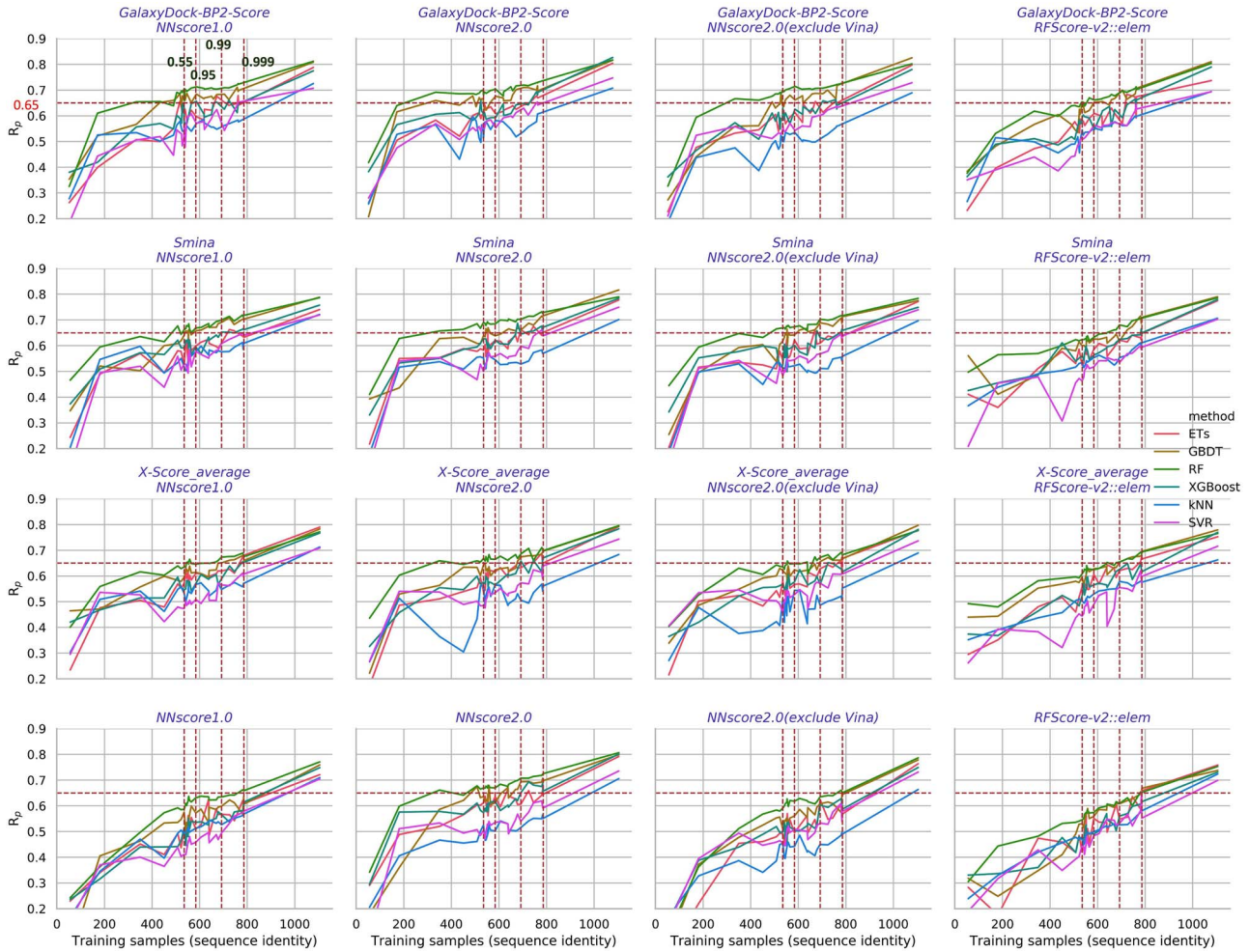

Figure 9. Test set performance (R_ps) of several hybrid ML-based SFs with increasingly similar training samples based on the sequence identity cutoff. Sequence identities of 0.55, 0.95, 0.99 and 0.999 are marked in the figure. Standard deviations are not shown in the figure because they are too dense to be displayed clearly.

in the training set (>400), the addition of more dissimilar proteins into the training set does not clearly influence the final performance too much. When the size of the training set is small, although the underlying samples are highly similar to the proteins in the test set, the performance still becomes worse, suggesting that dissimilar proteins in the training set can also provide partial valid information to improve the prediction accuracy.

Improving scoring power by combining features from multiple SFs

Inspired by the outstanding performance of NNscore2.0 reported in Section 3.1, we speculate that the combination of simple atom type pair counts and classical energy terms may provide more comprehensive information leading to higher prediction accuracy. Thus, we further explored whether the predictive power could be improved by combining two different types of features: one set of simple atom type pair counts from inherent ML-based SFs and another set of energy terms from classical SFs. The former set consists of four ML-based SFs, i.e. RFScore-v2::elem, NNscore1.0, NNscore2.0 and NNscore2.0 excluding six energy terms from Vina, whereas the latter set is composed of seven classical SFs, including GalaxyDock-BP2-Score, GlideScore-XP, ChemPLP@GOLD, London-dG@MOE, ChemPLP@PLANTS, Smina and Xscore_average. In total, 28 new ML-based hybrid SFs were constructed. The performances of these hybrid SFs based on 5-fold cross-validation of the training sets are shown in Supplementary Figures S31–S33 and those on the test sets in Figures S34–S36 (the *P*-values of any two compared ML methods based on the same set of features on the test sets are shown in Figures S37–S45). Similar to the observations described in Section 3.1, GBDT and RF yield the best R_p values for most of the SFs. However, the performances of XGBoost and SVR become rather comparable, especially when the features from NNscore2.0 are incorporated.

Next, for each hybrid SF, only the best score among all ML methods is retained, as shown in Figures 7, S46 and S47 for R_p, R_s and RMSE values, respectively (the corresponding *P*-values between any two of compared SFs are shown in Figures S48–S50). For the hybrid NNscore1.0 and RFScore-v2::elem SFs, the addition of additional classical SFs could clearly improve the predictive power in most cases. The most significant improvement is found for GalaxyDock-BP2-Score. For NNscore2.0, six energy terms of Vina were firstly removed and then the other terms were refitted by six ML algorithms, and the new SFs display decreased R_p values, suggesting that these energy terms of Vina and the simple atom type pair counts might be complementary with each other. Then, when using seven classical SFs to replace the original Vina in NNscore2.0,

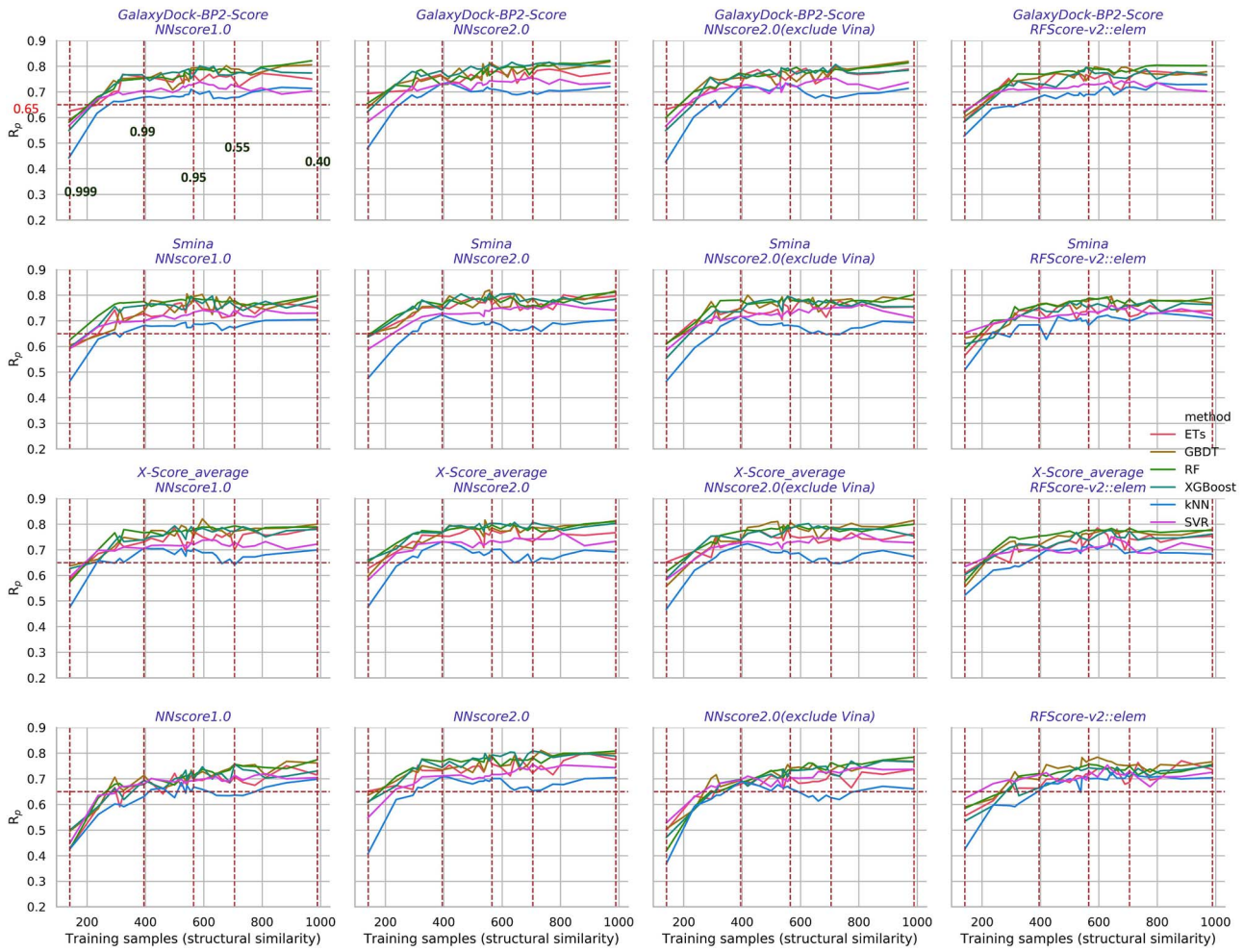

Figure 10. Test set performance (R_ps) of several hybrid ML-based SFs with increasingly dissimilar training samples based on the structural similarity cutoff. Structural similarities of 0.40, 0.55, 0.95, 0.99 and 0.999 are marked in the figure. Standard deviations are not shown in the figure because they are too dense to be displayed clearly.

improvements with varying magnitude are found for almost all hybrid SFs, except for some methods such as London-dG@MOE and Xscore_average for PDBbind v2007 (Figure 7). Among these seven classical SFs, Xscore_average and GalaxyDock-BP2-Score achieve the best performance when combining with NNscore2.0 excluding six energy terms from Vina. However, when the energy terms from classical SFs are directly hybridized with the original NNscore2.0, no obvious increase is observed. In some cases, such as ChemPLP@GOLD and ChemPLP@PLANTS, clear decrease in R_p values is observed.

To explore the impacts of structural and sequence similarities on these newly-developed ML-based SFs, three representative classical SFs, namely GalaxyDock-BP2-Score, Smina and Xscore_average, were combined with four inherent ML-based SFs characterized by simple atom type pair counts, and tested on dataset II. As shown in Figures 8 and 9 and S51–S54, significant improvements are achieved for NNscore1.0, NNscore2.0 (excluding Vina) and RFScore-v2::elem, which originally need approximately 700, 750 and 800 training samples to outperform classical SFs, respectively. However, by hybridizing with GalaxyDock-BP2-Score, Smina or Xscore_average, only around 400, 500 and 550 training samples are needed for NNscore1.0, 450, 500 and 550 for NNscore2.0 (excluding Vina), and 550, 600 and 700 for RFScore-v2::elem. No obvious improvement is found for NNscore2.0 with a combination of another classical SF. The test set performance (R_p, R_s and RMSE) based on training subsets with increasing structural and sequence dissimilarity are reported in Figures 10 and 11 and S55–S58. Similar to the results for individual SFs shown in Section 3.2, when the number of training samples is enough (>400), the performance of these combined SFs vary little by the addition of dissimilar training samples. Besides, an additional finding is that less training samples are needed for these SFs to reach $R_p \geq 0.65$. As shown in Figure 10, around 250, 200, 300 and 250 training samples are needed for NNscore1.0, NNscore2.0, NNscore2.0 (excluding Vina) and RFScore-v2::elem, respectively, to reach $R_p \geq 0.65$, while only 200 training samples are needed for hybrid SFs to reach $R_p \geq 0.65$. These findings further demonstrate the superiority of these hybrid SFs.

Impacts of the type and the number of features on scoring power

In this section, we investigate how many classical SFs can be combined in order to achieve the best performance. Here, a number of hybrid SFs were constructed by combining seven different SFs and the corresponding performance was evaluated. These seven SFs include NNscore2.0 (excluding Vina), GalaxyDock-BP2-score, Smina, Xscore_average, ChemPLP@PLANTS, London-dG@MOE and RFScore-v2::elem, which were marked as N, G, S, X, P, M and R, respectively.

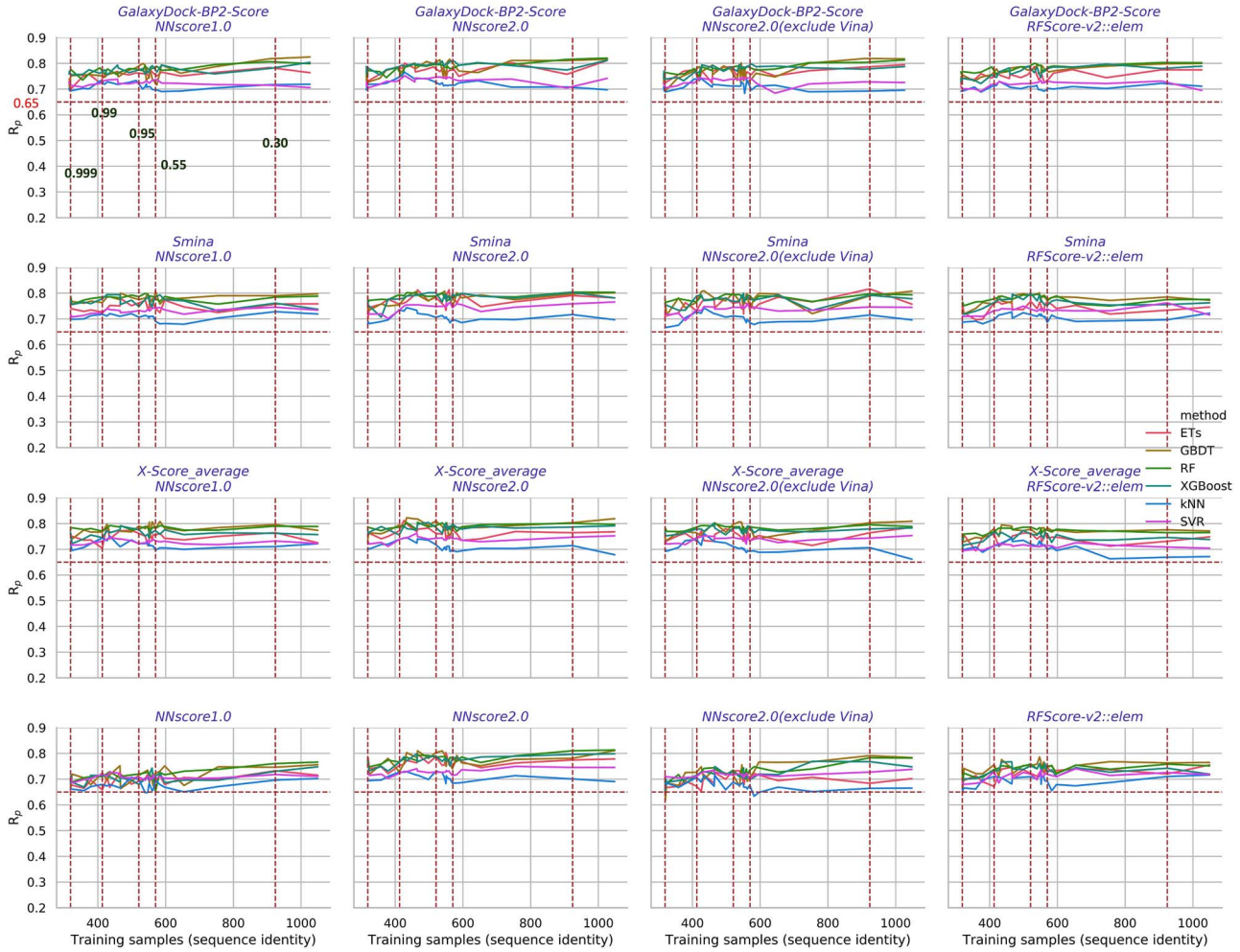

Figure 11. Test set performance (R_ps) of several hybrid ML-based SFs with increasingly dissimilar training samples based on the sequence identity cutoff. Sequence identities of 0.30, 0.55, 0.95, 0.99 and 0.999 are marked in the figure. Standard deviations are not shown in the figure because they are too dense to be displayed clearly.

First, NNscore2.0 (exclude Vina) or RFScore-v2::elem was used as the starting SF, and then other SFs were subsequently incorporated and trained using GBDT. R_p, R_s and RMSE values are reported for several representative hybrid SFs as shown in Figures 12, S59 and S60, respectively. The corresponding *P*-values of any two compared SFs in each subgraph can be found in Figures S61–S63. In Figure 12, taking the combination of NGSXPM as an example, the first point in the line represents the performance of the SFs featured by NNscore2.0 (excluding Vina) alone, the second point stands for the performance of the SFs by combining NNscore2.0 (excluding Vina) and GalaxyDock-BP2-score, the third point represents the performance of the SFs by combining NNscore2.0 (excluding Vina), GalaxyDock-BP2-score and Smina, and so on. Starting with N or R, each group roughly displays an increasing trend. However, the best performing combination is not always the last one, suggesting that more features do not necessarily lead to a better result. For example, in terms of v2007, NPMGS ($R_p = 0.821$) performs the best when starting from N while RPMG ($R_p = 0.813$) outperforms any other combination when starting with R, despite the fact that the differences between any two of the SFs are not always significant. In terms of v2013, NXPM ($R_p = 0.792$) and RXPMG ($R_p = 0.775$) show the best results based on different starting point and different numbers of combined SFs. Furthermore, for v2016, NX ($R_p = 0.832$) and RXPMG ($R_p = 0.815$) yield relatively better performance. Therefore, as we speculated above, on the one hand, some conflicts between the energy terms of two SFs may exist, thus leading to the decrease of the performance with the involvement of more SFs; on the other hand, since most classical SFs might provide similar interaction information (e.g. electrostatic, van der Waals and hydrogen-bond interactions), it is not surprising to observe that the overall performance by combining multiple classical SFs tends to become stable. Thus, some solutions are urgently needed to cope with these two problems. As for the former, feature selection [87] can be employed to select several important features from each individual SF. As for the latter, the involvement of different types of features representing protein–ligand interactions in a different prospect may effectively improve the celling of the performance, and this requires us to develop more useful feature representation methods. Besides, we constructed some SFs by only combining different classical SFs, exampled by GSXPM, SXPMG, XPMGS, PMGSX and MGSXP in Figure 12. Unfortunately, they can still not outperform most SFs involving simple atom pair counts, highlighting the importance of the incorporation of different types of features.

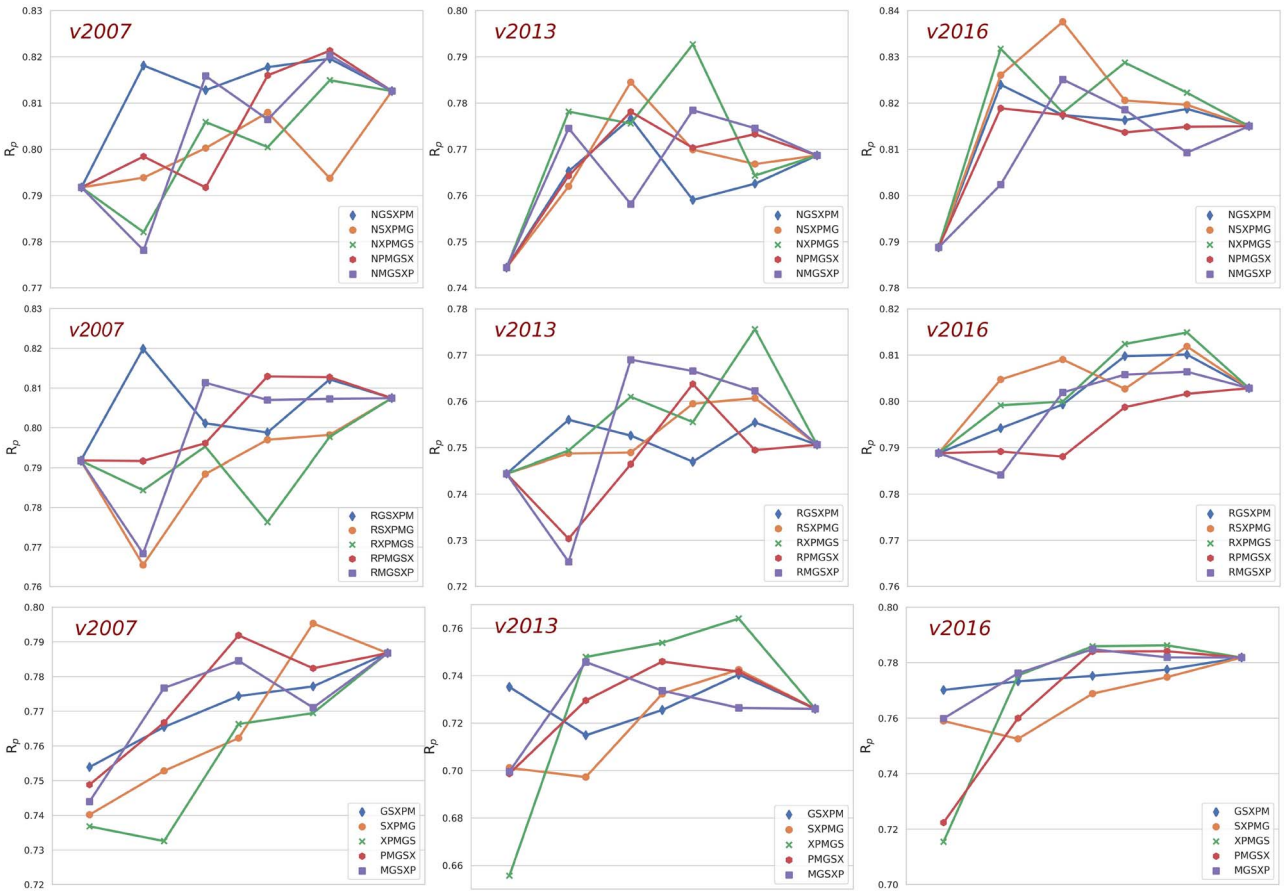

Figure 12. Test set performance (R_p) of several GBDT-based SFs. N, G, S, X, P, M and R represent NNscore2.0 (exclude Vina), GalaxyDock-BP2-score, Smina, Xscore_average, ChemPLP@PLANTS, London-dG@MOE and RFScore-v2::elem, respectively. The line of GSXPM means that the first point represents the performance of SF featured by GalaxyDock-BP2-score, the second point represents the performance of SF featured by GalaxyDock-BP2-score and Smina, the third point represents the performance of SF featured by GalaxyDock-BP2-score, Smina and Xscore_average and so on. Standard deviations are not shown in the figure because they are too dense to be displayed clearly.

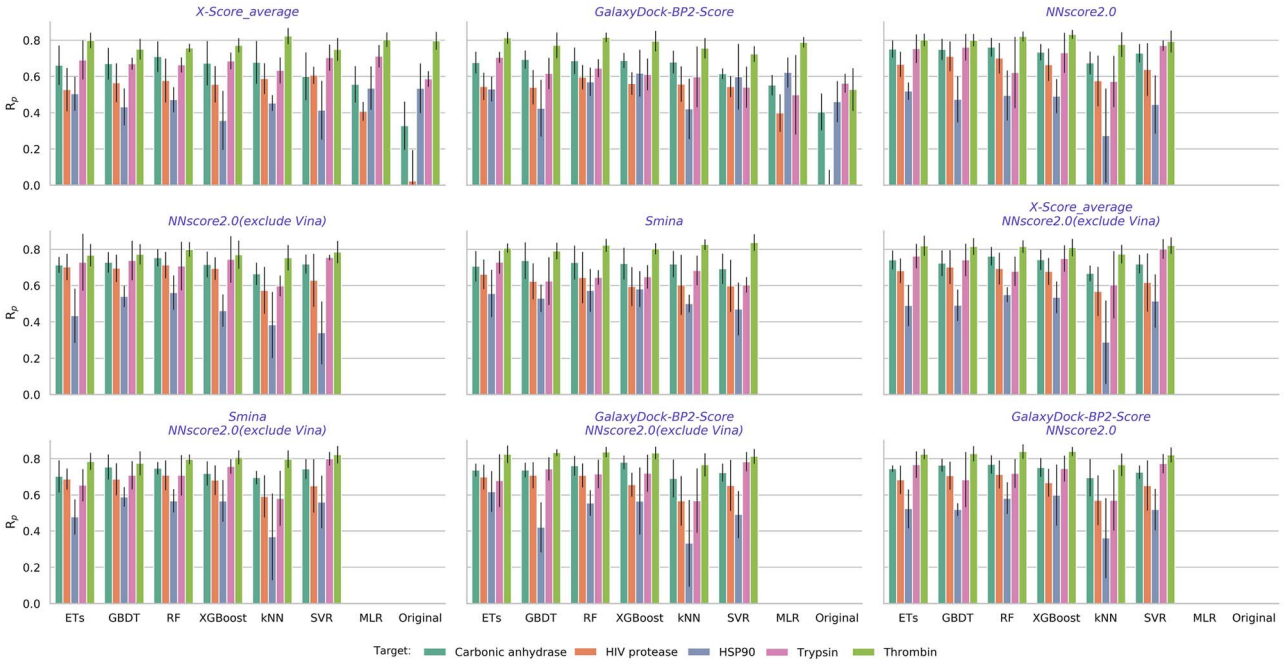

Figure 13. Comparison of the scoring powers (R_ps) of several ML-based SFs based on different ML approaches on five homogeneous test sets.

Scoring power for different targets

It is well-known that most SFs cannot yield universal performance in all situations. Target-specific SFs that only focus on a certain target gain increasing attention especially with the involvement of advanced ML methods [88]. Here, we assessed the performance of the ML-based SFs for five representative targets or target family for which relatively large numbers of complexes are available in PDBbind v2016, including carbonic anhydrases (with 214 complexes), HIV protease (285), HSP90 (61), trypsin (98) and thrombin (83). The data for each target/family were sorted first by their experimental binding affinities and then split into four equal subsets of equal sizes. To avoid the impact of dataset partitioning, three subsets were used for training and the remaining one was used for testing, and the average of the four repeats was considered as the final result. Other procedures were the same as those used to train the generic ML-based SFs. In this assessment, nine ML-based SFs were developed for each target, and the performance evaluated by R_p, R_s and RMSE values is shown in Figures 13, S64 and S65, respectively. It can be seen that the fluctuations are large for most SFs, suggesting the significant effect of training sample selection while a target-specific SF was constructed in practical applications. Additionally, the performance clearly differs among targets. For thrombin, almost all SFs can rank the binding affinities with high accuracy, whereas the performances become worse for the other four targets, especially for HSP90 that display the worst predictive power in most SFs irrespective of used ML methods. Thus, for a specific target, ML-based SFs do not guarantee better performance comparing to the classical ones; GBDT and RF are not always superior to the other ML methods; and ML-based hybrid SFs do not necessarily outperform single ones. In other words, it is not applicable to derive a generic target-specific SF or SF combination. Hence, for a given target, a systematic assessment is required to guarantee the reliability of SFs.

Conclusions

To solve the puzzle whether ML can consistently improve the scoring power of classical SFs, we have conducted a comparative assessment toward 25 commonly used SFs, where their individual energy terms were extracted from each original SF to construct new ML-based SFs. Five major aspects, including the scoring power on the single-point PDBbind core sets, the impacts of structural and sequence similarity on the scoring power, the impacts of combining features from multiple SFs on the scoring power, the impacts of the type and the number of features on the scoring power and the scoring power on homogeneous test sets, have been comprehensively discussed. For most classical SFs, the replacement of original MLR with advanced ML methods can improve their scoring power in most cases on single-point PDBbind core sets, and GBDT and RF in most cases can mostly obtain the best results. However, ML-based SFs modified from classical SFs still cannot outperform inherent ML-based SFs such as NNscore and RFScore, whose features are specifically designed for ML methods. Then, structural similarity and sequence identity between the proteins in training and test sets indeed exert a significant influence on the final result. The performance of ML-based SFs is indeed over-estimated to some extent. It is more distinct for those merely featured by simple atom type pair counts such as NNscore1.0 and RFScore-v2::elem. But, ML-based SFs can still outperform most of the classical SFs as long as enough similar proteins exist in the training samples. When combining the features from classical energy terms with simple atom type pair counts, improvements are observed not only on the single-point core set but also on the sets with varying similarity. However, when there are too many SFs used for combination, the improvement of the performance seems to reach a plateau. Sometimes, the predictive power even decreases as the involvement of more features. Finally, a number of ML-based SFs where the training and test sets both belong to the same target or family are constructed. Unfortunately, we cannot derive any universal target-specific SF or SF combination. Additional assessments might be required to evaluate its performance when a certain target-specific SF is constructed. Taken together, these findings could provide valuable insights into the development of ML-based SFs, and make further contributions to the progress of drug design and discovery.

Conflict of interest

There are no conflicts to declare.

Key Points

- A comparative assessment toward 25 commonly used SFs were conducted by refitting their individual energy terms to construct new ML-based SFs, and five major aspects of scoring power were comprehensively discussed.
- The newly-developed ML-based SFs consistently performed better than classical ones, but the ML-based SFs developed directly from classical SFs could not outperform those intrinsic ones, such as NNscore and RFScore.
- Structural and sequence similarities consistently exerted a deep influence on final performances, but the superiority of the ML-based SFs could be fully guaranteed when sufficient similar targets were contained in the training set.
- Incorporating different types of features could significantly improve the scoring power, and combining NNscore2.0 with one to four other classical SFs could yield the best result.
- Additional assessments might be required to evaluate its performance when a certain target-specific SF was constructed.

Supplementary Data

Supplementary data are available online at http://bib.oxford.journals.org/.

Funding

This work was financially supported by Key R&D Program of Zhejiang Province (2020C03010), the National Natural Science Foundation of China (21575128, 81773632) and Zhejiang Provincial Natural Science Foundation of China (LZ19H300001).

References

1. Anighoro A, Bajorath J, Rastelli G. Polypharmacology: challenges and opportunities in drug discovery. *J Med Chem* 2014;**57**:7874–87.

2. Jorgensen WL. Efficient drug lead discovery and optimization. *Acc Chem Res* 2009;**42**:724–33.
3. Kitchen DB, Decornez H, Furr JR, *et al*. Docking and scoring in virtual screening for drug discovery: methods and applications. *Nat Rev Drug Discov* 2004;**3**:935–49.
4. Guedes IA, Pereira FSS, Dardenne LE. Empirical scoring functions for structure-based virtual screening: applications, critical aspects and Challenges. *Front Pharmacol* 2018;**9**:1089.
5. Pagadala NS, Syed K, Tuszynski J. Software for molecular docking: a review. *Biophys Rev* 2017;**9**:91–102.
6. Ewing TJA, Makino S, Skillman AG, *et al*. DOCK 4.0: search strategies for automated molecular docking of flexible molecule databases. *J Comput Aided Mol Des* 2001;**15**:411–28.
7. Jones G, Willett P, Glen RC, *et al*. Development and validation of a genetic algorithm for flexible docking. *J Mol Biol* 1997;**267**:727–48.
8. Venkatachalam CM, Jiang X, Oldfield T, *et al*. LigandFit: a novel method for the shape-directed rapid docking of ligands to protein active sites. *J Mol Graph Model* 2003;**21**:289–307.
9. Trott O, Software News OAJ. Update AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization and Multithreading. *J Comput Chem* 2010;**31**:455–61.
10. Wang RX, Lai LH, Wang SM. Further development and validation of empirical scoring functions for structure-based binding affinity prediction. *J Comput Aided Mol Des* 2002;**16**:11–26.
11. Friesner RA, Banks JL, Murphy RB, *et al*. Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. *J Med Chem* 2004;**47**:1739–49.
12. Friesner RA, Murphy RB, Repasky MP, *et al*. Extra precision glide: docking and scoring incorporating a model of hydrophobic enclosure for protein-ligand complexes. *J Med Chem* 2006;**49**:6177–96.
13. Eldridge MD, Murray CW, Auton TR, *et al*. Empirical scoring functions 1. The development of a fast empirical scoring function to estimate the binding affinity of ligands in receptor complexes. *J Comput Aided Mol Des* 1997;**11**:425–45.
14. Muegge I, Martin YC. A general and fast scoring function for protein-ligand interactions: a simplified potential approach. *J Med Chem* 1999;**42**:791–804.
15. Velec HFG, Gohlke H, Klebe G. DrugScore(CSD)-knowledge-based scoring function derived from small molecule crystal data with superior recognition rate of near-native ligand poses and better affinity prediction. *J Med Chem* 2005;**48**:6296–303.
16. Debroise T, Shakhnovich EI, Cheron NA. Hybrid knowledge-based and empirical scoring function for protein-ligand interaction: SMoG2016. *J Chem Inf Model* 2017;**57**:584–93.
17. Baek M, Shin W-H, Chung HW, *et al*. GalaxyDock BP2 score: a hybrid scoring function for accurate protein-ligand docking. *J Comput Aided Mol Des* 2017;**31**:653–66.
18. Morris GM, Huey R, Lindstrom W, *et al*. AutoDock4 and AutoDockTools4: automated docking with selective receptor flexibility. *J Comput Chem* 2009;**30**:2785–91.
19. Ain QU, Aleksandrova A, Roessler FD, *et al*. Machine-learning scoring functions to improve structure-based binding affinity prediction and virtual screening, Wiley interdisciplinary reviews-computational molecular. *Science* 2015;**5**:405–24.
20. Shen C, Ding J, Wang Z, *et al*. From machine learning to deep learning: advances in scoring functions for protein-ligand docking. *Wiley Interdiscip Rev: Comput Mol Sci* 2020;**10**:e1429.
21. Ballester PJ, JBO M. A machine learning approach to predicting protein-ligand binding affinity with applications to molecular docking. *Bioinformatics* 2010;**26**:1169–75.
22. Ballester PJ, Schreyer A, Blundell TL. Does a more precise chemical description of protein-ligand complexes lead to more accurate prediction of binding affinity? *J Chem Inf Model* 2014;**54**:944–55.
23. Li H, Leung K-S, Wong M-H, *et al*. Improving AutoDock Vina using random forest: the growing accuracy of binding affinity prediction by the effective exploitation of larger data sets. *Mol Inf* 2015;**34**:115–26.
24. Zilian D, Sotriffer CA. SFCscore(RF): a random forest-based scoring function for improved affinity prediction of protein-ligand complexes. *J Chem Inf Model* 2013;**53**:1923–33.
25. Li G-B, Yang L-L, Wang W-J, *et al*. ID-Score: a new empirical scoring function based on a comprehensive set of descriptors related to protein-ligand interactions. *J Chem Inf Model* 2013;**53**:592–600.
26. Cang Z, Wei G-W. Integration of element specific persistent homology and machine learning for protein-ligand binding affinity prediction. *Int J Numer Methods Biomed Eng* 2018;**34**:e2914.
27. Jimenez J, Skalic M, Martinez-Rosell G, *et al*. K-DEEP: protein-ligand absolute binding affinity prediction via 3D-convolutional neural networks. *J Chem Inf Model* 2018;**58**:287–96.
28. Stepniewska-Dziubinska MM, Zielenkiewicz P, Siedlecki P. Development and evaluation of a deep learning model for protein-ligand binding affinity prediction. *Bioinformatics (Oxford, England)* 2018;**34**:3666–74.
29. Ragoza M, Hochuli J, Idrobo E, *et al*. Protein-ligand scoring with convolutional neural networks. *J Chem Inf Model* 2017;**57**:942–57.
30. Wang C, Zhang Y. Improving scoring-docking-screening powers of protein-ligand scoring functions using random Forest. *J Comput Chem* 2017;**38**:169–77.
31. Wojcikowski M, Ballester PJ, Siedlecki P. Performance of machine-learning scoring functions in structure-based virtual screening. *Sci Rep* 2017;**7**:46710.
32. Yan Y, Wang W, Sun Z, *et al*. Protein-ligand empirical interaction components for virtual screening. *J Chem Inf Model* 2017;**57**:1793–806.
33. Nogueira MS, Koch O. The development of target-specific machine learning models as scoring functions for docking-based target prediction. *J Chem Inf Model* 2019;**59**:1238–1252.
34. Pereira JC, Caffarena ER, dos Santos CN. Boosting docking-based virtual screening with deep learning. *J Chem Inf Model* 2016;**56**:2495–506.
35. Imrie F, Bradley AR, van der Schaar M, *et al*. Protein family-specific models using deep neural networks and transfer learning improve virtual screening and highlight the need for more data. *J Chem Inf Model* 2018;**58**:2319–2330.
36. Yasuo N, Sekijima M. Improved method of structure-based virtual screening via interaction-energy-based learning. *J Chem Inf Model* 2019;**59**:1050–1061.
37. Li L, Khanna M, Jo I, *et al*. Target-specific support vector machine scoring in structure-based virtual screening: computational validation, on vitro testing in kinases, and effects on lung cancer cell proliferation. *J Chem Inf Model* 2011;**51**:755–9.
38. Ding B, Wang J, Li N, *et al*. Characterization of small molecule binding. I. Accurate identification of strong inhibitors in virtual screening. *J Chem Inf Model* 2013;**53**:114–22.

39. Durrant JD, McCammon JA. NNScore: a neural-network-based scoring function for the characterization of protein-ligand complexes. *J Chem Inf Model* 2010;**50**:1865–71.
40. Durrant JD, McCammon JA. NNScore 2.0: a neural-network receptor-ligand scoring function. *J Chem Inf Model* 2011;**51**:2897–903.
41. Ouyang X, Handoko SD, Kwoh CK. Cscore: a simple yet effective scoring function for protein-ligand binding affinity prediction using modified Cmac learning architecture. *J Bioinform Comput Biol* 2011;**9**:1–14.
42. Arciniega M, Lange OF. Improvement of virtual screening results by docking data feature analysis. *J Chem Inf Model* 2014;**54**:1401–11.
43. Ashtawy HM, Mahapatra NR. BgN-Score and BsN-Score: bagging and boosting based ensemble neural networks scoring functions for accurate binding affinity prediction of protein-ligand complexes. *BMC Bioinf* 2015;**16**:S8.
44. Wang B, Zhao Z, Nguyen DD, *et al*. Feature functional theory-binding predictor (FFT-BP) for the blind prediction of binding free energies. *Theor Chem Acc* 2017;**136**:1–22.
45. Cang Z, Mu L, Wei G-W. Representability of algebraic topology for biomolecules in machine learning based scoring and virtual screening. *PLoS Comput Biol* 2018;**14**:e1005929.
46. Duc Duy N, Wei G-W. DG-GL: differential geometry-based geometric learning of molecular datasets. *Int J Numer Methods Biomed Eng* 2019;**35**:e3179.
47. Nguyen DD, AGL-Score WG-W. Algebraic graph learning Score for protein-ligand binding scoring, ranking, docking, and screening. *J Chem Inf Model* 2019;**59**:3291–3304.
48. Cang Z, TopologyNet WG. Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *PLoS Comput Biol* 2017;**13**:e1005690.
49. Li H, Peng J, Sidorov P, *et al*. Classical scoring functions for docking are unable to exploit large volumes of structural and interaction data. *Bioinformatics (Oxford, England)* 2019;**35**:3989–3995.
50. Ashtawy HM, Mahapatra NR. A comparative assessment of predictive accuracies of conventional and machine learning scoring functions for protein-ligand binding affinity prediction. *IEEE/ACM Trans Comput Biol Bioinform* 2015;**12**:335–47.
51. Li Y, Yang J. Structural and sequence similarity makes a significant impact on machine-learning-based scoring functions for protein-ligand interactions. *J Chem Inf Model* 2017;**57**:1007–12.
52. Wang RX, Fang XL, Lu YP, *et al*. The PDBbind database: collection of binding affinities for protein-ligand complexes with known three-dimensional structures. *J Med Chem* 2004;**47**:2977–80.
53. Berman HM, Westbrook J, Feng Z, *et al*. The protein data Bank. *Nucleic Acids Res* 2000;**28**:235–42.
54. Cheng T, Li X, Li Y, *et al*. Comparative assessment of scoring functions on a diverse test set. *J Chem Inf Model* 2009;**49**:1079–93.
55. Li Y, Liu Z, Li J, *et al*. Comparative assessment of scoring functions on an updated benchmark: 1. Compilation of the test set. *J Chem Inf Model* 2014;**54**:1700–16.
56. Su M, Yang Q, Du Y, *et al*. Comparative assessment of scoring functions: the CASF-2016 update. *J Chem Inf Model* 2018;**59**:895–913.
57. Zhang Y, Skolnick J. Scoring function for automated assessment of protein structure template quality. *Proteins: Struct Funct Bioinf* 2004;**57**:702–10.
58. Zhang Y. NW-align. http://zhanglab.ccmb.med.umich.edu/NW-align/.
59. Sastry GM, Adzhigirey M, Day T, *et al*. Protein and ligand preparation: parameters, protocols, and influence on virtual screening enrichments. *J Comput Aided Mol Des* 2013;**27**:221–34.
60. Kaminski GA, Friesner RA, Tirado-Rives J, *et al*. Evaluation and reparametrization of the OPLS-AA force field for proteins via comparison with accurate quantum chemical calculations on peptides. *J Phys Chem B* 2001;**105**:6474–87.
61. Olsson MHM, Sondergaard CR, Rostkowski M, *et al*. PROPKA3: consistent treatment of internal and surface residues in empirical pK(a) predictions. *J Chem Theory Comput* 2011;**7**:525–37.
62. Shelley JC, Cholleti A, Frye LL, *et al*. Epik: a software program for pK (a) prediction and protonation state generation for drug-like molecules. *J Comput Aided Mol Des* 2007;**21**:681–91.
63. Korb O, Stutzle T, Exner TE. Empirical scoring functions for advanced protein-ligand docking with PLANTS. *J Chem Inf Model* 2009;**49**:84–96.
64. Koes DR, Baumgartner MP, Camacho CJ. Lessons learned in empirical scoring with smina from the CSAR 2011 benchmarking exercise. *J Chem Inf Model* 2013;**53**:1893–904.
65. Molecular Operating Environment (MOE), 2018.01; Chemical Computing Group Inc., 1010 Sherbrooke St. West, Suite #910, Montreal, QC, Canada, H3A 2R7, 2018.
66. Jain AN. Scoring noncovalent protein-ligand interactions: a continuous differentiable function tuned to compute binding affinities. *J Comput Aided Mol Des* 1996;**10**:427–40.
67. Cao Y, Li L. Improved protein-ligand binding affinity prediction by using a curvature-dependent surface-area model. *Bioinformatics* 2014;**30**:1674–80.
68. Schreyer A, Blundell T. CREDO: a protein-ligand interaction database for drug discovery. *Chem Biol Drug Des* 2009;**73**:157–67.
69. Breiman L. Random forests. *Mach Learn* 2001;**45**:5–32.
70. Geurts P, Ernst D, Wehenkel L. Extremely randomized trees. *Mach Learn* 2006;**63**:3–42.
71. Friedman JH. Greedy function approximation: a gradient boosting machine. *Ann Stat* 2001;**29**:1189–232.
72. Chen T, Guestrin C. *XGBoost: A Scalable Tree Boosting System*. arXiv e-prints 2016.
73. Cortes C, Vapnik V. Support-vector networks. *Mach Learn* 1995;**20**:273–97.
74. Goldberger J, Roweis S, Hinton G, *et al*. Neighbourhood components analysis. In: *International Conference on Neural Information Processing Systems*, 2004.
75. Sainath TN, Vinyals O, Senior A, *et al*. Convolutional, long short-term memory, fully connected deep neural networks. 2015 Ieee international conference on acoustics. *Speech, Signal Process* 2015;4580–4.
76. Chollet F, others. Keras. 2015.
77. Swami A, Jain R. Scikit-learn: machine learning in python. *J Mach Learn Res* 2013;**12**:2825–30.
78. Shahriari B, Swersky K, Wang Z, *et al*. Taking the human out of the loop: a review of Bayesian optimization. *Proc IEEE* 2016;**104**:148–75.
79. Bergstra J, Komer B, Eliasmith C, *et al*. Hyperopt: a python library for model selection and hyperparameter optimization. *Comput Sci Discovery* 2015;**8**:014008.

80. Bergstra J, Bardenet R, Bengio Y, *et al*. Algorithms for hyper-parameter optimization. In: *International Conference on Neural Information Processing Systems*, 2011.
81. Nemenyi P. Distribution-free multiple comparisons. *Biometrics* 1962;**18**: 263-&.
82. Terpilowski M. Scikit-posthocs: pairwise multiple comparison tests in python. *J Open Source Software* 2019;**4**:1169.
83. Gohlke H, Hendlich M, Klebe G. Knowledge-based scoring function to predict protein-ligand interactions. *J Mol Biol* 2000;**295**:337–56.
84. Kramer C, Gedeck P. Leave-cluster-out cross-validation is appropriate for scoring functions derived from diverse protein data sets. *J Chem Inf Model* 2010;**50**:1961–9.
85. Ballester PJ, Mitchell JBO. Comments on "leave-cluster-out cross-validation is appropriate for scoring functions derived from diverse protein data sets": significance for the validation of scoring functions. *J Chem Inf Model* 2011;**51**: 1739–41.
86. Li H, Peng J, Leung Y, *et al*. The impact of protein structure and sequence similarity on the accuracy of machine-learning scoring functions for binding affinity prediction. *Biomolecules* 2018;**8**:12.
87. Saeys Y, Inza I, Larranaga P. A review of feature selection techniques in bioinformatics. *Bioinformatics* 2007;**23**:2507–17.
88. Seifert MHJ. Targeted scoring functions for virtual screening. *Drug Discov Today* 2009;**14**:562–9.