Machine Learning Scoring Functions Based on Random Forest and Support Vector Regression

Pedro J. Ballester

European Bioinformatics Institute, Cambridge, UK
pedro.ballester@ebi.ac.uk

Abstract. Accurately predicting the binding affinities of large sets of diverse molecules against a range of macromolecular targets is an extremely challenging task. The scoring functions that attempt such computational prediction exploiting structural data are essential for analysing the outputs of Molecular Docking, which is in turn an important technique for drug discovery, chemical biology and structural biology. Conventional scoring functions assume a predetermined theory-inspired functional form for the relationship between the variables that characterise the complex and its predicted binding affinity. The inherent problem of this approach is in the difficulty of explicitly modelling the various contributions of intermolecular interactions to binding affinity.

Recently, a new family of 3D structure-based regression models for binding affinity prediction has been introduced which circumvent the need for modelling assumptions. These machine learning scoring functions have been shown to widely outperform conventional scoring functions. However, to date no direct comparison among machine learning scoring functions has been made. Here the performance of the two most popular machine learning scoring functions for this task is analysed under exactly the same experimental conditions.

Keywords: molecular docking, scoring functions, machine learning, chemical informatics, structural bioinformatics.

1 Introduction

Docking has two stages: predicting the position, orientation and conformation of a molecule when docked to the target's binding site (pose generation), and predicting how strongly the docked pose of such putative ligand binds to the target (scoring). Whereas there are many relatively robust and accurate algorithms for pose generation, the inaccuracies of current scoring functions continue to be the major limiting factor for the reliability of docking [1]. Indeed, despite extensive research, accurately predicting the binding affinities of large sets of diverse protein-ligand complexes remains one of the most important and difficult active problems in computational chemistry.

Scoring functions are traditionally classified into three groups: force field, knowledge-based and empirical. Force-field scoring functions parameterise the potential energy of a complex as a sum of energy terms arising from bonded and non-bonded interactions [2]. The functional form of each of these terms is characteristic of the

T. Shibuya et al. (Eds.): PRIB 2012, LNBI 7632, pp. 14–25, 2012.

particular force field, which in turn contains a number of parameters that are estimated from experimental data and computer simulations. Knowledge-based scoring functions use the three dimensional co-ordinates of a large set of protein-ligand complexes as a knowledge base. In this way, a putative protein-ligand complex can be assessed on the basis of how similar its features are to those in the knowledge base. The features used are often the distributions of atom-atom distances between protein and ligand in the complex. Features commonly observed in the knowledge base score favourably, whereas less frequently observed features score unfavourably. When these contributions are summed over all pairs of atoms in the complex, the resulting score is converted into a pseudo-energy function, typically through a reverse Boltzmann procedure, in order to provide an estimate of the binding affinity (e.g. [3]). Lastly, empirical scoring functions calculate the free energy of binding as a sum of contributing terms, each identified with a physicochemically distinct contribution to the binding free energy such as: hydrogen bonding, hydrophobic interactions, van der Waals interactions and the ligand's conformational entropy. Each of these terms is multiplied by a weight and the resulting parameters estimated from binding affinities. In addition to scoring functions, there are other computational techniques, such as those based on molecular dynamics simulations, that provide a more accurate prediction of binding affinity. However, these expensive calculations remain impractical for the evaluation of large numbers of protein–ligand complexes and are generally limited to series of congeneric molecules binding to a single target [2,4,5].

For the sake of efficiency, scoring functions do not fully account for some physical processes that are important for molecular recognition, which in turn limits their ability to select and rank-order small molecules by computed binding affinities. It is generally believed [4] that the two major sources of error in scoring functions are their limited description of protein flexibility and the implicit treatment of solvent. In addition to these enabling simplifications, there is an important computational issue that has received little attention until recently [6]. Each scoring function assumes a predetermined theory-inspired functional form for the relationship between the variables that characterise the complex, which also include a set of parameters that are fitted to experimental or simulation data, and its predicted binding affinity. Such relationship takes the form of a sum of weighted physicochemical contributions to binding in the case of empirical scoring functions or a reverse Boltzmann methodology in the case of knowledge-based scoring functions. The inherent problem of this rigid approach is that it leads to poor predictivity in those complexes that do not conform to the modelling assumptions (see [7] for an insightful discussion of this issue). As an alternative to these conventional scoring functions, nonparametric machine learning can be used to implicitly capture binding interactions that are hard to model explicitly. By not imposing a particular functional form for the scoring function, intermolecular interactions can be directly inferred from experimental data, which should lead to scoring functions with greater generality and prediction accuracy. This unconstrained approach was likely to result in performance improvement, as it is well-known that the strong assumption of a predetermined functional form for a scoring function constitutes an additional source of error (e.g. imposing an additive form for the considered energetic contributions [8]). Incidentally, recent experimental results have resulted in a redefinition of molecular interactions such as the hydrogen bond [9] or the hydrophobic interaction [10] which means that previously proposed functional forms may need to be revised accordingly.

While there have been a number of machine learning classifiers exploiting x-ray structural data for discriminating between binders and non-binders (e.g. [11,12]), it is only recently that machine learning for nonlinear regression has been shown [6] to be a powerful approach to build generic scoring functions. This trend has been highlighted [13-15] as a particularly promising approach. Indeed, a growing number of studies showing the benefits of these techniques are being presented [6,15-18]. However, these new scoring functions have all been using different benchmarks to evaluate their performance. This prevents us from being able to compare them to each other, as the performance of a scoring function can vary dramatically depending not only on the selection of test set, but also that of the training set and interaction features. In this paper, the performance of the two most popular machine learning approaches to scoring, Random Forest (RF) [19] and SVM epsilon-regression (SVR) [20], is investigated. The focus will be on generic, rather than family-specific (e.g. [21]), scoring functions, which constitute a harder regression problem due to the higher nonlinearity introduced by diverse protein-ligand complexes.

2 Machine Learning Scoring Functions

2.1 RF-Score [6]

RF-Score uses RF as the regression model. A RF is an ensemble of many different decision trees randomly generated from the same training data. RF trains its constituent trees using the CART algorithm [22]. As the learning ability of an ensemble of trees improves with the diversity of the trees [19], RF promotes diverse trees by introducing the following modifications in tree training. First, instead of using the same data, RF grows each tree without pruning from a bootstrap sample of the training data (i.e. a new set of N complexes is randomly selected with replacement from the N training complexes, so that each tree grows to learn a closely related but slightly different version of the training data). Second, instead of using all features, RF selects the best split at each node of the tree from a typically small number (m_{try}) of randomly chosen features. This subset changes at each node, but the same value of m_{try} is used for every node of each of the P trees in the ensemble. RF performance does not vary significantly with P beyond a certain threshold and thus P=500 was set as a sufficiently large number of trees. In contrast, m_{try} has some influence on performance and thus constitutes the only tuning parameter of the RF algorithm. In regression problems, the RF prediction is given by arithmetic mean of all the individual tree predictions in the forest. RF also has a built-in tool to measure the importance of individual features across the training set based on the process of “noising up”.

RF-Score outperformed [6] 16 state-of-the-art scoring functions on the same independent test set (2007 PDBbind core set [23]). To investigate the impact of chance correlation [24], the relationship between features and binding affinity in the training set was destroyed by performing a random permutation of binding affinities, while leaving the interaction features untouched (a process known as Y-randomisation). After training, the resulting RF model was used to predict the test set. Over ten independent trials, performance on the test set was on average R=–0.018 with standard deviation S_R=0.095, which demonstrated the negligible contribution of chance correlation to RF-Score's prediction ability. Additional methodological considerations are discussed in [13].

2.2 Breneman and Co-workers [16]

The next three scoring functions used SVR as the regression model. SVR searches for the hyperplane that best discriminates between two classes of feature vectors: those for which the error in the value predicted by the regression model is below a sufficiently small value ε and those with a higher error. Vectors with higher error are used to guide this search. As in SVM classifiers, nonlinear kernels may be used to map input features onto a higher dimensional feature space where better discriminating hyperplanes are possible.

The 2005 release of the PDBbind benchmark was used in this study. Five different non-overlapping training/test data partitions were made: (refined-core)/core with 977/278 complexes, core/(refined-core) with 278/977 complexes and three random partitions with 278/977 complexes each. Each complex was represented by a set of Property-Encoded Shape Distributions (PESD) features encoding geometry, electrostatic potential and polarity for both the protein and the ligand interaction surfaces [16]. Several scoring functions based on SVM regression as implemented in the e1071 SVM R package [25] were presented. No feature selection was employed except for the removal of invariant columns prior to training. SVM was trained with two control parameters: the gamma parameter of the default radial kernel and cost of contraints violation parameter. A range of models was defined by considering a number of values for both parameters and for each of these five-fold cross-validation over the training set was carried out (this cross-validation process was repeated 10 times with different random seeds). The selected SVM model was that with the highest average correlation coefficient with measured binding affinity over the validation sets.

The performance of PESD-SVM scoring functions were compared against SFCScore, as the latter family of multivariate linear regression scoring functions was shown to perform better than 14 other scoring functions on a common test set [26]. The comparison between PESD-SVM and SFCScore could only be semiquantitative on comparably sized training/test partitions, as there was only an overlap of 700 complexes between the data sets used in each study. The performance was comparable in general and slightly improved in some cases. These results are particularly valuable taking into account that, unlike PESD-SVM, SFCScore also included nonsurface-based features, its training set had complexes in common with the test set and it was enriched with industrial data through the Scoring Function Consortium, a collaborative effort with various pharmaceutical companies and the Cambridge Crystallographic Data Center.

2.3 Xie, Bourne and Co-workers [15]

This study presents a SVM regression model to predict IC50 values. While this is not a generic scoring function, the study is relevant to our analysis in that it builds upon the idea that performance improvement can be achieved by circumventing error-prone modelling assumptions with nonparametric machine learning. In particular, the authors focus on the fact that noncovalent interactions often depend on one another in a nonlinear manner and hence a nonlinear function of energy terms should lead to more accurate scoring functions that the linear combinations widely used in standard scoring functions. The docking program eHiTS [27] was selected for this study because it calculates a large number of individual energy terms, which contribute to the overall energy score also known as the eHiTS-Energy scoring function.

SVM-light [28] was used to train a regression model with 80 InhA experimental IC50 values in negative log units. 67 of these 80 molecules were not co-crystallised with the target and hence had to be docked into a InhA structure (PDB code: 1BVR). The eHiTS output provided a total of 20 different energy terms contributing to the overall energy score. In order to determine the optimal combination of energy terms for regression, 128 different combinations of these features and four SVM kernel functions (linear, polynomial, radial basis function and sigmoid tanh) were considered. Five-fold cross-validation was applied to select the final model. The model with the highest mean Spearman's correlation coefficient over all five partitions was selected, which corresponded to the linear kernel and one of the considered combinations of features. Feature importance was measured by re-training the selected model using all but a given feature. The left-out feature that resulted in the largest decrease in performance was deemed as the most important feature for regression.

The selected SVM model obtained in a large improvement in the Spearman's correlation coefficient (0.607), when compared with that achieved by the eHiTS-Energy scoring function (0.117). The mean correlation coefficient in 100 Y-randomisation trials of this SVM model was 0.079, which means that chance correlation makes a very minor contribution to performance. These results demonstrate that assuming an additive form for empirical scoring functions is a suboptimal setting.

2.4 Meroueh and Co-workers [17]

Two generic scoring functions based on SVR were presented: SVR-KB and SVR-EP. SVR-KB employs the same representation as RF-Score, as it 1-tier encoding counts atomic contacts within the same distant cutoff. Additional features were considered through several binning strategies, different atom types and scaling the pairwise counts as pair knowledge-based potentials [17]. To build the SVR-EP model, a feature selection protocol using Simulated Annealing [29] was applied leading to the use of four of the 14 physicochemical properties considered. LibSVM v3.0 [30] was used for model training and prediction using the Gaussian radial basis function (RBF) kernel. Grid search was conducted on some of the most important learning control parameters, such as ε in the loss function, gamma in the RBF kernel as well as the tradeoff between training error and margin, to give the best performance in a five-fold cross-validation. In each cross-validation, 20 runs were performed on a random split bases and the quantity of average was recorded.

Two new test data sets from CSAR [31] were used. SVR-KB trained with the 2010 release of the PDBbind refined set (2292 complexes) resulted in the best performance as measured by several measures such as the square of Pearson's correlation coefficient (R^2=0.67). Compared to seven widely used scoring functions on these test sets, SVR-KB outperformed the best of these by nearly 0.2 in R^2. The SVR-EP also resulted in superior performance, although at a lower level than SVR-KB. In contrast, conventional scoring functions tested on the same test set obtained an R^2 in the range 0.44 to 0.00.

3 Experimental Setup

These machine learning techniques for regression are used here to learn the nonlinear relationship between the atomic-level description of the protein-ligand complex as provided by a X-ray crystal structure and its binding affinity. This approach requires the characterisation of each structure as a set of features relevant for binding affinity (Figure 1 illustrates such characterisation for a particular protein-ligand complex).

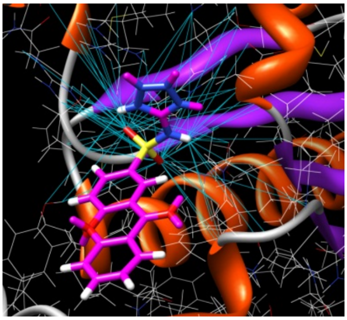

Fig. 1. Visualisation of the GAJ ligand molecule complexed with *Helicobacter Pylori* Type II Dehydroquinase (PDB code 2C4W). Protein-ligand atomic contacts are pictured as blue lines (only a fraction of these contacts are shown to avoid cluttering the figure).

Usually, each feature will comprise the number of occurrences of a particular protein-ligand atom type pair interacting within a certain distance range. This representation can have a significant impact on performance, as a number of conflicting objectives have to be balanced such as selecting atom types that result in dense features while allowing a direct interpretation in terms of which intermolecular interactions contribute the most to binding in a particular complex. On the other hand, the independent variable of this regression is the binding affinity of the ligand for this target. Binding affinities uniformly span many orders of magnitude and hence are typically log-transformed. It is also a common practice to merge dissociation constant (K_d) and inhibition constant (K_i) measurements in a single binding constant K, as this increments the amount of data that can be used to train the machine learning algorithm and has been seen elsewhere that distinguishing between both data types does not lead to significant performance improvement.

The PDBbind benchmark [23] is an excellent choice for validating generic scoring functions. It is based on the 2007 version of the PDBbind database [32], which contains a particularly diverse collection of protein-ligand complexes, assembled through a systematic mining of the entire Protein Data Bank [33]. The first construction step was to identify all the crystal structures formed exclusively by protein and ligand molecules. This excluded protein-protein and protein-nucleic acid complexes, but not oligopeptide ligands as they do not normally form stable secondary structures by themselves and therefore may be considered as common organic molecules. Secondly, Wang et al. collected binding affinity data for these complexes from the literature. Emphasis was placed on reliability, as the PDBbind curators manually reviewed all binding affinities from the corresponding primary journal reference in the PDB.

In order to generate a refined set suitable for validating scoring functions, the following data requirements were additionally imposed. First, only complete and binary complex structures with a resolution of 2.5Å or better were considered. Second, complexes were required to be non-covalently bound and without serious steric clashes. Third, only high quality binding data were included. In particular, only complexes with known K_d or K_i were considered, leaving those complexes with assay-dependent IC_{50} measurements out of the refined set. Also, because not all molecular modelling software can handle ligands with uncommon elements, only complexes with ligand molecules containing just the common heavy atoms (C, N, O, F, P, S, Cl, Br, I) were considered. In the 2007 PDBbind release, this process led to a refined set of 1300 protein-ligand complexes with their corresponding binding affinities. Still, the refined set contains a higher proportion of complexes belonging to protein families that are overrepresented in the PDB. This was considered detrimental to the goal of identifying those generic scoring functions that will perform best over all known protein families. To minimise this bias, a core set was generated by clustering the refined set according to BLAST sequence similarity (a total of 65 clusters were obtained using a 90% similarity cutoff). For each cluster, the three complexes with the highest, median and lowest binding affinity were selected, so that the resulting set had a broad and fairly uniform binding affinity coverage. By construction, this core set is a large, diverse, reliable and high quality set of protein-ligand complexes suitable for validating scoring functions. The PDBbind benchmark essentially consists of testing the predictions of scoring functions on the 2007 core set, which comprises 195 diverse complexes with measured binding affinities spanning more than 12 orders of magnitude.

Regarding representation, atom types are selected so as to generate features that are as dense as possible, while considering all the heavy atoms commonly observed in PDB complexes (C, N, O, F, P, S, Cl, Br, I). As the number of protein-ligand contacts is constant for a particular complex, the more atom types are considered the sparser the resulting features will be. Therefore, a minimal set of atom types is selected by considering atomic number only. Furthermore, a smaller set of interaction features has the additional advantage of leading to computationally faster scoring functions. In this way, the features are defined as the occurrence count of intermolecular contacts between elemental atom types i and j:

$$x_{j,i} \equiv \sum_{k=1}^{K_j} \sum_{l=1}^{L_i} \Theta(d_{cutoff} - d_{kl})$$

where d_{kl} is the Euclidean distance between k^{th} protein atom of type j and the l^{th} ligand atom of type i calculated from the PDBbind structure; K_j is the total number of protein atoms of type j and L_i is the total number of ligand atoms of type i in the considered complex; Θ is the Heaviside step function that counts contacts within a d_{cutoff} neighbourhood of the given ligand atom. For example, $x_{7,8}$ is the number of occurrences of protein nitrogen hypothetically interacting with a ligand oxygen within a chosen neighbourhood. This representation led to a total of 81 features, of which 45 are necessarily zero across PDBbind complexes due to the lack of proteinogenic amino acids with F, P, Cl, Br and I atoms. Therefore, each complex was characterised by a vector with 36 integer-valued features.

Lastly, just as in Cheng et al. [23], the 1105 complexes in the PDBbind 2007 refined set that are not in the core set will be used as the training set, whereas the core set of 195 complexes will be used as the independent test set. In this way, a set of protein-ligand complexes with measured binding affinity can be processed to give two non-overlapping data sets, where each complex is represented by its feature vector $\vec{x}^{(n)}$ and its binding affinity $y^{(n)}$:

$$D_{train} = \left\{\left(y^{(n)}, \vec{x}^{(n)}\right)\right\}_{n=1}^{1105}; D_{test} = \left\{\left(y^{(n)}, \vec{x}^{(n)}\right)\right\}_{n=1106}^{1300}; y \equiv -\log_{10} K$$

4 Results and Discussion

The SVR RBF kernel implementation in the caret package [34] of the statistical software suite R was used. As with previous studies [16], grid search was conducted on the gamma parameter in the RBF kernel (γ) and the cost of constraint violation parameter (C) to give the best performance in a five-fold cross-validation of the training set. In each cross-validation, SVR was trained using the 36 combinations of parameter values arising from $\gamma\in\{0.01,0.1,1,10,100,1000\}$ and $C\in\{0.25,0.5,1,2,4,8\}$. Thereafter, the average root mean square error between predicted and measured binding affinity across the five cross-validation sets (i.e. those not used to train the SVR) was calculated for each (γ,C) combination and that with the lowest value was selected to train on the entire training set to give SVR-Score≡SVR(γ=0.1,C=1). This model selection procedure is intended to find the model that is most likely to generalize to independent test data sets. When ran on the independent test set, SVR-Score achieved a Pearson's correlation of R=0.726, Spearman's correlation Rs=0.739 and standard deviation SD=1.70 as illustrated in Figure 2 (left).

The same R package was employed to build and run this version of RF-Score. Model selection was carried out by five-fold cross-validation. In each cross-validation, RF was trained using the 35 mtry values that cover all the feature subset sizes up to the number of interaction features, i.e. mtry $\in\{2,3,\ldots,36\}$. Thereafter, the average root mean square error between predicted and measured binding affinity across the five cross-validation sets was calculated for each mtry value and that with the lowest value was selected to train on the entire training set to give RF-Score≡RF(mtry=5). In the independent test set, RF-Score achieved a Pearson's correlation of R=0.774, Spearman's correlation Rs=0.762 and standard deviation of 1.59 as illustrated in Figure 2 (right).

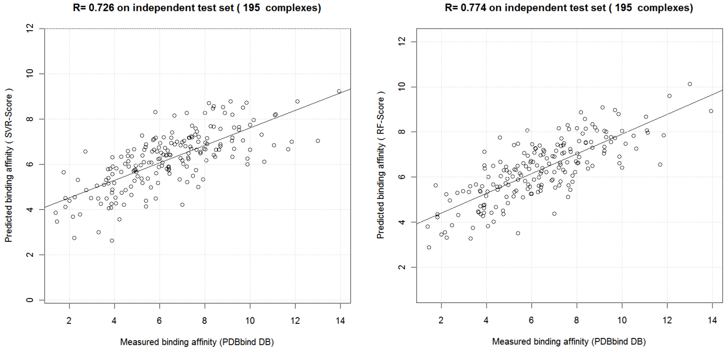

Fig. 2. SVR-Score predicted versus measured binding affinity (left) and RF-Score predicted versus measured binding affinity (right) on the independent test set (195 complexes)

Table 1. Performance of scoring functions on the PDBbind benchmark

scoring function	R	R_s	SD
RF-Score	0.774	0.762	1.59
SVR-Score	0.726	0.739	1.70
X-Score::HMScore	0.644	0.705	1.83
DrugScore^{CSD}	0.569	0.627	1.96
SYBYL::ChemScore	0.555	0.585	1.98
DS::PLP1	0.545	0.588	2.00
GOLD::ASP	0.534	0.577	2.02
SYBYL::G-Score	0.492	0.536	2.08
DS::LUDI3	0.487	0.478	2.09
DS::LigScore2	0.464	0.507	2.12
GlideScore-XP	0.457	0.435	2.14
DS::PMF	0.445	0.448	2.14
GOLD::ChemScore	0.441	0.452	2.15
SYBYL::D-Score	0.392	0.447	2.19
DS::Jain	0.316	0.346	2.24
GOLD::GoldScore	0.295	0.322	2.29
SYBYL::PMF-Score	0.268	0.273	2.29
SYBYL::F-Score	0.216	0.243	2.35

Next, the performance of RF-Score and SVR-Score is compared against that of a broad range of scoring functions on the PDBbind benchmark [23]. Using a pre-existing benchmark, where other scoring functions had previously been tested, ensures the optimal application of such functions by their authors and avoids the danger of constructing a benchmark complementary to the presented scoring function. Table 1 reports the performance of all scoring functions on the independent test set, with RF-Score obtaining the best performance followed by SVR-Score. In contrast, conventional scoring functions tested on the same test set obtained a lower correlation spanning from 0.216 to 0.644.

Given the secrecy of proprietary scoring functions, it is not possible to obtain full implementation details of these, often including training set composition. Consequently, in the context of this benchmark, it could only be reported [23] that, unlike RF-Score and SVR-Score, top scoring functions such as X-Score::HMScore, DrugScoreCSD, SYBYL::ChemScore and DS::PLP1 have an undetermined number of training complexes in common with this test set, which constitutes an advantage for the latter set of functions. On the other hand, calibration sets for conventional scoring functions typically contain around 100-300 selected complexes and hence training these functions with the 1105 complexes from this study could in principle lead to some improvement (note however that the latter strongly depends on whether the adopted regression model is sufficiently flexible to assimilate larger amounts of data and still keep overfitting under control). This issue was investigated in [23], where the third best performing function in Table 1 (best performing in that study), X-Score::HMScore, was recalibrated by its authors using exactly the same 1105 training complexes as RF-Score and SVR-Score (i.e. ensuring that training and test sets have no complexes in common). This gave rise to X-Score::HMScore v1.3, which obtained practically the same performance as v1.2 (R=0.649 versus R=0.644). Since RF-Score, SVR-Score and X-Score::HMScore v1.3 used exactly the same training set and were tested on exactly the same test set, this result also means that all the performance gain (R=0.774 and R=0.726 versus R=0.649) is guaranteed to come from the scoring function characteristics, ruling out any influence of using different training sets on performance. While this recalibration remains to be investigated for the remaining scoring functions (this can only be done by their developers), the fact that these perform much worse than RF-Score/SVR-Score along with the very small improvement obtained by recalibrating X-Score::HMScore strongly suggests that the top part of the ranking in Table 1 would remain exactly the same.

5 Conclusions and Future Prospects

Machine learning for nonlinear regression is a largely unexplored approach to develop generic scoring functions. Here, a comparison between RF and SVR as the regression models has been carried out. Using the same training set, test set, interaction features and model selection strategy, it was observed that RF-Score performs better than SVR-Score at predicting binding affinity. In turn, both machine learning scoring functions outperformed a set of 16 established scoring functions on the same independent test set, which demonstrate the benefits of circumventing problematic modeling assumptions via nonparametric machine learning.

Future prospects for this new class of scoring functions are exciting, as there is a number of promising research avenues which are likely to lead to further performance improvements. First, only three nonparametric machine learning techniques have been used to date (RF, SVR and Multi-Layer Perceptron) and hence alternative techniques might be more suitable for this problem. Second, only two model selection strategies have been applied so far (OOB and five-fold cross-validation) and therefore it remains to be seen whether other strategies could lead to reduced overfitting. Third, unlike models with fixed structure, nonparametric machine learning techniques are sufficiently flexible to effectively assimilate large volumes of training data. Indeed, it has been observed [6,17] that performance on the test set improves dramatically with increasing training set size. This means that ongoing efforts to compile and curate additional experimental data should eventually lead to more accurate and general scoring functions. Finally, in order to facilitate the use, analysis and future development of machine learning-based scoring functions, RF-Score code is made available at http://www.ebi.ac.uk/~pedrob/software.html.

Acknowledgements. The author thanks the Medical Research Council for a Methodology Research Fellowship.

References

1. Moitessier, N., et al.: Towards the development of universal, fast and highly accurate docking/scoring methods: a long way to go. Br. J. Pharmacol. 153, S7–S26 (2008)
2. Huang, N., et al.: Molecular mechanics methods for predicting protein-ligand binding. Phys. Chem. Chem. Phys. 8, 5166–5177 (2006)
3. Mitchell, J.B.O., et al.: BLEEP - potential of mean force describing protein-ligand interactions: I. Generating potential. J. Comput. Chem. 20, 1165–1176 (1999)
4. Guvench, O., MacKerell Jr., A.D.: Computational evaluation of protein-small molecule binding. Curr. Opin. Struct. Biol. 19, 56–61 (2009)
5. Michel, J., Essex, J.W.: Prediction of protein–ligand binding affinity by free energy simulations: assumptions, pitfalls and expectations. J. Comput. Aided Mol. Des. 24, 639–658 (2010)
6. Ballester, P.J., Mitchell, J.B.O.: A machine learning approach to predicting protein-ligand binding affinity with applications to molecular docking. Bioinformatics 26, 1169–1175 (2010)
7. Marshall, G.R.: Limiting assumptions in structure-based design: binding entropy. J. Comput. Aided Mol. Des. 26(1), 3–8 (2012)
8. Baum, B., Muley, L., Smolinski, M., Heine, A., Hangauer, D., Klebe, G.: Non-additivity of functional group contributions in protein-ligand binding: a comprehensive study by crystallography and isothermal titration calorimetry. J. Mol. Biol. 397, 1042–1054 (2010)
9. Arunan, E., et al.: Definition of the hydrogen bond (IUPAC Recommendations 2011). Pure and Applied Chemistry 83, 1637–1641 (2011)
10. Snyder, P.W., et al.: Mechanism of the hydrophobic effect in the biomolecular recognition of arylsulfonamides by carbonic anhydrase. Proceedings of the National Academy of Sciences 108, 17889–17894 (2011)
11. Li, L., Li, J., Khanna, M., Jo, I., Baird, J.P., Meroueh, S.O.: Docking to Erlotinib Off-Targets Leads to Inhibitors of Lung Cancer Cell Proliferation with Suitable in Vitro Pharmacokinetics. ACS Med. Chem. Lett. 1(5), 229–233 (2010)
12. Durrant, J.D., McCammon, J.A.: NNScore: A Neural-Network-Based Scoring Function for the Characterization of Protein–Ligand Complexes. J. Chem. Inf. Model. 50(10), 1865–1871 (2010)

13. Ballester, P.J., Mitchell, J.B.O.: 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. 51, 1739–1741 (2011)
14. Cheng, T., Li, Q., Zhou, Z., Wang, Y., Bryant, S.H.: Structure-Based Virtual Screening for Drug Discovery: a Problem-Centric Review. The AAPS Journal 14(1), 133–141 (2012)
15. Kinnings, S.L., Liu, N., Tonge, P.J., Jackson, R.M., Xie, L., Bourne, P.E.: A Machine Learning-Based Method to Improve Docking Scoring Functions and its Application to Drug Repurposing. J. Chem. Inf. Model. 51, 408–419 (2011)
16. Das, S., Krein, M.P., Breneman, C.M.: Binding Affinity Prediction with Property-Encoded Shape Distribution Signatures. J. Chem. Inf. Model. 50, 298–308 (2010)
17. Li, L., Wang, B., Meroueh, S.O.: Support Vector Regression Scoring of Receptor-Ligand Complexes for Rank-Ordering and Virtual Screening of Chemical Libraries. J. Chem. Inf. Model. 51, 2132–2138 (2011)
18. Durrant, J.D., McCammon, J.A.: NNScore 2.0: A Neural-Network Receptor–Ligand Scoring Function. J. Chem. Inf. Model. 51(11), 2897–2903 (2011)
19. Breiman, L.: Random Forests. Mach. Learn. 45, 5–32 (2001)
20. Vapnik, V.: The nature of statistical learning theory. Springer, New York (1995)
21. Amini, A., et al.: A general approach for developing system-specific functions to score protein-ligand docked complexes using support vector inductive logic programming. Proteins 69, 823–831 (2007)
22. Breiman, L., et al.: Classification and regression trees. Chapman & Hall/CRC (1984)
23. Cheng, T., Li, X., Li, Y., Liu, Z., Wang, R.: Comparative Assessment of Scoring Functions on a Diverse Test Set. J. Chem. Inf. Model. 49, 1079–1093 (2009)
24. Rucker, C., Rucker, G., Meringer, M.: y-Randomization and its variants in QSPR/QSAR. J. Chem. Inf. Model. 47, 2345–2357 (2007)
25. The Comprehensive R Archive Network (CRAN) Package e1071, `http://cran.r-project.org/web/packages/e1071/index.html` (last accessed November 2, 2011).
26. Sotriffer, C.A., Sanschagrin, P., Matter, H., Klebe, G.: SFCscore: scoring functions for affinity prediction of protein-ligand complexes. Proteins 73, 395–419 (2008)
27. Zsoldos, Z., Reid, D., Simon, A., Sadjad, S.B., Johnson, A.P.: eHiTS: a new fast, exhaustive flexible ligand docking system. J. Mol. Graph. Model. 26, 198–212 (2007)
28. Joachims, T.: Making large-Scale SVM Learning Practical. In: Schölkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods - Support Vector Learning. MIT Press (1999)
29. Kirkpatrick, S.C., Gelatt, D., Vecchi, M.P.: Optimization by simulated annealing. Science 220, 671–680 (1983)
30. LIBSVM - A Library for Support Vector Machines,
`http://www.csie.ntu.edu.tw/~cjlin/libsvm/`
(last accessed November 2, 2011).
31. CSAR, `http://www.csardock.org` (last accessed November 2, 2011).
32. The PDBbind database, `http://www.pdbbind-cn.org/` (last accessed November 2, 2011).
33. Berman, H.M., et al.: The Protein Data Bank. Nucleic Acids Res. 28, 235–242 (2000)
34. The Comprehensive R Archive Network (CRAN) Package caret, `http://cran.r-project.org/web/packages/caret/index.html` (last accessed November 2, 2011).