Raulia R. Syrlybaeva* and Marat R. Talipov*

CBSF: A New Empirical Scoring Function for Docking Parameterized by Weights of Neural Network

https://doi.org/10.1515/cmb-2019-0009
Received October 7, 2019; accepted December 4, 2019

Abstract: A new CBSF empirical scoring function for the estimation of binding energies between proteins and small molecules is proposed in this report. The final score is obtained as a sum of three energy terms calculated using descriptors based on a simple counting of the interacting protein-ligand atomic pairs. All the required weighting coefficients for this method were derived from a pretrained neural network. The proposed method demonstrates a high accuracy and reproduces binding energies of protein-ligand complexes from the CASF-2016 test set with a standard deviation of 2.063 kcal/mol (1.511 log units) and an average error of 1.682 kcal/mol (1.232 log units). Thus, CBSF has a significant potential for the development of rapid and accurate estimates of the protein-ligand interaction energies.

Keywords: Molecular docking, Scoring function, Machine learning

MSC: 92C40, 92E10

1 Introduction

Remarkable progress in the field of artificial intelligence and increasing availability of high-quality reference data have resulted in a rapid development of protein-ligand interaction scoring functions(Huang, Grinter, & Zou, 2010; Nguyen, Zhou, & Minh, 2018; Yadava, 2018) using machine learning algorithms(Chen, Engkvist, Wang, Olivecrona, & Blaschke, 2018) such as vector support machines or neural networks. Neural networks designed for the prediction of binding energies between receptors and ligands are typically based on the pattern recognition and computer vision ideas and have deep architecture utilizing 2D- or 3D-convolution(Gomes, Ramsundar, Feinberg, & Pande, 2017; Gonczarek et al., 2018; Ragoza, Hochuli, Idrobo, Sunseri, & Koes, 2017; Stepniewska-Dziubinska, Zielenkiewicz, & Siedlecki, 2018; Sunseri, King, Francoeur, & Koes, 2019) or graph-convolution(Feinberg et al., 2018; Lim, Ryu, Park, Choe, & Ham, 2019; Torng & Altman, 2018) approaches. Accordingly, these models produce results by detecting nonlinear dependencies which are hard to express in a functional form.

At the same time, machine learning techniques are able to find connections between input and output data that can be explicitly represented by a simple functional form, e.g. linear correlation (Fracchia, Frate, Mancini, Rocchia, & Barone, 2018; Sander, 2014). The weights of these trained models may be directly used as constant coefficients in the corresponding calculations if the data providing as variables is of the same type as those used for the model training. In this regard, it is appealing to design a neural network for predicting protein-ligand binding energies, the trained version of which amounts to a practically usable mathematical formula. Such approach possesses important advantages: the resulted expression is essentially an empirical scoring function, and empirical scoring functions are very fast and allow a rapid screening of a vast array

***Corresponding Author: Raulia R. Syrlybaeva:** Department of Chemistry and Biochemistry, New Mexico State University, Las Cruces, New Mexico 88003, United States
Current address: College of Pharmacy, University of Georgia, Athens, Georgia 30602, United States, E-mail: raulia@mail.ru
***Corresponding Author: Marat R. Talipov:** Department of Chemistry and Biochemistry, New Mexico State University, Las Cruces, New Mexico 88003, United States, E-mail: talipovm@nmsu.edu

of complexes(Huang et al., 2010), and usage of the neural network for producing of optimal constants for empirical soring functions holds potential for the development of highly accurate models.

In this paper we propose a new empirical scoring function, which is based on counting the protein-ligand interacting atom pairs as descriptors and contains certain weights retrieved from pre-trained neural networks as constants. While the descriptors defined by the atom pair interactions are not uncommon in the field(Ballester & Mitchell, 2010; Durrant, Friedman, Rogers, & McCammon, 2013; Durrant & Mccammon, 2011; Durrant & McCammon, 2010; Sotriffer, Sanschagrin, Matter, & Klebe, 2008; Wójcikowski, Ballester, & Siedlecki, 2017) because of their simplicity, they also have been criticized for their insensitiveness to the ligand poses and to the protonation states of the molecules(Gabel, Desaphy, & Rognan, 2014). Distance dependent terms and atom types as in Open Babel(Boyle et al., 2011) were incorporated into the model to address these deficiencies.

Several probe scoring functions were tested on 285 complexes of CASF-2016 benchmark set(Su et al., 2019), and the most accurate scoring function named CBSF reproduces the reference binding energies with a standard deviation of 2.063 kcal/mol (1.511 log units) and Pearson correlation coefficient (R) of 0.718. Therefore, the accuracy of our scoring function is comparable or higher than for most other known methods, both classical (empirical, force-field and knowledge-based, $R \sim$ 0.4-0.7(Y. Li et al., 2014; Su et al., 2019)) and machine-learning based ($R \sim$ 0.6-0.8)(H. Li et al., 2018; Y. Li et al., 2014; Su et al., 2019). We believe that CBSF might find use in molecular docking software for this reason. The implemented approach to empirical scoring function based on the neural network weights might also be of interest to further developments in the field.

2 Methods

2.1 Datasets and Preprocessing of Data

Structures of protein-ligand binding complexes and their corresponding binding affinities were obtained from PDBBind database version 2018(Liu et al., 2015, 2017). A total of 4128 complexes were chosen from the "refined" subset for the training set of the neural networks while the CASF-2016 core set(Y. Li et al., 2014; Su et al., 2019) consisting of 285 complexes was used to test and evaluate the performance of the neural networks and the corresponding scoring functions. The criteria of the selection of the complexes for the training set is provided below.

The cartesian coordinates of the protein atoms were extracted from the PDB files containing the structure of binding pockets, all water molecules were removed. The coordinates of the ligand atoms were read from the SDF files. The protein-ligand atom pairs and total numbers of atoms in ligands were used as descriptors for the scoring functions and the neural networks. The protein-ligand atom pair was initially defined as two atoms within 5 Å and such atom pairs were fed into the neural networks. At the same time, the neural networks were just the supplementary tools in the development of the scoring functions and one of their main tasks was to determine better individual cutoffs for different types of atom pairs. Thereby the selection of atom pairs for the scoring functions was processed according to the cutoffs produced by means of the neural networks, these values can be found among the project's files available at https://github.com/rsyrlyb/CBSF. We did not take into account hydrogen atoms except those connected to oxygen atoms or nitrogen atoms of amino groups. Atom types were assigned by the Open Babel software (Boyle et al., 2011). The atom pair labels were defined by concatenating the atomic types of the contributing atoms sorted alphabetically with the semicolon symbol used as a separator, e.g. 'C+;C3'.

The input data for the neural networks was generated as follows. The descriptors collected from a single protein-ligand complex were initially organized in the matrix form (Figure 1), where each column contains all protein-ligand distances for a certain protein-ligand atom pair type. Complexes that are not matching with the complexes of the test set and with no more than 130 atom pairs of the same type were used for the model training (that is, the input matrices had no more than 130 rows), which resulted in the final training set of 4128

complexes. The resulting matrices were broadcasted to the same shape and stacked into a three-dimensional table (number of samples × number of atom pair types (605) × 130). Further, atom pair types encountered in molecules no more than 10 times were separated into additional matrix of lower dimensionality (number of samples × number of rare atom pair types (474) × 10) for faster training of the neural networks. The obtained distances were normalized by dividing their values by 5. In addition to this 3D structures, a matrix containing the total numbers of atoms in ligands and target values of the binding energies (in kcal/mol) was used in the training of the neural networks.

		C3;O.co2	C3;O2	C2;Car	C1;Car	....
		2.7	4.6	2.6	2.7	....
≤ 130 rows	{	2.8	100.0	4.5	100.0	....
		....	....	....	....	....
		Empty cells filled by 100.0				

Figure 1: Example of a matrix prepared as input for the neural network and containing information about atom pairs in a single complex.

2.2 Scoring Function

The proposed scoring function is based on three terms: (1) the number of atom-atom pairwise interactions selected within a certain cutoff distance, (Figure 2); (2) number of interactions per atom of the ligand (further denoted as the density of interactions), and (3) correction that takes into account the atomic pair types. The overall free energy change according to our scheme is expressed as:

$$\Delta G_{bind} = E_1 + E_2 + E_3 \tag{1}$$

where ΔG_{bind} is the predicted binding free energy and $E_1 - E_3$ are the terms introduced above.

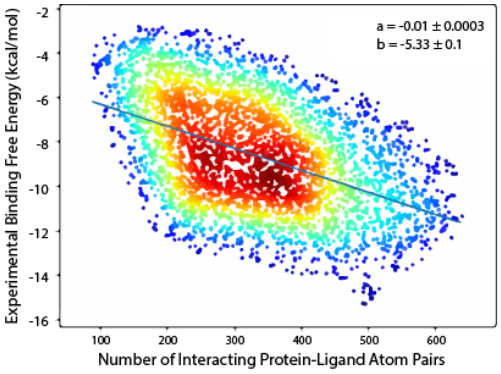

Figure 2: Dependence of the experimental binding free energy on the number of protein-ligand atom pairs selected within 5 Å cutoff, the outliers were removed using the elliptic envelope approach. Colors in the plots represent the density of the points. A 95% confidence interval for the slope and intersection are presented in the figure.

Calculation of E_1 . The term E_1 is based on the linear dependence between the number of protein-ligand atom pairs and the binding free energy. The scoring function ΔG is quite sensitive to this descriptor, and thus the distance cutoffs used for selecting the atom pairs are very important. The individual cutoffs for each atom pair type (605 in total) were determined using the neural network and applied in the scoring function. In a

basic version of the scoring function:

$$E_1 = a_1 N_p + b_1 \tag{2}$$

$$N_p = \sum_{A \in L} \sum_{B \in P} H[R^0(T_A; T_B) - R_{AB}] \tag{3}$$

$$H[x] = \begin{cases} 0, & x < 0, \\ 1, & x \geq 0 \end{cases} \tag{4}$$

where a_1 and b_1 are coefficients of the linear equation, N_p is the number of atom pairs, A and B are the atoms at the ligand (L)-protein (P) interface, R_{AB} is the interatomic distance between atoms A and B, $R^0(T_A; T_B)$ is the distance cutoff for the atom types of A (i.e. T_A) and B (i.e. T_B).

As a further modification of the method, the protein-ligand interatomic distances were divided into a certain number of intervals (K) and the pairs of (a_1, b_1) parameters were assigned individually to each interval. The atom pairs from the intervals sharing the same sequential index are processed using the same (a_1, b_1) parameters. The final equation is:

$$E_1 = \sum_{i=1}^{K} \left(a_{1,i} N_{p,i} + b_{1,i} \right) \tag{5}$$

$$N_{p,i} = \sum_{A \in L} \sum_{B \in P} \left\{ H \left[R_i^0 (T_A; T_B) - R_{AB} \right] - H[R_{i-1}^0 (T_A; T_B) - R_{AB}] \right\} \tag{6}$$

where K is the number of intervals, $N_{p,i}$ is the total number of atom pairs from the i-th interval, $R_i^0(T_A; T_B)$ is the upper distance cutoff for the i-th interval for the atom types T_A and T_B, [note that $R_0^0 (T_A; T_B) = 0$], and ($a_{1,i}$, $b_{1,i}$) are the constants (a_1, b_1) assigned to the intervals i.

The upper limits of the intervals with the sequential number i make up an array of distances $d_i = (d_{i1}, d_{i2}, \ldots, d_{i605})$, the lower limits are determined by the array of the preceding intervals *i-1*, the first intervals have no lower limits. The exact same intervals with their characteristic arrays were applied for calculations of the rest two terms, as well. We explored the performance of several models that differ only in the numbers of used intervals (K = 1-4), which will be referred to as the basic scoring function (SF1), SF2, SF3 and CBSF (SF4), respectively.

Calculation of E_2. The term E_2 is taking into account the buriedness of ligand as the ratio of atom pairs to the total number atoms on the ligand. This term is calculated in a similar manner as the previous term:

$$E_2 = \sum_{i=1}^{K} a_{2,i} D_i \tag{7}$$

where $a_{2,i}$ are the fitting constants assigned to the i-th intervals and D_i is the density of interactions calculated according to formula:

$$D_i = \frac{N_{p,i}}{N_L} \tag{8}$$

where N_L is the total number of atoms in the ligand.

Equation 7 echoes the approach reported in (Spitzer, Cleves, Varela, & Jain, 2014), where the buriedness was measured by taking the ratio of near-ligand protein atoms to the total number of heavy atoms on the ligand.

Calculation of E_3. The term E_3 is calculated as a sum of contributions from each atom pair in the complex depending on their types:

$$E_3 = \sum_{i=1}^{K} \sum_{j=1}^{605} n_{ij} w_{ij} \tag{9}$$

where n_{ij} is the number of atom pairs of j-th type in the i-th interval, w_{ij} is the distance-dependent energy contribution assigned to the j-th atom pair in the i-th interval.

All required constants of the scoring functions, such as d_i, $(a_{1,i}, b_{1,i})$, $a_{2,i}$, and w_{ij}, were determined by training of the convergent neural networks.

2.3 Architecture of the Neural Networks

Several neural networks were developed to generate the scoring functions with different K value. Training and the testing of the neural networks were performed using Keras(Chollet & others, 2015) with Tensorflow(Abadi et al., 2016) as the back-end using in-house software implemented in Python 3.7 programming language.

Neural network designed for the basic scoring function

The architecture of the neural network for obtaining of the parameters of the scoring function with one interval of distances ($K = 1$) is presented on Figure 3. The neural network consists of two blocks, preliminary and main. The preliminary block contains an input layer, followed by a noise layer adding Gaussian noise with a standard deviation of 0.008 to the input data to prevent overfitting. The level of noise has been chosen by testing the neural network using different values of standard deviation of the noise distribution based on comparison of performances of the models. The main block consists of layers 2-5 described below.

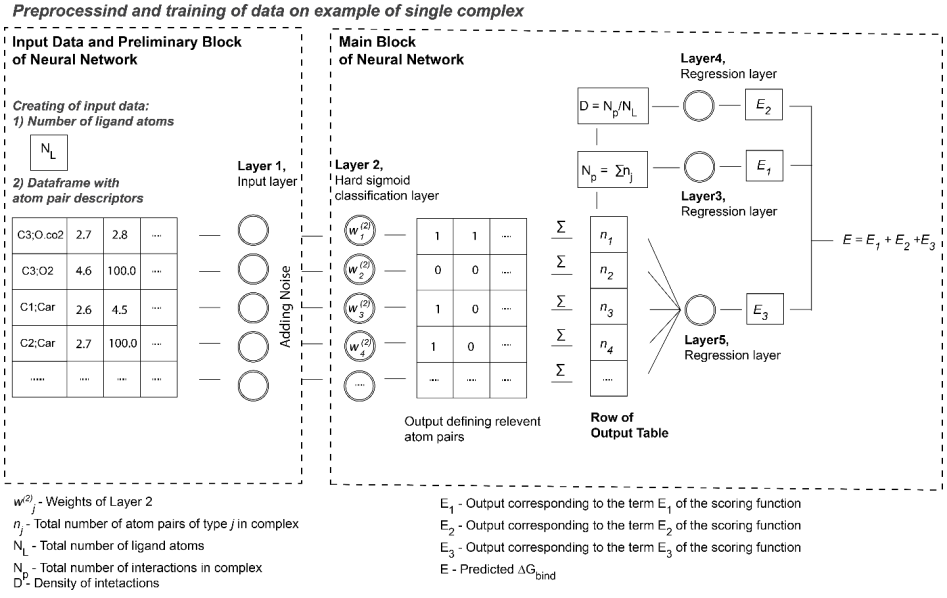

Figure 3: Architecture of the neural network for the basic scoring function. The data frames are shown in a transposed form for convenience.

Layer 2 determines the relevant cutoffs for the atom pair types. The layer was trained to find a probability whether the distances between atoms are close enough to be considered. The output of this custom layer with custom hard sigmoid activation function F:

$$y = F(-R_{AB} + w_j^{(2)}) \tag{10}$$

where $w_j^{(2)}$ is the weight of layer 2 assigned to the corresponding column of the input data frame. The hard sigmoid function in this study has a transition zone of 0.1 Å, according to the features of the activation function:

$$y(x) = \begin{cases} 0, & x \le -0.05 \\ 10x + 0.5, & -0.05 < x \le 0.05 \\ 1, & x > 0.05 \end{cases} \quad (11)$$

Therefore, the weights $w_j^{(2)}$ are the maximum possible distances between the atoms making up the atom pair. These values multiplied by 5 (denormalization) form the array of upper limits of distances d discussed earlier.

The output values are summarized for counting of the total numbers of atom pairs of different types n_j in the complex. These values are collected into a single table with rows corresponding to separate complexes. The total number of atom pairs N_p in the complex is found as a sum of elements of the row. This value is passed to the regression layer 3 calculating the term E_1. Besides, N_p is used for the calculation of the density of interactions D which is the input for the regression layer 4 designed for the obtaining of the term E_2. Both trainable weight and biase were used in the layer 3 and trainable weight in layer 4, the weight and the bias of the layer 3 are relevant to the constants (a_1, b_1) of the scoring function (eq. 2) and the weight of the layer 4 is $a_{2,1}$ constant (eq. 7).

Parallelly, the term E_3 taking into account the types of atom pairs is calculated by the regression layer 5. As distinguished from the layers 3-4, the layer 5 takes multiple values as inputs and does not have a bias. The weights of the layer 5 form increments w_{1j} assigned to the atom pair types (eq. 9). The last operation of the neural network is summarizing of the outputs of the layers 3-5.

Neural networks for scoring functions with several cutoff distance intervals

Neural networks for the scoring functions with more than one intervals of distances consist of the same blocks as in the basic model, but the number of the main blocks is equal to the number of intervals and the final predicted energy is the sum of the outputs of all these blocks (Figure 4).

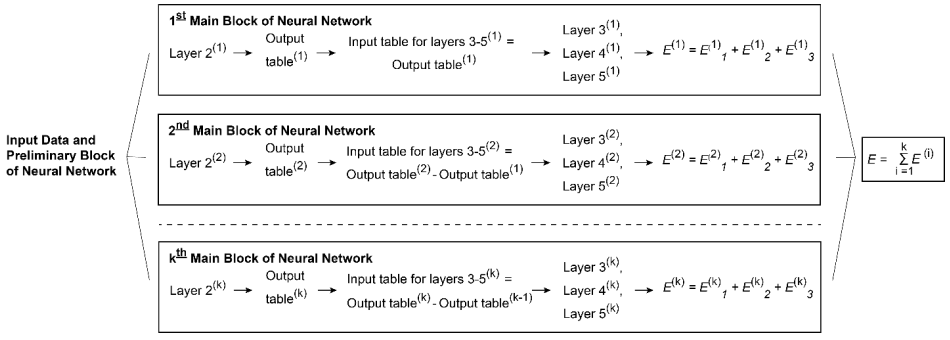

Figure 4: Architecture of the neural networks with K intervals of distances.

Another difference is that the layers 3-5 of the main blocks process the atom pairs belonging to the corresponding distance intervals only. This was organized in the following manner: the layers 2 define the upper limits d of the ranges (as described in the basic model) and select all atom pairs within the interval (0, d). Output tables of the layers 2 of two subsequent main blocks *i-1* and *i* contain the atom pairs from ranges (0, d_{i-1}) and (0, d_i) respectively, and therefore, the atom pairs belonging to the interval (d_{i-1}, d_i) which have to be processed by the layers 3-5 of the main block *i* are found by the pointwise subtraction of the tables. Neural networks consist of 2-4 main blocks are denoted as NN-SF2, NN-SF3 and NN-SF4 further.

2.4 Training of the neural networks

Initial weights of the neural networks were set as follows:

1) Minimal and maximal observed distances for each type of atom pair were retrieved from the preprocessed data. The obtained ranges of distances were normalized and divided into K equal intervals, the upper boundaries of these intervals were used as initial weights of the layers 2 of the main blocks.

1. Coefficients (a_1, b_1) initially estimated using SciPy library and presented in Figure 2, were used as the weight and bias of the layer 3 of the main blocks.
2. Weights of the layers 5 were set to zero.

The trainings were carried out in three stages: the weights of the layers 3-4 were optimized in the first step, the other weights were frozen. The weights of the layers 5 were unfrozen in the second step, and optimizations of all weights, including the weights of the layers 2, were provided in the step three. K-fold cross-validation approach was used for each step of the training with k=9. The numbers of epochs for each round of learning were 50, 50 and 40 in the base model and 20, 20, 15 in the models with a few intervals for the stages 1-3 respectively.The Adam optimizer was used to minimize the mean squared error during optimization.

2.5 Evaluation methods

Mean absolute error (MAE) and median error (median) were calculated using corresponding functions of NumPy (van der Walt, 2011). Estimation of scoring power was carried out by calculating of Pearson's correlation coefficient R and Standard deviation in fitting (σ). Ranking power is presented by means of Spearman correlation coefficient (SP), Kendall correlation coefficient (τ) and predictive index (PI). Ready scripts for these calculations were taken from CASF-2016 benchmark(Su et al., 2019).

3 Results and Discussion

3.1 Comparative analysis of the composition of the suggested scoring functions

Our method is somewhat similar to the scoring functions based on counting of protein-ligand atom pairs, e.g. NNScore(Durrant et al., 2013; Durrant & Mccammon, 2011; Durrant & McCammon, 2010), SFCscore(Sotriffer et al., 2008) and RF-Score(Ballester & Mitchell, 2010; Wójcikowski et al., 2017). A brief review of such methods can be found in (Guedes, Pereira, & Dardenne, 2018). Furthermore, some of these scoring functions, such as RF-Score and NNScore, were developed by means of the machine-learning techniques. However, there are significant differences between these methods and our approach. In this section, we will discuss the main ideas that distinguish our scoring function from the previous works.

The main difference is related to the cutoffs for selecting atom pairs. Typically, a single cutoff is used, which is independent of the identity of interacting atoms. This approach could be augmented by introducing several equally separated atom type-independent cutoffs that define distance intervals with different free energy contributions. Types of the atom pairs tailed within the determined ranges of distances have a direct crucial impact to the resulting output of the scoring function. Herein, we use an opposite approach, in which distance cutoffs were dependent on the type of atom pair, but it was assumed that all types of atom pairs make approximately the same contribution to the predicted energy value. The resulting distances are divided into K ranges in the K-interval versions of the scoring functions, but again, the interval breaks are individual for each type of atom pair. For example, in the scoring function SF5, the cutoff breaks of 2.55 Å, 3.69 Å, 4.61 Å were used for the 'C3;C3' atom pair and 2.76 Å, 3.75 Å, 4.67 Å for 'C3;Car'. Maximum possible equality of the energy contributions of the atom pairs from the intervals with the same sequential number was requested as the main goal during the choice of the borders of the ranges in our model training. Therefore, equations 2 and 3 for the calculation of terms E_1 and E_2 take the total number of all atom pairs from the intervals with the same number as the variables, and the constants ($a_{1,i}$, $b_{1,i}$) or $a_{2,i}$ were independent of the types of atom pairs.

The types of atom pairs are directly included into the scoring function via the term E_3. This term echoes some of the features of the other methods mentioned before, but in our model this term is estimated as a simple linear function, while similar descriptors are processed using nonlinear operations in the related machine-learning based scoring functions. The term E_3, to a greater extent, was inspired by the knowledge-based scoring functions(Dittrich, Schmidt, Pfleger, & Gohlke, 2019; Mirzaie & Sadeghi, 2010; Zheng et al., 2011) in which the distance-dependent potentials are evaluated from continuous function based on an inverse Boltzmann approach. The difference from these potentials, besides the method of their retrieving, is that the distance-dependent increments w_{ij} from our method are assigned to the intervals of distances. Another major difference is that the values of the most increments w_{ij} are close to zero, since, as it was indicated earlier, the scoring functions were designed so that the term E_3 has low weight. Figure SI1 shows the distribution of the values of w_{ij} in the basic scoring function, the median and the standard deviation are -0.035 and 0.69 kcal/mol, respectively.

It should be noted here that both term E_1 and term E_3 are taking into account the impact of the atom pairs on the binding energy. The difference is that E_1 accounts this factor using average value of the potential assigned to all types of atom pairs – a_1, meanwhile w_j increments are essentially deviations of the full potentials of particular atom pair types from that average impact. Therefore, these terms can be merged as follows:

$$E_1 + E_3 = \sum_{i=1}^{K}\sum_{j=1}^{605}(n_{ij} * a_{1,i} + b_{1,i}) + \sum_{i=1}^{K}\sum_{j=1}^{605} n_{ij}w_{ij} =$$

$$= \sum_{i=1}^{K}\sum_{j=1}^{605}(n_{ij} * (a_{1,i} + w_{ij}) + b_{1,i}) = \sum_{i=1}^{K}\sum_{j=1}^{605}(n_{ij} * W_{ij} + b_{1,i}) \quad (12) \tag{12}$$

where $W_{ij} = a_{1,i} + w_{ij}$.

On the other side, introducing of two separate terms improves accuracy of the neural network: this effect likely has the same origin as the effectiveness of the batch normalization implying normalizing of each neuron into zero mean and unit variance, since the median for w_{ij} values is zero and $|w_{ij}|<|W_{ij}|$ in general. Besides, such splitting can increase the speed of the calculations since the atom pair types with low w_{ij} can be excluded from calculations of the term E_3.

An appealing feature of our scoring functions is their simplicity, as they are not overloaded with supplementary terms present in other empirical scoring functions (counting hydrogen bonds and rotational bonds, taking into account partial charges or electrostatic potentials and so on(Baek, Shin, Chung, & Seok, 2017; Guedes et al., 2018; Jain, 1996; R. Wang, Lai, & Wang, 2002)) and do not contain terms from third-party scoring functions (Pereira, Caffarena, & Dos Santos, 2016; Tanchuk, Tanin, Vovk, & Poda, 2016; C. Wang & Zhang, 2017). The term E_2, reflecting the buriedness of ligands(Oprea & Marshall, 2005; R. Wang et al., 2002), is the only conventional empirical term. But this factor is measured in a simple way: as the linear function of the ratio of the total number of atom pairs to the number of ligand atoms. High values of the term E_2 are characteristic for ligands deeply buried in the protein structure.

The terms E_1 are the main contributors in the scoring functions, as they accounted for 60-70% of the overall binding energy (Table S1). The terms E_2 and E_3 have approximately same values of 15-30%. There is no noteworthy difference in the proportions of the developed scoring functions with multiple intervals, the influence of the terms are close in all models. The second term of the base model, unlike in other models, has low contribution – 3%.

Due to the regression layers 3-5, our models can be presented as simple mathematical expressions, unlike other scoring functions with hidden knowledge from the machine learning techniques. This simplicity makes the calculations using the proposed scoring function more transparent, convenient, and robust, because there is no need in loading of the neural network requiring the strict preprocessing (preparing of the input data of 2D dimensionality for each complex, in our case) and additional neural-network libraries.

As to drawbacks of the model, they are similar with problems of other methods based on counting of atom pairs. Structural data of high quality is required to ensure high accuracy of the method during both training and usage of the method and only 13% of complexes of the training set have resolution better that 1.5

Å currently. Besides, 161 atom pair types are represented by less than10 instances, 72 of them have interatomic distances ranging within 0.5 Å – it is hard or impossible to determine proper values of cutoffs and increments in such cases.

3.2 Performance of the scoring functions

Performance of developed scoring functions on the test set complexes comprising mean absolute error (MAE), median error, Pearson's correlation coefficient R, standard deviation (σ), Spearman correlation coefficient (SP), Kendall correlation coefficient (τ) and predictive index (PI) are presented in Figure 5. These results were compared with the assessment data on the known scoring functions reported in (Su et al., 2019) and obtained using the same benchmark.

The scoring power of all developed scoring functions is higher than for the most known scoring functions. So, even the basic version of the scoring function with a single set of distance cutoffs predicts the binding energies with the standard deviation of 2.129 kcal/mol and Pearson's correlation coefficient of 0.698, and these results are more accurate than that of AutoDock Vina, GlideScore, DrugScore scoring functions and other well-established methods (Table SI2). The scoring ability of the method improves with increased number of intervals, but quickly reaches a limit due to increasing of the number of parameters for optimization: each new interval adds 2 * number of considered atom pair types + 3 variables. Excess of the number of variables over training samples leads to tremendous overfitting. So, the difference between the standard deviations of the basic model and the best SF4 scoring function is 0.066 kcal/mol only, their performances are very close, and in fact, the ranging power of SF1 is slightly better.

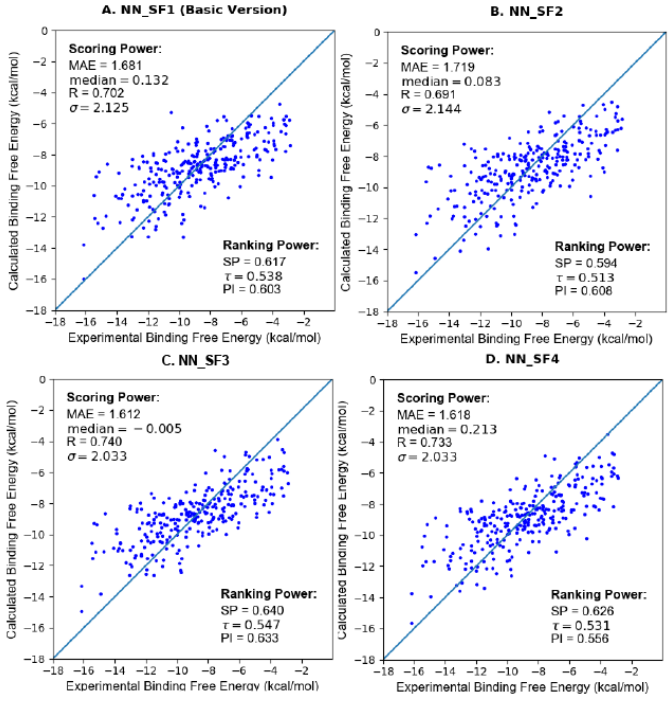

Figure 5: Accuracy of the scoring functions tested on CASF-2016 benchmark

The observed saturation of the model quality with increased number of cutoff intervals was the reason for stopping at four intervals of distances for counting of the atom pairs. The resulting scoring function was named CBSF (contacts-based scoring function) and is considered by us as the main version of the presented scoring functions. The numerical assessment of the scoring power of CBSF (σ = 2.063 kcal/mol, R=0.718,

SP=0.605) indicates a high accuracy of the method, and only $\Delta_{vina}RF_{20}$scoring function(Su et al., 2019; C. Wang & Zhang, 2017)shows better scoring power. Considering the simplicity and accuracy of the scheme together with its straightforward implementation, CBSF definitely overcomes the main shortcomings characteristic for other schemes based on counting of the number of atom pairs and therefore represents a great interest for practical use in relevant chemical software.

The performances of the convergent neural networks applied for the development of the scoring functions are presented in Figure SI2. The accuracy of the neural network corresponding to CBSF (σ = 1.618 kcal/mol, R=0.733, SP=0.626) is noticeably higher than that of CBSF, but the differences are not critical. This discrepancy is due to the features of the hard sigmoid activation function (eq. 11) used in the neural network. The best predicted binding energies are obtained using NN-SF3 model (σ = 1.612 kcal/mol, R=0.740, SP=0.640).

CBSF particularly can be implemented in docking programs or to be applied for rescoring of poses generated by other docking packages displaying lower accuracy in the scoring. The parameters of the scoring functions, such as the cutoffs or the distance dependent increments can be utilized in the searching algorithms. Besides, the approach used for developing of the suggested scoring functions is generally universal and can find a use in further experiments in the field.

4 Conclusion

The new empirical scoring function named CBSF for estimating of the binding affinity of protein-ligand complexes with known three-dimensional structure is developed. CBSF outperforms most of the known scoring functions, the standard deviation obtained during its testing on CASF-2016 is 2.063 kcal/mol, while Pearson's correlation coefficient is 0.713. All parameters and coefficients of the scoring function are found by means of the neural network; this approach allowed to make considerable simplifications in the scoring function without harm to its accuracy.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., … Zheng, X. (2016). TensorFlow: A System for Large-Scale Machine Learning. In *12th USENIX Symposium on Operating Systems Design and Implementation* (pp. 266–284). Savannah: Berkeley: USENIX Association.

Baek, M., Shin, W. H., Chung, H. W., & Seok, C. (2017). GalaxyDock BP2 score: a hybrid scoring function for accurate protein–ligand docking. *Journal of Computer-Aided Molecular Design*, *31*(7), 653–666. https://doi.org/10.1007/s10822-017-0030-9

Ballester, P. J., & Mitchell, J. B. O. (2010). A machine learning approach to predicting protein-ligand binding affinity with applications to molecular docking. *Bioinformatics*, *26*(9), 1169–1175. https://doi.org/10.1093/bioinformatics/btq112

Boyle, N. M. O., Banck, M., James, C. A., Morley, C., Vandermeersch, T., & Hutchison, G. R. (2011). Open Babel: An open chemical toolbox. *Journal of Cheminformatics*, *3*(33), 1–14. https://doi.org/10.1186/1758-2946-3-33

Chen, H., Engkvist, O., Wang, Y., Olivecrona, M., & Blaschke, T. (2018). The rise of deep learning in drug discovery. *Drug Discovery Today*, *23*(6), 1241–1250. https://doi.org/10.1016/j.drudis.2018.01.039

Chollet, F., & others. (2015). *Keras*. Retrieved from https://github.com/fchollet/keras

Dittrich, J., Schmidt, D., Pfleger, C., & Gohlke, H. (2019). Converging a Knowledge-Based Scoring Function: DrugScore 2018. *Journal of Chemical Information and Modeling*, *59*(1), 509–521. https://doi.org/10.1021/acs.jcim.8b00582

Durrant, J. D., Friedman, A. J., Rogers, K. E., & McCammon, J. A. (2013). Comparing neural-network scoring functions and the state of the art: Applications to common library screening. *Journal of Chemical Information and Modeling*, *53*(7), 1726–1735. https://doi.org/10.1021/ci400042y

Durrant, J. D., & Mccammon, J. A. (2011). NNScore 2.0: A Neural-Network Receptor-Ligand Scoring Function. *Journal of Chemical Information and Modeling*, *51*, 2897–2903.

Durrant, J. D., & McCammon, J. A. (2010). NNScore: A neural-network-based scoring function for the characterization of protein-ligand complexes. *Journal of Chemical Information and Modeling*, *50*(10), 1865–1871. https://doi.org/10.1021/ci100244v

Feinberg, E. N., Sur, D., Wu, Z., Husic, B. E., Mai, H., Li, Y.,… Pande, V. S. (2018). PotentialNet for Molecular Property Prediction. *ACS Central Science*, 4(11), 1520-1530. https://pubs.acs.org/doi/full/10.1021/acscentsci.8b00507

Fracchia, F., Frate, G. Del, Mancini, G., Rocchia, W., & Barone, V. (2018). Force Field Parametrization of Metal Ions from Statistical Learning Techniques. *Journal of Chemical Theory and Computation*, *14*, 255–273. https://doi.org/10.1021/acs.jctc.7b00779

Gabel, J., Desaphy, J., & Rognan, D. (2014). Beware of Machine Learning-Based Scoring Functions - On the Danger of Developing Black Boxes. *Journal of Chemical Information and Modeling*, *54*, 2807-2815. https://doi.org/10.1021/ci500406k

Gomes, J., Ramsundar, B., Feinberg, E. N., & Pande, V. S. (2017). *Atomic Convolutional Networks for Predicting Protein-Ligand Binding Affinity*. 1–17. Retrieved from http://arxiv.org/abs/1703.10603

Gonczarek, A., Tomczak, J. M., Zaręba, S., Kaczmar, J., Dąbrowski, P., & Walczak, M. J. (2018). Interaction prediction in structure-based virtual screening using deep learning. *Computers in Biology and Medicine*, *100*, 253–258. https://doi.org/10.1016/j.compbiomed.2017.09.007

Guedes, I. A., Pereira, F. S. S., & Dardenne, L. E. (2018). Empirical scoring functions for structure-based virtual screening: Applications, critical aspects, and challenges. *Frontiers in Pharmacology*, *9*(SEP), 1–18. https://doi.org/10.3389/fphar.2018.01089

Huang, S. Y., Grinter, S. Z., & Zou, X. (2010). Scoring functions and their evaluation methods for protein-ligand docking: Recent advances and future directions. *Physical Chemistry Chemical Physics*, *12*(40), 12899–12908. https://doi.org/10.1039/c0cp00151a

Jain, A. N. (1996). Scoring noncovalent protein-ligand interactions: A continuous differentiable function tuned to compute binding affinities. *Journal of Computer-Aided Molecular Design*, *10*(5), 427–440. https://doi.org/10.1007/BF00124474

Li, H., Peng, J., Leung, Y., Leung, K. S., Wong, M. H., Lu, G., & Ballester, P. J. (2018). The impact of protein structure and sequence similarity on the accuracy of machine-learning scoring functions for binding affinity prediction. *Biomolecules*, *8*(1). https://doi.org/10.3390/biom8010012

Li, Y., Han, L., Liu, Z., Wang, R., Li, J., Han, L., … Wang, R. (2014). Comparative assessment of scoring functions on an updated benchmark: 1. Compilation of the Test Set. *Journal of Chemical Information and Modeling*, *54*(6), 1700–1716. https://doi.org/10.1021/ci500081m

Lim, J., Ryu, S., Park, K., Choe, Y. J., & Ham, J. (2019.). Predicting drug-target interaction using 3D structure-embedded graph representations from graph neural networks. 1–20. https://arxiv.org/abs/1904.08144

Liu, Z., Li, Y., Han, L., Li, J., Liu, J., Zhao, Z., … Wang, R. (2015). PDB-wide collection of binding data: Current status of the PDBbind database. *Bioinformatics*, *31*(3), 405–412. https://doi.org/10.1093/bioinformatics/btu626

Liu, Z., Su, M., Han, L., Liu, J., Yang, Q., Li, Y., & Wang, R. (2017). Forging the Basis for Developing Protein-Ligand Interaction Scoring Functions. *Accounts of Chemical Research*, *50*(2), 302–309. https://doi.org/10.1021/acs.accounts.6b00491

Mirzaie, M., & Sadeghi, M. (2010). Knowledge-based potentials in protein fold recognition. *Journal of Paramedical Sciences*, *1*(4), 65–75.

Nguyen, T. H., Zhou, H. X., & Minh, D. D. L. (2018). Using the fast fourier transform in binding free energy calculations. *Journal of Computational Chemistry*, *39*(11), 621–636. https://doi.org/10.1002/jcc.25139

Oprea, T. I., & Marshall, G. R. (2005). Receptor-Based Prediction of Binding Affinities. *3D QSAR in Drug Design*, 35–61. https://doi.org/10.1007/0-306-46857-3_3

Pereira, J. C., Caffarena, E. R., & Dos Santos, C. N. (2016). Boosting Docking-Based Virtual Screening with Deep Learning. *Journal of Chemical Information and Modeling*, *56*(12), 2495–2506. https://doi.org/10.1021/acs.jcim.6b00355

Ragoza, M., Hochuli, J., Idrobo, E., Sunseri, J., & Koes, D. R. (2017). Protein - Ligand Scoring with Convolutional Neural Networks. *Journal of Chemical Information and Modeling*, *57*(4), 942–957. https://doi.org/10.1021/acs.jcim.6b00740

Sander, T. (2014). lopP Prediction.

Sotriffer, C. A., Sanschagrin, P., Matter, H., & Klebe, G. (2008). SFCscore: Scoring functions for affinity prediction of protein – ligand complexes. *Proteins: Struct, Funct, Bioinf*, *73*(2), 395–419. https://doi.org/10.1002/prot.22058

Spitzer, R., Cleves, A. E., Varela, R., & Jain, A. N. (2014). Protein Function Annotation By Local Binding Site Surface Similarity. *Proteins*, *82*(4), 679–694. https://doi.org/10.1038/jid.2014.371

Stepniewska-Dziubinska, M. M., Zielenkiewicz, P., & Siedlecki, P. (2018). Development and evaluation of a deep learning model for protein-ligand binding affinity prediction. *Bioinformatics*, *34*(21), 3666–3674. https://doi.org/10.1093/bioinformatics/bty374

Su, M., Yang, Q., Du, Y., Feng, G., Liu, Z., Li, Y., & Wang, R. (2019). Comparative Assessment of Scoring Functions: The CASF-2016 Update. *Journal of Chemical Information and Modeling*, *59*(2), 895–913. https://doi.org/10.1021/acs.jcim.8b00545

Sunseri, J., King, J. E., Francoeur, P. G., & Koes, D. R. (2019). Convolutional neural network scoring and minimization in the D3R 2017 community challenge. *Journal of Computer-Aided Molecular Design*, *33*(1), 19–34. https://doi.org/10.1007/s10822-018-0133-y

Tanchuk, V. Y., Tanin, V. O., Vovk, A. I., & Poda, G. (2016). A New, Improved Hybrid Scoring Function for Molecular Docking and Scoring Based on AutoDock and AutoDock Vina. *Chemical Biology and Drug Design*, *87*(4), 618–625. https://doi.org/10.1111/cbdd.12697

Torng, W., & Altman, R. B. (2018). Graph Convolutional Neural Networks for Predicting Drug-Target Interactions. *BioRxiv*. https://doi.org/10.1101/473074

van der Walt, S., Colbert, S.C., Varoquaux, G. (2011). The NumPy Array: A Structure for Efficient Numerical Computation. Computing in Science & Engineering, 13(2), 22-30. http://10.1109/MCSE.2011.37

Wang, C., & Zhang, Y. (2017). Improving scoring-docking-screening powers of protein–ligand scoring functions using random forest. *Journal of Computational Chemistry*, *38*(3), 169–177. https://doi.org/10.1002/jcc.24667

Wang, R., Lai, L., & Wang, S. (2002). Further development and validation of empirical scoring functions for structure-based binding affinity prediction. *Journal of Computer-Aided Molecular Design*, *16*(1), 11–26. https://doi.org/10.1023/A:1016357811882

Wójcikowski, M., Ballester, P. J., & Siedlecki, P. (2017). Performance of machine-learning scoring functions in structure-based virtual screening. *Scientific Reports*, *7*(December 2016), 1–10. https://doi.org/10.1038/srep46710

Yadava, U. (2018). Search algorithms and scoring methods in protein-ligand docking. *Endocrinology&Metabolism International Journal*, *6*(6), 359–367. https://doi.org/10.15406/emij.2018.06.00212

Zheng, M., Xiong, B., Luo, C., Li, S., Liu, X., Shen, Q., … Jiang, H. (2011). Knowledge-based scoring functions in drug design: 3. A two-dimensional knowledge-based hydrogen-bonding potential for the prediction of protein-ligand interactions. *Journal of Chemical Information and Modeling*, *51*(11), 2994–3004. https://doi.org/10.1021/ci2003939

Supporting information

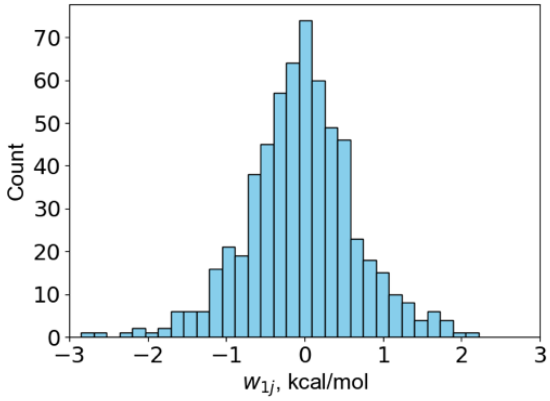

Figure S.1: Distribution of values of increments w_{1j} assigned to atom pair types defined for SF1 scoring function

Table S.1: Contribution fractions of terms E_1-E_3 to the overall sum according to formula (1)

No. of intervals	E_1	E_2	E_3
1	72%	3%	25%
2	59%	28%	13%
3	63%	16%	21%
4	56%	25%	19%

Table S.2: Scoring and ranking powers evaluated on "CASF-2016" benchmark of the top 10 scoring functions producing the best Pearson correlation coefficients reported in (Su et al., 2019)

	Rank[a]	R[b]	SD[c]	SP[d]	τ[e]	PI[f]
$\Delta_{Vina}RF_{20}$	1	0.816	1.26	0.674	0.614	0.691
X-Score	2	0.631	1.69	0.595	0.523	0.625
X-ScoreHS	3	0.629	1.69	0.560	0.489	0.590
ΔSAS	4	0.625	1.70	0.589	0.515	0.608
X-ScoreHP	5	0.621	1.70	0.566	0.509	0.596
ASP@GOLD	6	0.617	1.71	0.542	0.463	0.569
ChemPLP@GOLD	6	0.614	1.72	0.618	0.540	0.647
X-ScoreHM	7	0.609	1.73	0.611	0.536	0.642
Autodock Vina	7	0.604	1.73	0.470	0.414	0.512
DrugScore2018	7	0.602	1.74	0.596	0.505	0.633
DrugScoreCSD	8	0.596	1.75	0.591	0.505	0.626
ASE@MOE	9	0.591	1.75	0.435	0.365	0.459
ChemScore@SYBYL	9	0.590	1.76	0.542	0.474	0.572
PLP1@DS	10	0.581	1.77	0.584	0.505	0.614

[a]Scoring functions are ranked by the Pearson correlation coefficients.

[b]The Pearson correlation coefficient between the experimental binding data and computed binding scores.

[c]The standard deviation (in log Ka units) in fitting the experimental binding data and computed binding scores.

[d]Average Spearman correlation coefficient between the experimental binding data and computed binding scores as obtained on 57 clusters.

[e]Average Kendall correlation coefficient between the experimental binding data and computed binding scores as obtained on 57 clusters.

[f]Average predictive index between the experimental binding data and computed binding scores as obtained on 57 clusters.

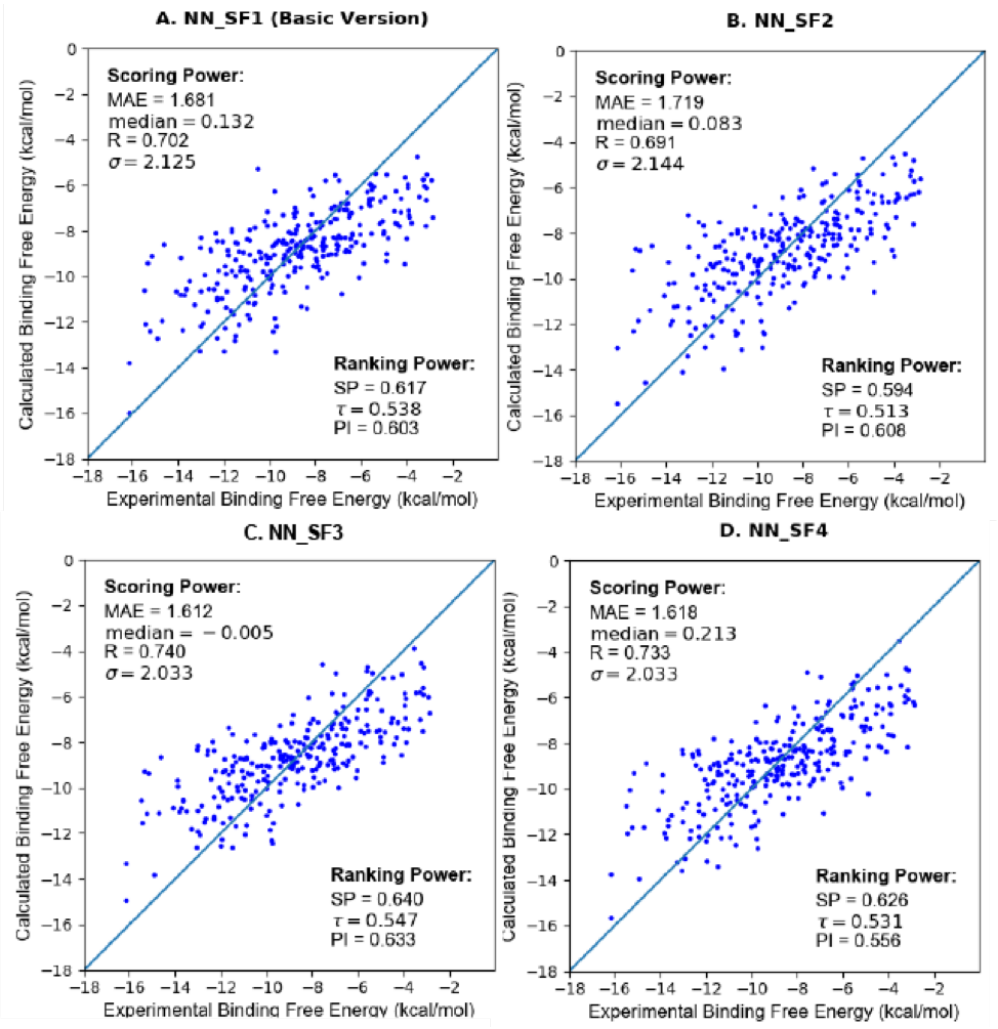

Figure S.2: Accuracy of neural networks used for developing of the scoring functions tested on CASF-2016 benchmark