Computers in Biology and Medicine 151 (2022) 106323

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Deffini: A family-specific deep neural network model for structure-based virtual screening

Dixin Zhou [a,b,1], Fei Liu [a,e,1], Yiwen Zheng [d], Liangjian Hu [d], Tao Huang [b,**], Yu S. Huang [a,c,e,*]

[a] Drug Discovery and Design Center, State Key Laboratory of Drug Research, Shanghai Institute of Materia Medica, Chinese Academy of Sciences, Shanghai, 201203, China
[b] Shenzhen Zhiyao Information Technology Co. Ltd., Shenzhen, Guangdong, China
[c] Genecast Biotechnology Co. Ltd., Wuxi, China
[d] Department of Statistics, Donghua Univerisity, 2999 North Renmin Road, Shanghai, 201620, China
[e] University of Chinese Academy of Sciences, Beijing, 100049, China

ARTICLE INFO

Keywords:
Virtual screening
Protein family-specific model
Structure-based
Convolutional neural network
Drug discovery

ABSTRACT

Deep learning-based virtual screening methods have been shown to significantly improve the accuracy of traditional docking-based virtual screening methods. In this paper, we developed Deffini, a structure-based virtual screening neural network model. During training, Deffini learns protein-ligand docking poses to distinguish actives and decoys and then to predict whether a new ligand will bind to the protein target. Deffini outperformed Smina with an average AUC ROC of 0.92 and AUC PRC of 0.44 in 3-fold cross-validation on the benchmark dataset DUD-E. However, when tested on the maximum unbiased validation (MUV) dataset, Deffini achieved poor results with an average AUC ROC of 0.517. We used the family-specific training approach to train the model to improve the model performance and concluded that family-specific models performed better than the pan-family models. To explore the limits of the predictive power of the family-specific models, we constructed Kernie, a new protein kinase dataset consisting of 358 kinases. Deffini trained with the Kernie dataset outperformed all recent benchmarks on the MUV kinases, with an average AUC ROC of 0.745, which highlights the importance of quality datasets in improving the performance of deep neural network models and the importance of using family-specific models.

1. Introduction

In the early stages of drug discovery, it is critical to find new molecules that bind to protein targets of pharmacological interest. Traditional high-throughput screening assays are expensive and time-consuming, with a high failure rate [1]. Generations of computational screening methods have been developed to help reduce the cost and time in this lead-discovery stage. Traditional methods of virtual screening, such as docking, which employs empirical and knowledge-based scoring functions, rely on pre-defined parameters that model the intermolecular potential energies. Broadly, there are two categories of screening techniques: ligand-based and structure-based [2]. Ligand-based virtual screening is based on searching molecules similar to the known actives without considering the target protein structure, while structure-based virtual screening methods utilize information of both the ligand and the protein target structure to estimate the likelihood that the ligand will bind to the protein with high affinity [3].

Despite the undoubted advantages of docking methods, previous studies have proved that additional methods are needed to correct the docking result even for ligands with the best binding scores [4,5]. Machine learning methods can adapt much more scoring functions such as Support Vector Machine [6], Random Forest [7], Deep Neural Network, Convolutional Neural Network(CNN), and Graph Neural Network [8], utilizing large-scale training sets [3,9] and combining multiple sources of information [10,11], can provide great adaptability and transferability [9,12] to solve multiple prediction tasks or classification tasks simultaneously [13]. Therefore, it has been routinely and successfully applied machine learning methods to re-calculate binding scoring

* Corresponding author. Drug Discovery and Design Center, State Key Laboratory of Drug Research, Shanghai Institute of Materia Medica, Chinese Academy of Sciences, Shanghai, 201203, China.
** Corresponding author.
E-mail addresses: thuang@deepdrug.com (T. Huang), polyactis@gmail.com (Y.S. Huang).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.compbiomed.2022.106323
Received 12 September 2022; Received in revised form 31 October 2022; Accepted 14 November 2022
Available online 17 November 2022

functions.

CNN is a class of deep learning models commonly applied to analyze visual images. When Krizhevsky et al. [14] won the large-scale ImageNet competition by a significant margin over other machine learning methods, CNNs showed to be competitive to solve non-image problems [15–17], and most notably the structure-based virtual screening [18–21]. The first CNN-based scoring function was introduced by Wallach et al. [18], which gained much better performance than docking, considering local 3D structures of protein-ligand complexes. Recently, more CNN models were presented to predict the binding affinity of a ligand and its target protein [22,23]. The advantage of the CNN model in information extraction is undoubted, however, when applying a CNN-based model in transfer learning, the performance may fall short of expectations [24].

In this work, we developed a CNN-based model, Deffini, to predict the probability that a small molecule compound binds a target. During the development of the model, we found that to improve generalizability, a family-specific approach to train a model is much better than the pan-family approach. Secondly, inspired by the work of Imrie et al. [25], which postulates that if more target proteins from the same family and their associated ligands are added to the training set, the generalizability of the model will be enhanced, we constructed the Kernie dataset, covering 358 kinases and 32,000 compounds, to evaluate the performance of the family-specific model. Through various comparative analyses, we demonstrated that Deffini outperformed other methods in predicting the binding affinity of a ligand and its target protein.

2. Materials and methods

2.1. Datasets

Three different datasets were used to evaluate the performance of our virtual screening model, Directory of Useful Decoys – Enhanced (DUD-E), Maximum Unbiased Validation (MUV) and Kernie (a self-collected kinase dataset).

DUD-E is an enhanced and rebuilt version of Directory of Useful Decoys (DUD) [26], designed to help benchmark structure-based virtual screening methods. The 102 targets cover a diverse set of protein families, including 22,886 clustered ligands. For each active, 50 decoys are drawn from ZINC [27], bearing similar physicochemical properties but dissimilar 2-D topology from the active. Ligand clustering is done to reduce the number of ChEMBL [28] ligands down to a manageable size while also increasing the scaffold diversity as suggested by Good and Oprea [29].

MUV is a collection of benchmark datasets that is equally unbiased for the assessment of the quality of virtual screening methods. The MUV datasets were designed to avoid analog bias and artificial enrichment, which produce overly optimistic predictions of virtual screening performance. Selected from confirmatory screens, actives are maximally spread based on simple descriptors and embedded in the chemical space of the decoys, with a ratio of actives to decoys of 1:500 (=30:15000).

Kernei, a large kinase-specific dataset, was constructed by curating data from ChEMBL, PubChem, and PDB. In total, we collected 358 kinase targets and 32,000 compounds by following the criteria proposed by Wallach et al. [18], where targets have annotated binding sites with the highest resolution and actives with IC_{50} or K_i lower than 1 μM. For each active compound, 50 decoys were generated using the same procedure as that of DUD-E.

To train Deffini, we utilized two training sets. The first training set is the DUD-E dataset [30], through which we optimized the network topology and hyperparameters of Deffini by 3-fold clustered cross-validation. We then created two family-specific models by training Deffini with the kinase and the protease subsets from DUD-E and evaluated them on the family-specific independent test sets from the MUV (Maximum Unbiased Validation) dataset [31]. The MUV dataset was used only for testing, not training. The second training set is Kernie, a self-collected kinase-only dataset consisting of 358 kinase targets and 32,000 compounds.

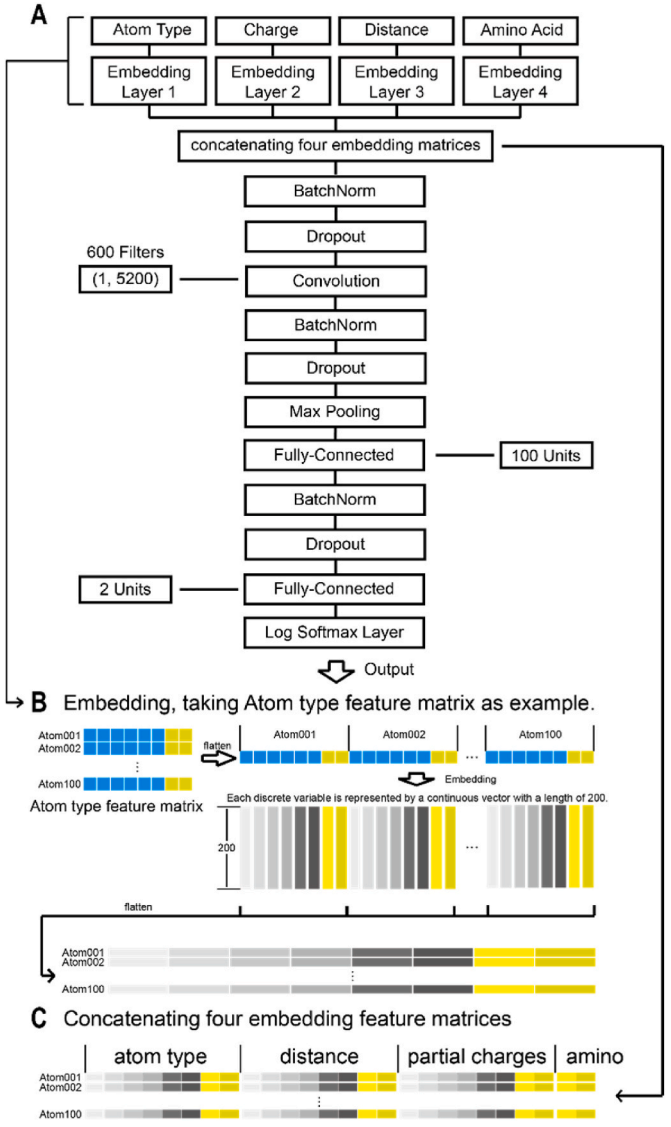

Fig. 1. The structure of Deffini model. (A) The input and model structure of Deffini. (B) The way to embed the basic attribute matrices. (C) Four embedding feature matrices concatenate to the embedding layer.

2.2. Model input

Ligand-target complexes were first generated by the Smina [32] docking program. We retained the top (the lowest binding energy) binding pose of the compound and its protein target and obtained the pdbqt files containing the atomic type, coordinate, and charge information of all atoms in the complex. To transform a ligand-target 3D pose into a 2D matrix that is suitable for CNN processing, we adopted the structure-information-extraction method proposed by Pereira et al. [33], in which the context of an atom is defined by a set of physiochemical attributes of its neighborhood, and adopted the atom-information-extraction criteria presented by DeepDock [34]. Deffini stores the context of each atom in the ligand in one row of the 2D matrix and constrains the maximum number of ligand atoms to 100 (if the total number of atoms in a ligand is less than 100, the extra rows are filled with zeros). Horizontally, the 2D matrix can be decomposed into four 2D matricies, corresponding to four attributes respectively. For one reference atom *a* of a ligand, Deffini extracts three attributes (the atom

type, charge, and the distance to the reference atom) of $k_c = 6$ neighboring atoms in the ligand and $k_p = 2$ neighboring atoms in the target protein and stores them in three separate 100 * 8 (100 ligand atoms * 8 neighboring atoms) sub-matrices. For the last attribute (amino acid type), which is only applicable to $k_p = 2$ atoms in the target protein, Deffini stores the data in a 100 *2 sub-matrix. This approach is inspired by findings from previous studies [35] which highlight the importance of physicochemical attributes of neighboring atoms in both the ligand and the target protein for structure-based drug design. For encoding the four attributes, we discretized the categorical variables (the atom type and the amino acid type) by mapping them to integers, and discretized the continuous variables (the atomic partial charge and the distance to the reference atom) by assigning them to equidistant bins between a pre-defined minimum value and maximum value. The final 2D matrix is formed by concatenating the four sub-matrices into a 100*26 matrix, which approximates the physiochemical environment of the binding pocket of a ligand-target complex.

2.3. Model architecture

The Deffini model as in Fig. 1A contains one input layer, three batch normalization layers, one embedding layer, one convolutional layer, three dropout layers, one max-pooling layer, two fully-connected(fc) layers, and a sigmoid output that predicts the probability of binding.

In the embedding layer, each row in the basic attribute matrix is transformed into a corresponding embedding matrix in the way that each value of a basic attribute is mapped to a real-valued column vector of fixed size which is the dimensionality of the embedding (Fig. 1B), and four embedding feature matrices are concatenated to generate one feature matrix representing the binding information of a ligand (Fig. 1C), which forms the basis for the subsequent convolutional layers to extract relevant information.

The convolution layer consists of a variable number of filters. Units in one filter take inputs only from a small subregion of the input 2D matrix, and all units in a filter are constrained to share the same weight values, which can serve as feature detectors. All units in a feature map detect the same pattern despite their different locations in the input matrix. When the atoms or physiochemical attributes are shifted, the activations of the feature map will be shifted by the same amount but will otherwise be unchanged. This provides the basis for the invariance of the outputs to the rotations of the protein-ligand 3D pose.

The Deffini model has three dropout layers, which can effectively avoid overfitting. Based on a pre-determined probability of dropout, these layers temporarily remove a random subset of neurons and their connections from the model during training and then update the weight parameters of the retained neurons.

One max pooling layer follows the convolutional layer. We chose the max pooling over the average pooling as the latter was shown to obliterate predictive performance [25]. The pooling layer reduces the dimensionality of the representation, the number of parameters, the memory footprint, and the amount of computation in the network. It also prevents overfitting. Our max-pooling operation selects the maximum value in the matrix along the vertical axis for each column vector. After this operation, the input 2D matrix becomes a one-dimensional vector, speeding up the training time.

The final output layer is a fully connected layer with one output neuron. The output neuron employs a sigmoid activation function whose output is the binding probability of the ligand to the protein target. The Deffini model is implemented in the TensorFlow framework [36].

Table 1
Results of Deffini and other models in 3-fold cross-validation on the DUD-E dataset.

Method	AUC_ROC	AUC_PRC	EF1%	EF5%	EF10%
Deffini	0.921 ± 0.077	0.440 ± 0.224	21.597 ± 11.131	11.861 ± 4.350	7.426 ± 1.977
GanDTI	0.860 ± 0.087	0.240 ± 0.1823	16.171 ± 11.472	8.921 ± 4.301	5.930 ± 2.099
CNN	0.858 ± 0.098	0.247 ± 0.185	16.573 ± 11.030	9.166 ± 4.506	5.972 ± 2.213
Transformer	0.825 ± 0.100	0.195 ± 0.156	13.581 ± 10.425	7.562 ± 4.058	5.212 ± 2.104
Smina	0.712 ± 0.119	0.132 ± 0.117	8.273 ± 8.516	4.474 ± 2.845	3.342 ± 1.707

2.4. Model training

To measure the classification (binding or not) performance on a dataset, Deffini uses the binary cross-entropy as its loss function,

$$\mathbf{Loss} = -\frac{\mathbf{1}}{\boldsymbol{N}}\sum_{i=1}^{N}\boldsymbol{y}_i \bullet \boldsymbol{log}(\boldsymbol{p}(\boldsymbol{y}_i)) + (\mathbf{1}-\boldsymbol{y}_i) \bullet \boldsymbol{log}(\mathbf{1}-\boldsymbol{p}(\boldsymbol{y}_i))$$

where $\mathbf{y}$ is the label (1 for active and 0 for decoy) and $p(y_i)$ is the predicted probability that ligand i is active. During the model training, the loss is minimized via the backpropagation (BP) algorithm, widely used in training feedforward neural networks in supervised learning. In deep learning, the BP algorithm computes the gradient of the loss function with respect to the node weights of the network backwardly and adjusts the weights to further reduce the loss function. To prevent arithmetic overflow and underflow in calculating the probability that ligand i binds the target protein, we applied Log Softmax

$$\boldsymbol{L}_i = \boldsymbol{log}\left(\frac{\boldsymbol{e}^{\boldsymbol{x}_i}}{\sum_{j=1}^{n}\boldsymbol{e}^{\boldsymbol{x}_j}}\right) = \boldsymbol{log}\left(\frac{\boldsymbol{e}^{\boldsymbol{x}_i-\boldsymbol{M}}}{\sum_{j=1}^{n}\boldsymbol{e}^{\boldsymbol{x}_j-\boldsymbol{M}}}\right) = \boldsymbol{x}_i - \boldsymbol{M} - \boldsymbol{log}\left(\sum_{j=1}^{n}\boldsymbol{e}^{\boldsymbol{x}_j-\boldsymbol{M}}\right),$$

in which $M = \max(x_i), i = 1,\ldots,n$.

During training, the model is optimized by the Adam algorithm which leverages the power of adaptive learning rates to find the optimal parameters(weights) faster and using fewer resources. We set the initial learning rate as 0.0001, the weight decay as 0.001 and the dropout rate as 0.1. We train our model with a batch size of 1024 for 11 epochs. The order of training data is shuffled for each epoch. The balance of positive and negative samples is achieved by having an equal number of negative samples to positive samples in each sampling batch.

2.5. Model evaluation

It is important to evaluate the generalizability of Deffini to new proteins and ligands, rather than its ability to mesmerize the training data. We evaluated the performance of the Deffini, against the docking program Smina and deep learning methods (Transformer [37], CNN, and GanDTI [38]) by three-fold clustered cross-validation, in which Transformer and CNN employ the SMILES representation of a ligand as model input while GanDTI employs the SMILES and the target protein sequence. Proteins were clustered by sequence similarity using CD-HIT [39], and only targets with greater than 50% sequence identity were included in the same fold to avoid testing on highly similar targets.

We used three evaluation metrics: the Area Under the Curve of the Receiver Operating Characteristic (AUC ROC), the Area Under the Curve of the Precision-Recall Curve (AUC PRC), and the enrichment factor (EF). AUC ROC and AUC PRC are highly correlated and are used to evaluate binary classification problems in machine learning on a given dataset. If the classification is completely random, the AUC ROC will be equal to 0.5 but the AUC PRC will be proportional to the imbalance of the data. The enrichment factor (EF) at x% measures the enrichment of actives among the top x% ranked compounds.

$$\text{EF at x\%} = \frac{\text{number of actives at top x\%}}{\text{number of molecule at top x\%}} \times \frac{\text{number of total molecules}}{\text{number of total actives}}.$$

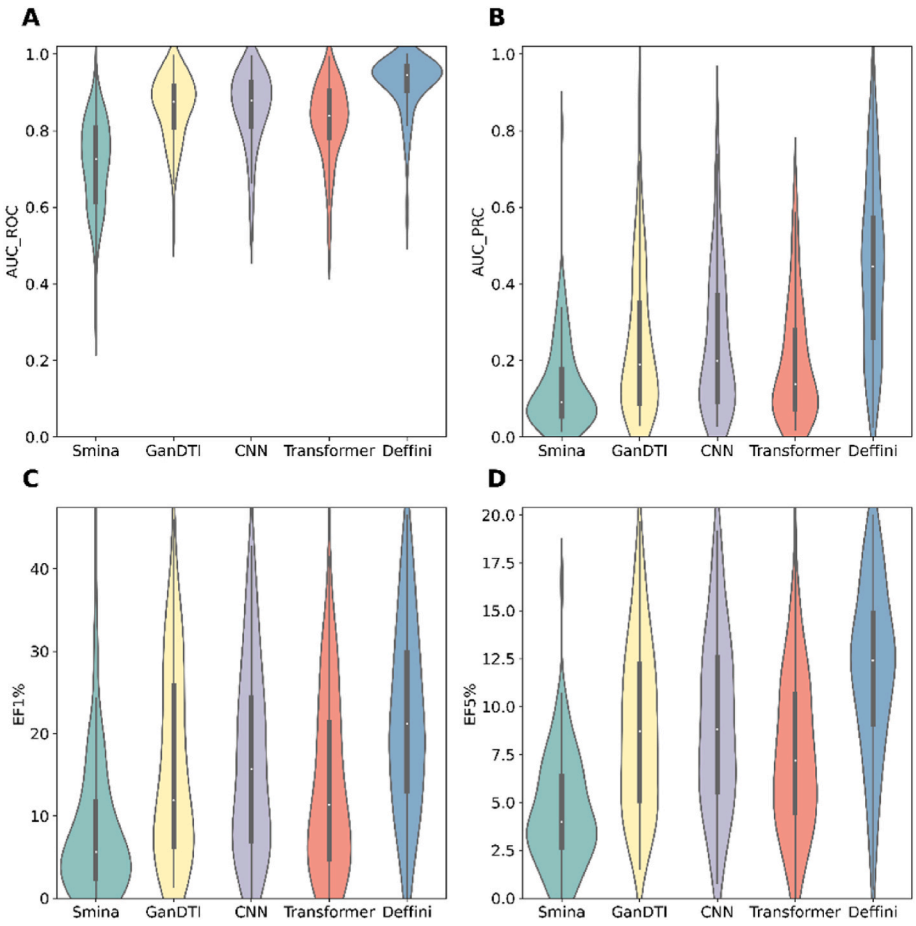

Fig. 2. Performance comparison of Deffini and the other models in clustered three-fold cross-validation of the DUD-E. Violin plots were used to display the distribution of each evaluation metric. (A) AUC of ROC curve (AUC_ROC). (B) AUC of PRC curve (AUC_PRC). (C) Enrichment factor at 1% (EF1%). (D) Enrichment factor at 5% (EF5%). Deffini outperformed Smina and several deep learning approaches with respect to all metrics in this setting.

Table 2
Testing results on the whole MUV dataset of different models trained with the entire DUD-E.

Method	AUC_ROC	AUC_PRC	EF1%	EF5%	EF10%
Smina	0.529 ± 0.082	0.003 ± 0.001	1.787 ± 1.722	1.127 ± 1.067	1.410 ± 0.894
Deffini	0.517 ± 0.064	0.003 ± 0.003	1.531 ± 2.910	1.179 ± 0.867	1.179 ± 0.520
Transformer	0.519 ± 0.054	0.002 ± 0.000	0.510 ± 1.245	0.718 ± 0.506	0.846 ± 0.483
CNN	0.498 ± 0.066	0.002 ± 0.000	0.255 ± 0.920	0.564 ± 0.534	0.743 ± 0.512
GanDTI	0.463 ± 0.082	0.002 ± 0.001	0.000 ± 0.000	0.718 ± 0.743	0.743 ± 0.454

3. Results

3.1. Transfer learning: training and testing on different proteins within the same dataset. Performance on the three-fold clustered cross-validation of DUD-E

We carried out a three-fold clustered cross-validation on the DUD-E dataset for Deffini. Deffini achieved an average AUC_ROC of 0.921, AUC PRC of 0.440, EF at 1% of 21.6, EF at 5% of 11.9, and EF at 10% of 7.4, which significantly outperformed Smina and other deep learning approaches, Table 1, Fig. 2.

3.2. Transfer learning: training with the entire DUD-E, testing on MUV

We then derived the pan-family model by training Deffini with the entire DUD-E dataset and tested its performance on the MUV dataset. The pan-family model achieved an average AUC ROC of 0.517, AUC PRC of 0.003, EF1% of 1.531, EF5% of 1.179, and EF10% of 1.179, Table 2. The results of deep-learning models were even worse than that of Smina. Overall, their results were significantly worse than those observed in the three-fold clustered cross-validation of DUD-E.

One of the main causes for the poor results is the significant imbalance of actives and decoys in the MUV dataset. Decoys are characterized by their structural similarity to actives, but their physiochemical properties are different from actives, and thus have no activity with the corresponding protein. During training, a high-learning capacity model tries to internalize the target-binding differences between decoys and actives as much as it can, but its capacity is limited by the scope of the training dataset. The DUD-E dataset has an active to decoy ratio of 1:50 while the MUV dataset has an active to decoy ratio of 1:500. When the ratio of decoys increases, the chance of misclassifying ligands will also increase, which sets a lower bar to begin with. The higher active-to-decoy ratio, 10 times more imbalanced than DUD-E, and the much larger amount of decoys in MUV make it highly challenging for any model trained on DUD-E to generalize well on MUV.

Furthermore, the curse of dimensionality in the high-dimensional physiochemical space of all proteins and ligands also lead to the poor performance. The large inter-protein-family distance limited coverage of DUD-E in the feature space. However, due to the sequence similarity, proteins from the same family are likely to cluster locally in the

Table 3
Testing results on the MUV kinase subset of Deffini models trained with different DUD-E subsets.

Training set	AUC_ROC	AUC_PRC	EF1%	EF5%	EF10%
DUD-E Kinase	0.685 ± 0.155	0.006 ± 0.004	4.147 ± 4.175	2.665 ± 1.961	2.583 ± 1.730
DUD-E GPCR	0.555 ± 0.068	0.004 ± 0.002	2.488 ± 3.177	1.832 ± 1.138	1.417 ± 0.4120
DUD-E	0.545 ± 0.070	0.003 ± 0.001	0.829 ± 1.659	1.166 ± 0.333	1.333 ± 0.720
DUD-E Miscellaneous	0.507 ± 0.052	0.003 ± 0.001	1.659 ± 1.915	0.833 ± 0.838	0.583 ± 0.500
DUD-E Protease	0.459 ± 0.060	0.002 ± 0.000	0.830 ± 1.659	0.666 ± 0.000	0.583 ± 0.319

Table 4
Testing results on the MUV protease subset of Deffini models trained with different DUD-E subsets.

Training set	AUC_ROC	AUC_PRC	EF1%	EF5%	EF10%
DUD-E Protease	0.564 ± 0.050	0.003 ± 0.001	3.318 ± 3.318	1.332 ± 1.332	1.111 ± 0.509
DUD-E	0.532 ± 0.081	0.002 ± 0.001	0.000 ± 0.000	0.666 ± 0.666	1.333 ± 0.333
DUD-E Miscellaneous	0.498 ± 0.050	0.002 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.444 ± 0.509
DUD-E Kinase	0.474 ± 0.037	0.002 ± 0.000	0.000 ± 0.000	0.444 ± 0.769	1.000 ± 0.667
DUD-E GPCR	0.470 ± 0.041	0.002 ± 0.000	0.000 ± 0.000	0.222 ± 0.385	0.333 ± 0.333

physiochemical space. Likewise, the active ligands and decoys from the same protein family are also likely to cluster in the physiochemical space. Overall, it is a sparse space with local concentration for individual protein families and their associated actives and decoys. Deffini trained with the whole DUD-E is likely to pick up biases between all active ligands and decoys, rather than protein-ligand binding information [40], which explains why Deffini performed well in the cross-validation on the dataset itself but generalized poorly to new proteins and ligands in MUV. Learning bias in the training stage and the unbalance of active vs decoy in the MUV dataset resulted in Deffini and other deep learning models to perform poorly in the MUV dataset.

3.3. *Transfer learning: training with family-specific DUD-E data showed improved performance on MUV*

Proteins from the same family usually share a common origin with similar 3D structure, function, and significant sequence similarity, which are easily to show local aggregation in the materialization space,

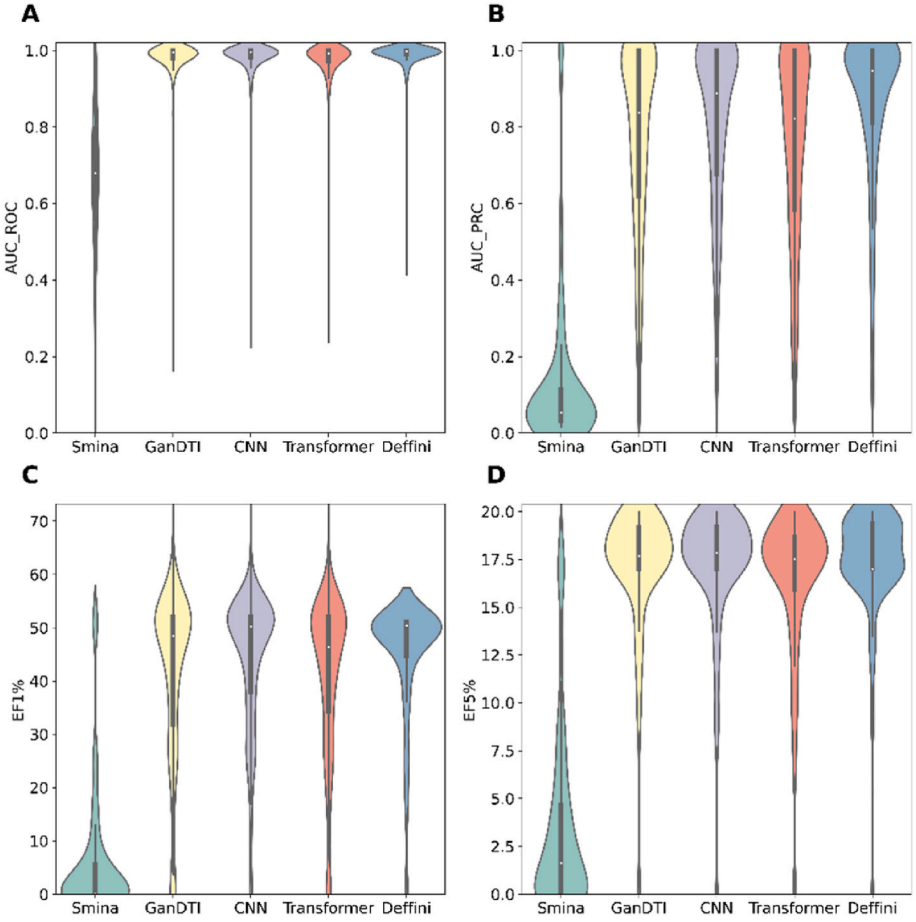

Fig. 3. Performance comparison of Smina, several deep learning methods and Deffini in clustered three-fold cross-validation of the Kernie dataset. Violin plots were used to display the distribution of each evaluation metric. (A) AUC of ROC curve (AUC_ROC). (B) AUC of PRC curve (AUC_PRC). (C) Enrichment factor at 5% (EF5%). (D) Enrichment factor at 10% (EF10%). Deffini outperformed Smina with respect to all metrics in this setting.

Table 5
Results of Deffini and other models in 3-fold cross-validation on the Kernie dataset.

Metric	AUC_ROC	AUC_PRC	EF1%	EF5%	EF10%
Deffini	0.985 ± 0.046	0.857 ± 0.206	44.841 ± 11.178	17.228 ± 3.290	9.103 ± 1.371
CNN	0.980 ± 0.055	0.808 ± 0.229	43.864 ± 13.218	17.170 ± 3.316	9.103 ± 1.242
GanDTI	0.977 ± 0.060	0.768 ± 0.250	40.921 ± 16.307	16.941 ± 3.675	9.036 ± 1.507
Transformer	0.973 ± 0.066	0.748 ± 0.255	40.726 ± 15.340	16.402 ± 3.912	8.942 ± 1.430
Smina	0.664 ± 0.192	0.136 ± 0.216	5.879 ± 11.966	3.379 ± 4.725	2.538 ± 2.646

Table 6
Testing results on the entire MUV dataset of different models trained with Kernie.

Method	AUC_ROC	AUC_PRC	EF1%	EF5%	EF10%
Deffini	0.600 ± 0.132	0.008 ± 0.011	3.828 ± 4.250	2.614 ± 2.251	2.000 ± 1.764
Transformer	0.585 ± 0.104	0.006 ± 0.008	3.061 ± 4.779	2.460 ± 2.655	1.820 ± 1.561
CNN	0.548 ± 0.074	0.004 ± 0.002	1.020 ± 1.592	1.281 ± 0.960	1.154 ± 0.618
GanDTI	0.541 ± 0.091	0.004 ± 0.003	2.551 ± 3.621	1.794 ± 1.618	1.538 ± 0.918
Smina	0.529 ± 0.082	0.003 ± 0.001	1.787 ± 1.722	1.127 ± 1.067	1.410 ± 0.894

so we speculate that restricting the training and test set data to be from the same protein family may exhibit better performance.

To test the hypothesis that employing training data the model generalizability, we derived family-specific models by restricting training data from a single protein family, which was shown to outperform a pan-family model trained with data from all protein families [41,42]. We constructed two family-specific models training by the 26 kinase subset and 15 protease subset of DUD-E, and tested on 4 kinase targets and 3 protease targets from MUV, respectively.

Adopting the kinase-specific model instead of the pan-family model led to average improvement in AUC_ROC of 25.7%, and AUC_PRC of 100%, Table 3. Consistent results were also exhibited in the kinase-specific experiment with an improvement of AUC_ROC by 6.0%, and AUC_PRC by 50%, Table 4, compared to the pan-family model. And both the kinase-specific and protease-specific model showed better performance in the family-matching MUV subsets. We conclude that family-specific models are better at extracting protein-binding-specific information than the pan-family model, which leads to their improved generalizability.

3.4. Deffini performed well on the three-fold cross-validation of Kernie

The kinase protein family-specific Deffini model was constructed with the kinase targets data in the DUD-E dataset. However, due to the small amount of data in the training set, it is easy to lead to model overfitting and poor model generalization ability, which affects the performance on the test set. Therefore, we constructed a dataset, Kernie, with much larger amount of data than the kinase targets data in DUD-E.

Kernie contains 358 kinase target 3D information and Fig. 3 shows the evaluation result of three-fold clustered cross-validation on the Kernie dataset, with an average AUC_ROC of 0.985, AUC_PRC of 0.857, EF at 1% of 44.8, EF at 5% of 17.2, and EF at 10% of 9.1, outperforming other deep learning methods and significantly outperforming Smina, Table 5.

Table 7
Testing results on the MUV kinase subset of different models trained with Kernie.

Training set	AUC_ROC	AUC_PRC	EF1%	EF5%	EF10%
Deffini	0.745 ± 0.112	0.018 ± 0.017	7.465 ± 5.666	5.163 ± 2.201	3.750 ± 2.007
Transformer	0.648 ± 0.148	0.012 ± 0.014	5.803 ± 7.836	4.497 ± 4.260	2.999 ± 2.356
CNN	0.611 ± 0.096	0.003 ± 0.001	0.829 ± 1.658	1.832 ± 1.477	1.584 ± 0.569
GanDTI	0.607 ± 0.043	0.004 ± 0.001	0.000 ± 0.000	2.166 ± 0.999	2.166 ± 0.577
Smina	0.505 ± 0.121	0.003 ± 0.002	1.659 ± 1.915	0.999 ± 1.586	1.250 ± 1.198

Table 8
Testing results on the MUV kinase and protease subsets of Deffini trained with different Kernie subsets.

Training set	AUC_ROC	AUC_PRC	EF1%	EF5%	EF10%
Testing on MUV kinase subset					
Kernie	0.745 ± 0.112	0.018 ± 0.017	7.465 ± 5.666	5.163 ± 2.201	3.750 ± 2.007
Kernie-minus-MUV	0.736 ± 0.141	0.016 ± 0.012	6.636 ± 4.692	5.163 ± 3.095	3.833 ± 1.934
Testing on MUV protease subset					
Kernie	0.483 ± 0.026	0.002 ± 0.000	0.000 ± 0.000	0.222 ± 0.385	0.111 ± 0.192
Kernie-minus-MUV	0.522 ± 0.032	0.002 ± 0.000	0.000 ± 0.000	0.444 ± 0.385	0.778 ± 0.385

3.5. Family-specific models trained with Kernie showed further improvement in MUV

The prior family-specific experiments showed that concentrating the training dataset on a particular protein family improved the generalizability of the model, therefore, we further evaluated the performance of Deffini when training with Kernie dataset and testing on various datasets. The Kernie-trained Deffini outperformed other models with respect to all metrics on the MUV dataset, with an average AUC ROC of 0.600, AUC PRC of 0.008, EF1% at 3.828, EF5% at 2.614, and EF10% at 2.000, Table 6. However, the Deffini results were significantly lower than that of the 3-fold cross-validation results on the Kernie dataset. This is attributed to the fact that all Kernie targets are from the kinase protein family, while the MUV dataset has non-kinase proteins.

Thus in the next step, we evaluated the independent performance of the Kernie-trained model by testing Deffini on the MUV kinase subset, and the results showed that the performance of the kinase-specific model is substantially better, with an average AUC ROC of 0.745, AUC PRC of 0.018, EF1% at 7.465, EF5% at 5.163, and EF10% at 3.750, which outperformed other models, Table 7. The improved performance shows how family-specific models significantly improve the power of drug virtual screening models.

To evaluate the generalizability of the family-specific model more objectively, we trained Deffini with Kernie or Kernie-minus-MUV (a Kernie dataset excluding kinases that are also present in MUV) and tested its performance on the MUV kinase and protease subset, Table 8, Fig. 4. The results indicate that testing on proteins from a different family completely destroyed the model performance. But the exclusion of MUV kinase targets in Kernie only slightly decreased its performance, which further validated the family-specific approach.

4. Discussion

Structure-based virtual screening is an important tool for compound prioritization, but traditional force field-based or physics-based scoring functions or empirical scoring function do not work well for such

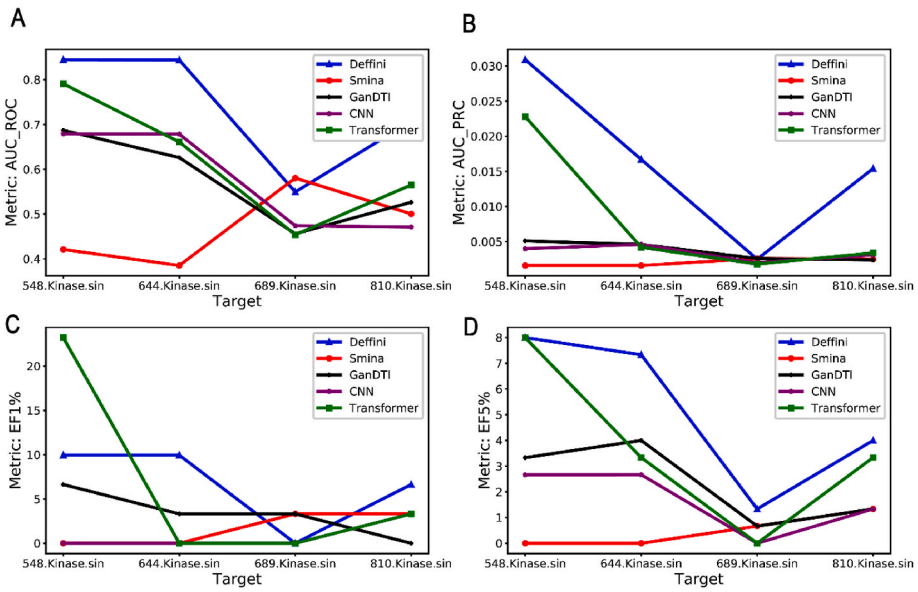

Fig. 4. Testing results on MUV kinases of different models trained by the Kernie-minus-MUV dataset.

problems. Deep learning-based methods provide another promising approach for prioritizing compounds during virtual screening. In this work, we present Deffini, a deep convolutional neural network for scoring protein-ligand interactions, which is trained to classify compounds as binders or nonbinders using 3D structural information of a protein-ligand complex, so as to provide medicinal chemists with candidate molecules with strong binding potential to target proteins.

The better performance of the Deffini model is attributed to the refined deep-learning neural network architecture and highly dependent on the quality of training and testing dataset. Deffini extracts the structural features of the 3D ligand-protein binding pocket with the CNN model, and then sort the relevant features through max pooling. Used in combination, the max pooling and dropout layers in Deffini effectively prevent the model from overfitting, which results in the improved its generalizability.

Pan-family models (trained from mix-protein-family data such as DUD-E) performed well in cross-validation but performance deteriorated in independent validation datasets, implying a poor generalizability. In contrast, the family-specific training and testing strategy enabled Deffini to gain consistent results. Because ligand-target binding mode varies significantly from one protein family to another, factors that are important to binding in one protein family are likely inapplicable to another family, which is why family-specific models consistently outperform pan-family models.

As expected, increasing the size of the protein family dataset (kinases in our case) further improved the performance for deep-learning models. A large training dataset could help to correct the inherent bias (intra-target and inter-target) in small datasets [40]. Deffini showed significantly better performance when being trained on the Kernie dataset and tested on the MUV dataset especially on the MUV kinase subset, indicating its better learning capacity compared to other deep-learning models. Being a much larger dataset, Kernie helps Deffini to better calibrate the weights of different factors that contribute to the ligand-target binding. Despite the improved performance, the AUC ROC did not increase beyond 0.75. The main reason could be attributed to the inherent differences between computer-generated decoys in Kernie and the experimentally validated decoys in MUV, which could confound deep-learning models with a high learning capacity and lead to the limited generalizability. Additionally, the noises from inaccurate input docking poses (the top-ranked pose is not always the accurate one) would also reduce to the generalizability of a model. In the next-phase of our research, more experimentally-validated actives and decoys and higher learning capacity neural network modules such as 3D CNN model will be used to improve upon the current work. However, we believe that family-specific models will be widely used to better predict the ligand-protein binding potential in virtual screening.

5. Conclusion

We provided a novel solution to the problem of generalizing deep learning-based scoring functions in virtual screening. Utilizing different family-specific datasets to train models shows that family-specific models outperform the pan-family model. It suggests that a training set composed of a mixture of various protein family bioactivity data seems to interfere with the prediction performance of the model. Our family-specific model showed outstanding results on MUV, a difficult virtual screening benchmark dataset when being trained with Kernie, a self-collected larger kinase-specific dataset, which shows that a high-quality family specific bioactivity dataset is helpful to establish a powerful virtual screening model.

Funding sources

"Personalized Medicines - Molecular Signature-based Drug Discovery and Development" Strategic Priority Research Program of the Chinese Academy of Sciences XDA12050202 to YSH.

Declaration competing interest

None Declared.

References

[1] D. Hecht, G.B. Fogel, Computational intelligence methods for docking scores, Curr. Comput. Aided Drug Des. 5 (2009) 56–68.
[2] C. McInnes, Virtual screening strategies in drug discovery, Curr. Opin. Chem. Biol. 11 (2007) 494–502.
[3] R.T. Kroemer, Structure-based drug design: docking and scoring, Curr. Protein Pept. Sci. 8 (2007) 312–328.
[4] D. Ramírez, J. Caballero, Is it reliable to take the molecular docking top scoring position as the best solution without considering available structural data? Molecules (2018) 23.
[5] D. Ramírez, J. Caballero, Is it reliable to use common molecular docking methods for comparing the binding affinities of enantiomer pairs for their protein target? Int. J. Mol. Sci. 17 (2016).
[6] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (1995) 273–297.

[7] A. Liaw, M. Wiener, Classification and regression by randomForest, R. News 2 (2002) 18–22.
[8] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, M. Sun, Graph neural networks: a review of methods and applications, AI Open 1 (2020) 57–81.
[9] X. Li, X. Wu, Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition, in: 2015 Ieee International Conference on Acoustics, Speech and Signal Processing (Icassp), IEEE, 2015, pp. 4520–4524.
[10] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 1798–1828.
[11] K. Stahl, M. Schneider, O. Brock, EPSILON-CP: using deep learning to combine information from multiple sources for protein contact prediction, BMC Bioinf. 18 (2017) 1–11.
[12] L.Y. Pratt, Discriminability-based transfer between neural networks, Adv. Neural Inf. Process. Syst. (1992) 5.
[13] R. Caruana, Multitask learning, Mach. Learn. 28 (1997) 41–75.
[14] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Commun. ACM 60 (2017) 84–90.
[15] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, A. Baskurt, Sequential Deep Learning for Human Action Recognition, International Workshop on Human Behavior Understanding, Springer, 2011, pp. 29–39.
[16] N. Kalchbrenner, E. Grefenstette, P. Blunsom, A Convolutional Neural Network for Modelling Sentences, 2014 arXiv preprint arXiv:1404.2188.
[17] H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong, Q. Zhang, Time-series anomaly detection service at microsoft, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 3009–3017.
[18] I. Wallach, M. Dzamba, A. Heifets, AtomNet: a Deep Convolutional Neural Network for Bioactivity Prediction in Structure-Based Drug Discovery, 2015 arXiv preprint arXiv:1510.02855.
[19] J. Gomes, B. Ramsundar, E.N. Feinberg, V.S. Pande, Atomic Convolutional Networks for Predicting Protein-Ligand Binding Affinity, 2017 arXiv preprint arXiv:1703.10603.
[20] A. Gonczarek, J.M. Tomczak, S. Zaręba, J. Kaczmar, P. Dąbrowski, M.J. Walczak, Learning Deep Architectures for Interaction Prediction in Structure-Based Virtual Screening, 2016 arXiv preprint arXiv:1610.07187.
[21] A. Gonczarek, J.M. Tomczak, S. Zaręba, J. Kaczmar, P. Dąbrowski, M.J. Walczak, Interaction prediction in structure-based virtual screening using deep learning, Comput. Biol. Med. 100 (2018) 253–258.
[22] Z. Wang, L. Zheng, Y. Liu, Y. Qu, Y.-Q. Li, M. Zhao, Y. Mu, W. Li OnionNet-2, A convolutional neural network model for predicting protein-ligand binding affinity based on residue-atom contacting shells, Front. Chem. 9 (2021).
[23] J. Son, D. Kim, Development of a graph convolutional neural network model for efficient prediction of protein-ligand binding affinities, PLoS One 16 (2021), e0249404.
[24] F. Imrie, A.R. Bradley, M. van der Schaar, C.M. Deane, 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. 58 (2018) 2319–2330.
[25] M. Ragoza, J. Hochuli, E. Idrobo, J. Sunseri, D.R. Koes, Protein–ligand scoring with convolutional neural networks, J. Chem. Inf. Model. 57 (2017) 942–957.
[26] N. Huang, B.K. Shoichet, J.J. Irwin, Benchmarking sets for molecular docking, J. Med. Chem. 49 (2006) 6789–6801.
[27] J.J. Irwin, B.K. Shoichet, ZINC– a free database of commercially available compounds for virtual screening, J. Chem. Inf. Model. 45 (2005) 177–182.
[28] A. Gaulton, L.J. Bellis, A.P. Bento, J. Chambers, M. Davies, A. Hersey, Y. Light, S. McGlinchey, D. Michalovich, B. Al-Lazikani, ChEMBL: a large-scale bioactivity database for drug discovery, Nucleic Acids Res. 40 (2012) D1100–D1107.
[29] A.C. Good, T.I. Oprea, Optimization of CAMD techniques 3. Virtual screening enrichment studies: a help or hindrance in tool selection? J. Comput. Aided Mol. Des. 22 (2008) 169–178.
[30] M.M. Mysinger, M. Carchia, J.J. Irwin, B.K. Shoichet, Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking, J. Med. Chem. 55 (2012) 6582–6594.
[31] S.G. Rohrer, K. Baumann, Maximum unbiased validation (MUV) data sets for virtual screening based on PubChem bioactivity data, J. Chem. Inf. Model. 49 (2009) 169–184.
[32] D.R. Koes, M.P. Baumgartner, C.J. Camacho, Lessons learned in empirical scoring with smina from the CSAR 2011 benchmarking exercise, J. Chem. Inf. Model. 53 (2013) 1893–1904.
[33] J.C. Pereira, E.R. Caffarena, C.N. Dos Santos, Boosting docking-based virtual screening with deep learning, J. Chem. Inf. Model. 56 (2016) 2495–2506.
[34] Z. Liao, R. You, X. Huang, X. Yao, T. Huang, S. Zhu, DeepDock: Enhancing Ligand-Protein Interaction Prediction by a Combination of Ligand and Structure Information, 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2019, pp. 311–317.
[35] Y. Yuan, J. Pei, L. Lai, Binding site detection and druggability prediction of protein targets for structure-based drug design, Curr. Pharmaceut. Des. 19 (2013) 2326–2333.
[36] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, {TensorFlow}: a System for {Large-Scale} Machine Learning, 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 2016, pp. 265–283.
[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. (2017) 30.
[38] S. Wang, P. Shan, Y. Zhao, L. Zuo, GanDTI: a multi-task neural network for drug-target interaction prediction, Comput. Biol. Chem. 92 (2021), 107476.
[39] Y. Huang, B. Niu, Y. Gao, L. Fu, W. Li, C.D.-H.I.T. Suite, A web server for clustering and comparing biological sequences, Bioinformatics 26 (2010) 680–682.
[40] L. Chen, A. Cruz, S. Ramsey, C.J. Dickson, J.S. Duca, V. Hornak, D.R. Koes, T. Kurtzman, Hidden bias in the DUD-E dataset leads to misleading performance of deep learning in structure-based virtual screening, PLoS One 14 (2019), e0220113.
[41] Y. Wang, Y. Guo, Q. Kuang, X. Pu, Y. Ji, Z. Zhang, M. Li, A comparative study of family-specific protein–ligand complex affinity prediction based on random forest approach, J. Comput. Aided Mol. Des. 29 (2015) 349–360.
[42] A. Amini, P.J. Shrimpton, S.H. Muggleton, M.J. Sternberg, A general approach for developing system-specific functions to score protein–ligand docked complexes using support vector inductive logic programming, Proteins: Struct., Funct., Bioinf. 69 (2007) 823–831.