frontiers | Frontiers in Artificial Intelligence

TYPE Methods
PUBLISHED 23 March 2023
DOI 10.3389/frai.2023.1069353

OPEN ACCESS

EDITED BY
Zongliang Yue,
University of Alabama at Birmingham,
United States

REVIEWED BY
Ming Huang,
Mayo Clinic, United States
Ehsan Saghapour,
University of Alabama at Birmingham,
United States
Yuwei Song,
University of Alabama at Birmingham,
United States

*CORRESPONDENCE
Shuxing Zhang
✉ shuxing@imdlab.net

†PRESENT ADDRESS
Rajan Chaudhari,
Eurofins Beacon Discovery, San Diego, CA,
United States

SPECIALTY SECTION
This article was submitted to
Medicine and Public Health,
a section of the journal
Frontiers in Artificial Intelligence

RECEIVED 13 October 2022
ACCEPTED 03 March 2023
PUBLISHED 23 March 2023

CITATION
Huang B, Fong LWR, Chaudhari R and Zhang S (2023) Development and evaluation of a java-based deep neural network method for drug response predictions.
Front. Artif. Intell. 6:1069353.
doi: 10.3389/frai.2023.1069353

Development and evaluation of a java-based deep neural network method for drug response predictions

Beibei Huang, Lon W. R. Fong, Rajan Chaudhari† and Shuxing Zhang*

Intelligent Molecular Discovery Laboratory, Department of Experimental Therapeutics, The University of Texas MD Anderson Cancer Center, Houston, TX, United States

Accurate prediction of drug response is a crucial step in personalized medicine. Recently, deep learning techniques have been witnessed with significant breakthroughs in a variety of areas including biomedical research and chemogenomic applications. This motivated us to develop a novel deep learning platform to accurately and reliably predict the response of cancer cells to different drug treatments. In the present work, we describe a Java-based implementation of deep neural network method, termed JavaDL, to predict cancer responses to drugs solely based on their chemical features. To this end, we devised a novel cost function and added a regularization term which suppresses overfitting. We also adopted an early stopping strategy to further reduce overfit and improve the accuracy and robustness of our models. To evaluate our method, we compared with several popular machine learning and deep neural network programs and observed that JavaDL either outperformed those methods in model building or obtained comparable predictions. Finally, JavaDL was employed to predict drug responses of several aggressive breast cancer cell lines, and the results showed robust and accurate predictions with r^2 as high as 0.81.

KEYWORDS

artificial intelligence (AI), deep learning, deep neural network, quantitative structure activity relationship (QSAR), multilayer neural network (MNN), triple-negative breast cancer (TNBC), drug response

1. Introduction

Over the past ten years, many machine-learning methods have been developed to tackle the problem of identifying drugs that would work best for a given patient population (Murphy, 2011; Bizzego et al., 2019; Freedman, 2019; Steventon et al., 2019; Vamathevan et al., 2019; Woo, 2019). Recently a new branch of machine learning known as deep learning has been gaining significant attention (Tan et al., 2016a; Lee et al., 2019; You et al., 2019; Zeng et al., 2019; Adam et al., 2020; Baskin, 2020). In particular, the resurgence of neural networks took place around 2005 when more efficient training algorithms were developed and improvements in overfitting were made. This has led to the success of deep neural networks with many applications to biomedical research such as protein structure prediction and drug repositioning (Reddy et al., 2014; Heffernan et al., 2015; Tan et al., 2016b; Carpenter and Huang, 2017; Chaudhari et al., 2017; Smalley, 2017; Tsao et al., 2017; Evans et al., 2018; Lo et al., 2018; Robichaux et al., 2018; Segler et al., 2018; Senior et al., 2020; Humphreys et al., 2021; Jumper et al., 2021). Moreover, deep learning has been employed to diagnose diseases based on medical images (Acharya et al., 2018; Du et al., 2018; Jang et al., 2018; Gurovich et al., 2019; Indraswari et al., 2019; Jeyaraj and Nadar, 2019; Ma et al., 2019).

In the canonical configuration, a normal deep neural network (DNN) consists of an input layer where an input signal is fed, an output layer where predictions are generated, and several hidden (middle) layers which capture features during the training (Kang et al., 2018). DNN is also considered a type of representation learning, in that it allows a machine to be fed raw data and automatically discover the representations needed for detection. During the training process, the raw data in the form of certain signals are fed into the input layer and then transported from one hidden layer to another, in each one leaving a trail forming a certain increasingly abstractive pattern. For instance, in image processing the raw data are arrays of pixel values, while in QSAR modeling it may be descriptors including a variety of physical chemical properties of compounds. Such applications involve a large amount of input data, exposing the drawbacks of previous multilayer neural network (MNN) programs, which only accept limited numbers of input descriptors and have limited numbers of hidden layers and neurons in a hidden layer. Due to these limitations, networks with only a single hidden layer were largely abandoned (Winkler and Le, 2017).

One of the major issues with artificial neural networks is that the models are significantly complex, since neural networks usually have large numbers of layers containing many neurons. The number of connections in these models is astronomical, easily reaching the millions, and overfitting thus becomes very common (Gawehn et al., 2016). In general, there is a direct trade-off between overfitting and model complexity. Inadequately complex models may not be powerful enough to capture all the information necessary to solve a problem, but overly complex ones (especially with a limited amount of data) tend to run into the risk of overfitting (Gawehn et al., 2016). Besides overfitting, another challenge in drug response prediction comes from the activity cliff that is formed when a pair of structurally similar molecules display a large difference in potency (Bajorath, 2017; Watanabe et al., 2017). From the perspective of chemical structures, the activity cliff indicates a lack of assayed compounds in the surrounding space. This can also lead to poorly generalized models and overfitting. Furthermore, the distance between compounds is defined based on their relationship to neighboring compounds. Hence almost all current published prediction models are flawed to some degree due to the limitations that arise from one or more of the above problems.

In the present study, we aim to develop a novel method and build robust models that accurately predict drug response in cancer cells. To this end, we designed an approach which employs deep learning algorithms for drug activity prediction. Our software package, termed JavaDL, integrates several of the latest improved techniques, including regularization, dropout, and early stopping, to mitigate the issues caused by overfitting and activity cliffs. In order to assess its robustness, we used two very different datasets: a Caco-2 dataset for permeability prediction and a hERG dataset for cardiovascular toxicity prediction. We also evaluated the ability and robustness of JavaDL to perform big data analysis using the Merck Molecular Activity Challenge dataset from Kaggle. Finally, our software was employed to predict the drug response of cancer cells using data we recently curated and experimentally measured. The results show that JavaDL could obtain significantly improved prediction and have broad applicability in a variety of datasets as well as a high capability to handle big data problems.

2. Methods

2.1. Dataset

For model building and prediction evaluation, we compared JavaDL with several published studies using the exact same datasets. The first dataset was curated by our group and it has been used to build Caco-2 permeability prediction models with kNN and SVM (Du-Cuny et al., 2009; Smith et al., 2018). The whole dataset contains 174 compounds with 334 descriptors calculated using MOE. Another dataset contains hERG blockers with their corresponding known hERG inhibition activity (pIC_{50}) obtained from previous publications (Du-Cuny et al., 2011). This dataset includes 639 hERG active and inactive compounds. Similarly, MOE was used to calculate the molecular descriptors which were normalized to avoid disproportional weighting for both datasets (Du-Cuny et al., 2011). In addition, to evaluate the ability of JavaDL to handle big data, we obtained a dataset for the Merck Molecular Activity Challenge on Kaggle. The original training data set contained 1,569 compounds with 4,505 descriptors, which after washing shrank to 1,983 descriptors.[1] Finally, we applied JavaDL to predict the TNBC cell response to drugs. The experimental data of cancer cell response to drugs were collected from the MIPE project at the National Center for Advancing Translational Science (NCATS)[2] and our own laboratory's data. We focused on triple-negative breast cancer (TNBC) due to our own research interest, but the approach can be easily applied to other cancer cell lines or even other diseases. In particular, we selected four breast cancer cell lines, HCC-1937, MDA-MB-436, MDA-MB-231, and MDA-MB-453, representing four different TNBC molecular subtypes. The dataset contains 274 compounds used in all four cell lines. Each compound is described by 205 molecular descriptors and four half maximal activity (log) concentration (LAC_{50}) values (one for each cell line). Although internal cross validation is generally considered sufficient to justify model predictive power (Tropsha et al., 2003; Zhang et al., 2006a), many researchers have argued that external validation is crucial. In this study compounds in our datasets are divided into training and test sets *via* a rational approach based on the Sphere Exclusion algorithm (Golbraikh et al., 2003; Zhang et al., 2006b).

2.2. Techniques to address challenges in DNN

2.2.1. Activation function

For our program, we essentially adopted a backpropagation algorithm to train the deep neural network. Based on our experience and other reports (Hornik, 1991; Maggiora, 2006; Hinton, 2014; Ma et al., 2015; Angermueller et al., 2016;

1 https://www.kaggle.com/c/MerckActivity

2 https://tripod.nih.gov/matrix-client/

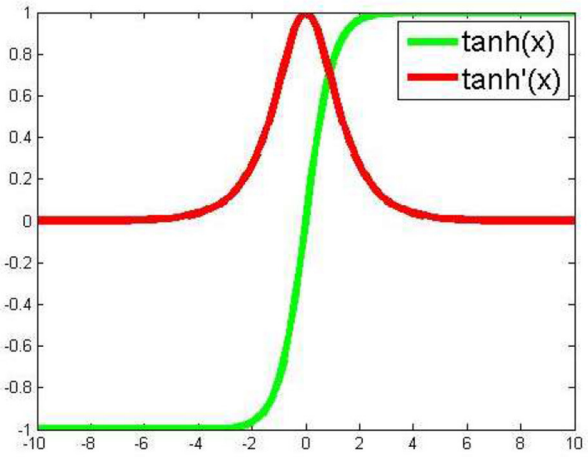

FIGURE 1
Activation function based on the Tanh function tanh(x) and its derivative tanh'(x).

Gawehn et al., 2016), we included five total layers for our deep neural networks, with three hidden layers. Assuming five nodes in the input layer and one node in the output layer, three hidden layers each containing 20 hidden nodes makes the total number of variables (weight variables) 920. In contrast to frequently used activation functions in previous studies (Ma et al., 2015), on each hidden node we adopted an activation function $\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$ (Figure 1), with which one of the advantages is to avoid bias in the gradient during training. For the loss function we chose mean squared error. In the backpropagation process we employed the stochastic gradient descent algorithm for optimization. Additionally, JavaDL has been implemented in a way to deal with data featuring different scales and complexity. For instance, to build a model from a limited dataset, the five-layer structure with each layer containing 20 neurons is sufficient. However, for large datasets such as those in the Merck Molecular Activity Challenge competition, JavaDL can grow accordingly the number of layers and neurons with corresponding increase of the connections between neighboring layers.

2.2.2. Regularization

To address the overfitting issue, we adopted a common regularization technique: addition of a regularization term to the cost function. It is based on the relation between the regularized overall squared-error cost function and the correlation coefficient q^2. Given a training set of m examples, the overall squared-error cost function $J(W, \lambda)$ is considered below:

$$J(W,\lambda) = \frac{1}{m}\sum_{i=1}^{m}\frac{\left\|h_W\left(x^i\right)-y^i\right\|}{2} + \frac{\lambda}{2}\sum_{n=1}^{n_l-1}\sum_{i=1}^{S_l}\sum_{j=1}^{S_{l+1}}\left(W_{ij}^l\right)^2 \quad (1)$$

where n_l denotes the number of layers in the network. S_l represents the number of nodes on lay l, and W_{ij}^l denotes the weight of the connection between node i on lay l and node j on lay $l+1$. The first term is an average sum-of-squares error. For simplicity, we neglect the bias term, but it can be easily taken into account. The second term is the weight decay term, which suppresses overfitting *via* tuning the value of λ to decrease the magnitude of the weights. The method of minimizing the cost function by searching for the optimal value of λ is introduced in Burden and Winkler (2008). This method, called Bayesian Regularized Neural Networks, involves incorporating Bayes' theorem into the regularization scheme. It has been frequently used in current research.

We defined our own cost function by replacing the average sum-of-squares error with – q^2

$$q^2 = 1 - \frac{\sum_{i=1}^{m}\left\|h_W\left(x^i\right)-y^i\right\|^2}{\sum_{i=1}^{m}\left\|y^i-y\right\|^2} \quad (2)$$

where y is the average actual activity of the training set. The relation in equation 3 guarantees the consistency between the results from the minimization of J and the maximization of q^2; therefore we consider only the minimization of cost $J(W, \lambda)$ hereafter.

$$J(W,\lambda)\infty(1-q^2)\frac{\sum_{i=1}^{m}\left\|y^i-y\right\|^2}{2m} \quad (3)$$

2.2.3. Dropout and early stopping

It has been suggested to use at least two hidden layers with a minimum of 250 neurons in each layer (Ma et al., 2015). Since our DNNs have a large number of layers containing many neurons, the increasing number of connections between these neurons makes the corresponding Hessian matrix in the backpropagation process increase complexity by $O(N^2)$, where N is the layer size, and the number of variables in these models can reach into the millions easily. To improve efficiency, JavaDL adopts a process of randomly "dropping out" some neurons in the hidden layers during the training process, which involves temporarily removing the randomly selected neurons from the network, along with all their incoming and outgoing connections (Srivastava et al., 2014). This strategy reduces the complexity of JavaDL's backpropagation computation and ameliorates overfitting as well. Several other methods have also been used to implement the early stopping strategy in JavaDL. The default early stopping criterion of JavaDL is based on the evaluation of the cost function value on a test set.

2.3. Implementation of JavaDL

2.3.1. The overall framework

The overall framework of our system consists of four steps as illustrated in Figure 2. The first step is to curate compound structures and their corresponding bioactivities from literature and other online sources. In the second step we calculate the descriptors of compounds with MOE[3] and CDK.[4] The third step is to clean the data as described below, and in the final step the data is used as input to JavaDL to train the program and build models. We

3 https://www.chemcomp.com/Products.htm
4 https://cdk.github.io/

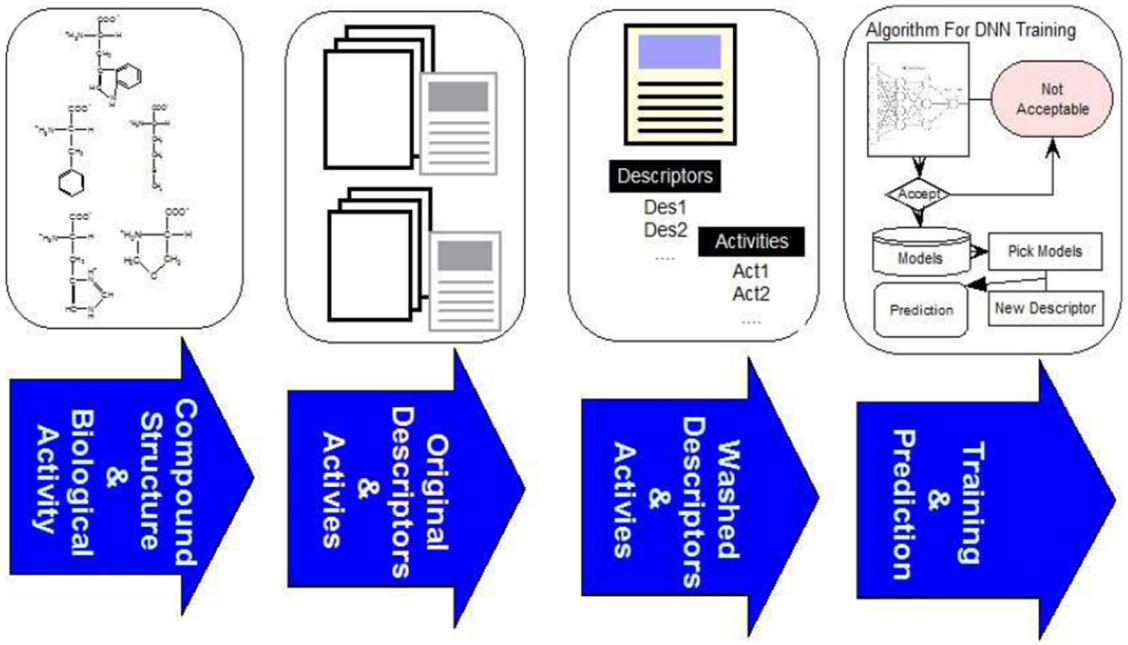

FIGURE 2
The overall framework of JavaDL including data processing and DNN implementation.

chose XML as the format for parameter input to provide extra flexibility for the use of the program: since it is verbose, self-describing and extendable, it can be easily incorporated in other applications. Our software was developed using DL4J, an open-source, distributed deep-learning library. For easy-to-access and easy-to-use, the software package is made available online along with an instruction manual (https://www.imdlab.net/JavaDL/) for download upon request.

2.3.2. Data cleaning

Online or published data are rich but often "dirty," and thus generally need further cleaning and curation. Detection of true outliers is of particular importance; therefore, outlier separation from the main data is included in the data-washing step. The premise here is that similar structures have similar biological activity and that the multidimensional response surface to densely sampled data follows a normal distribution (Maggiora, 2006; Husby et al., 2015). Therefore, we adopted the Pauta criterion to detect abnormal activity-value points, defining a compound with activity value over three standard deviations (3σ) from the mean as an outlier. However, to be statistically significant, the number of compounds similar to the outlier in the descriptor space should be more than 10.

Currently over 5,000 molecular descriptors have been reported (Xue et al., 2004; Ballabio et al., 2009), thus requiring selection of the most relevant and independent descriptors that can represent specific properties of the chemical entities, with smaller numbers of descriptors preferred. Principal components analysis (PCA) has been frequently employed to reduce the dimension of the descriptor matrix (Chan Phooi M'ng and Mehralizadeh, 2016; Zhang et al., 2018). Our initial implementation was based on PCA: given a descriptor matrix $M_{m\times n} = \left(a_1^T, a_2^T, a_3^T \cdots a_n^T\right)$, in which n denotes the number of compounds and a_i^T represents the vector containing m descriptors; the algorithm is then described in the following steps:

Step 1. Calculate the mean of each column $\overline{a_i} = \frac{1}{m}\sum_{j=1}^{m} a_{ij}$ and substract the mean $(\tilde{a_1}, \tilde{a_2}, \tilde{a_3} \cdots \tilde{a_n})$;

Step 2. Calculate the covariance matrix;

Step 3. Calculate the eigenvectors and eigenvalues of the covariance matrix M_{cov}, then order them by eigenvalues from highest to lowest;

Step 4. Select P important eigenvectors from the highest P eigenvalues to compose a feature matrix $M_{feature,n\times p} = \left(v_1, v_2, \cdots v_p\right)$;

Step 5. With the new feature matrix, we derive a new low-dimension matrix *via* $M^T_{feature,n\times p} \times M^T_{m\times n}$ for model building.

However, during the process, the challenge was that the descriptor covariance is too high and P is difficult to select. To overcome this problem, we implemented a simple workaround: instead of considering the covariance of descriptors between two compounds, we considered the correlation between two descriptors. In the descriptor matrix $M_{m\times n} = \left(b_1, b_2 \cdots b_n\right)$, b_i represents the ith descriptor vector. We assume if there is a high correlation between descriptor vectors b_i and b_j, they are not considered orthogonal to each other, since either alone carries enough information to distinguish signals during the training process. Our final implementation with a simple algorithm to exclude the redundant descriptors is as follows:

1. For i = 1 to n
2. For j = i to n

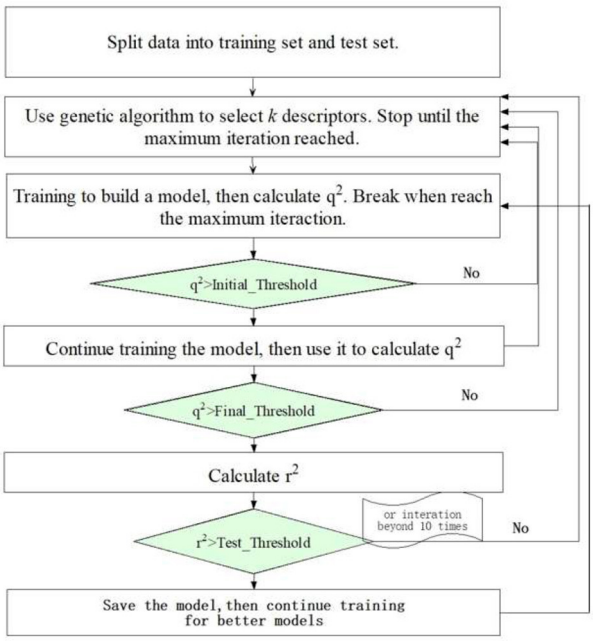

FIGURE 3
The overall flowchart of JavaDL Algorithm.

3. *If* corr(b_i, b_j) > threshold
4. Delete column j from M
5. End if
6. End for

where *corr(b_i, b_j)* is the function calculating the value of correlation between b_i and b_j, and "threshold" denotes the tolerance value for the correlation coefficient. Equation 2 is used to evaluate the correlation directly. This is much more efficient and effective.

2.3.3. Deep learning implementation

Our JavaDL employs deep layer neural networks as the primary training engine. A variable selection procedure was also implemented as an option, in particular for small datasets, where for each predefined number of variables it seeks to optimize the models with the highest correlation coefficient (q^2) (equation 2) for both internal training set and external test set. Figure 3 shows the flowchart of JavaDL training process, in which the backpropagation algorithm is utilized. The most time-consuming step in this algorithm is the calculation of the stochastic gradient descent (SGD) in each layer and the weight (neuron connections) adjustment to minimize the errors derived from our overall squared-error cost function *J(W,λ)* (equation 1) in each iteration. In other words, through the backpropagation algorithm, JavaDL knows how to reduce the error from the cost function, rather than from blindly wandering in space or being guided by a scalar quantity such as the random walk in the Metropolis algorithm.

Generally, the time complexity is determined by the number of descriptors chosen and the structure of the neural network. Let k denote the number of descriptors picked from K total descriptors, m, n the number of layers and neurons, respectively, in each layer (assume each layer contains the same number of neurons), and d the size of the mini-batch representing the minimum iteration time in the training process, the time complexity is then $O(C_K^k mnd)$.

3. Results and discussion

3.1. Comparison with other machine learning programs

Before training, we first performed data wash as described previously and then divided the data into two sets using the Sphere Exclusion (SE) algorithm (Golbraikh et al., 2003; Zhang et al., 2006b). For the Caco-2 dataset, one is for training with 80 compounds and the other is for testing with 20 compounds. It should be noted that the division process is executed each time after changing the descriptors during iterations. The same procedure was followed for the hERG dataset, with 133 compounds for training and 14 compounds for testing. Results from JavaDL are compared with those obtained by kNN, SVM, Random Forest, XGBoost, and CNN (1D) methods as shown in Tables 1, 2. Also as illustrated in Figure 4 our models show high correlation between the actual and predicted activities for the test sets. The statistical parameters used to assess models, including q^2 and r^2, demonstrate the comparable performance of JavaDL to other machine learning algorithms, particularly when datasets are small. For the first three ML methods list in Tables 1, 2, we directly adopted sklearn libraries, such as sklearn.neighbors.KNeighborsRegressor, sklearn.svm.SVR and sklearn.ensemble.RandomForestRegressor, for the regression tasks. The default optimized parameters were used unless otherwise stated. The architectures and parameters of CNN and our own JavaDL are described in the Supporting Information.

We also examined our JavaDL program using a large data set from the Merck Molecular Activity Challenge as described above. Since this would make the training computationally intensive, we have implemented a parallel computing function for JavaDL so that it can efficiently handle big data with a large number of compounds and high dimensional space (number of descriptors). With 250 compounds in the test set, our model achieved robust predictions with $q^2 = 0.99$ and $r^2 = 0.65$. These results demonstrate significant superiority to all published models released by the Merck Molecular Activity Challenge where r^2 was below 0.49. Our models and predictions are shown in Figure 5.

3.2. Predicting cancer response to drugs

As discussed, it is critical to predict individual cancer cell response to different drugs. Such information can provide insight into what drugs can be used to treat what type of cancer with the highest sensitivity. With TNBC screening data from NCATS and our own experiments, we performed data pre-processing, and the total number of compounds for each of the four TNBC cell lines was reduced to 186, 163, 146, and 172, respectively. In each group we randomly selected 40 compounds as the external test set, and set $q^2_{threshold} = 0.60$ and $r^2_{threshold} = 0.55$ as criteria to determine if a model is acceptable. Since the data is sufficiently large and JavaDL can handle big data efficiently, we took into

TABLE 1 Method comparison with the Caco-2 dataset.

Caco-2	KNN		SVM		Random Forest		XGBoost		CNN		JavaDL	
Model	q^2	r^2	q^2	r^2	q^2	r^2	q^2	r^2	q^2	r^2	q^2	r^2
1	0.54	0.78	0.50	0.80	0.85	0.61	0.83	0.65	0.70	0.81	0.80	0.78
2	0.54	0.78	0.50	0.80	0.73	0.50	0.83	0.55	0.70	0.80	0.80	0.78
3	0.46	0.77	0.50	0.80	0.85	0.59	0.90	0.70	0.70	0.80	0.80	0.73
4	0.51	0.76	0.47	0.79	0.86	0.67	0.83	0.60	0.77	0.81	0.70	0.77
5	0.48	0.77	0.47	0.79	0.85	0.23	0.83	0.65	0.65	0.81	0.65	0.75
6	0.51	0.76	0.47	0.79	0.86	0.30	0.75	0.55	0.70	0.82	0.70	0.72
7	0.51	0.76	0.43	0.76	0.89	0.65	0.91	0.80	0.40	0.65	0.69	0.80
8	0.42	0.72	0.43	0.76	0.85	0.60	0.85	0.80	0.77	0.81	0.62	0.75
9	0.42	0.72	0.43	0.76	0.84	0.43	0.88	0.65	0.89	0.80	0.65	0.60

TABLE 2 Method comparison with the hERG dataset.

hERG	KNN		SVM		Random		XGBoost		CNN		JavaDL	
Model	q^2	r^2	q^2	r^2	q^2	r^2	q^2	r^2	q^2	r^2	q^2	r^2
1	0.48	0.70	0.41	0.87	0.81	0.46	0.80	0.78	0.67	0.80	0.75	0.72
2	0.48	0.70	0.43	0.87	0.82	0.37	0.77	0.58	0.72	0.80	0.80	0.83
3	0.45	0.68	0.43	0.87	0.82	0.31	0.88	0.73	0.61	0.38	0.80	0.83
4	0.45	0.68	0.46	0.82	0.81	0.18	0.90	0.77	0.77	0.68	0.80	0.83
5	0.45	0.68	0.46	0.82	0.84	0.81	0.85	0.65	0.73	0.80	0.90	0.67
6	0.50	0.66	0.45	0.81	0.80	0.65	0.80	0.72	0.72	0.80	0.90	0.73
7	0.50	0.66	0.52	0.81	0.81	0.55	0.79	0.80	0.73	0.62	0.95	0.63
8	0.50	0.66	0.52	0.81	0.83	0.67	0.82	0.75	0.70	0.59	0.90	0.73
9	0.48	0.65	0.44	0.81	0.80	0.47	0.85	0.80	0.62	0.80	0.93	0.63
10	0.48	0.65	0.44	0.81	0.76	0.55	0.80	0.78	0.73	0.80	0.90	0.67

account all descriptors of the compounds for model building and final predictions. The measured and predicted activities with our best models are shown in Figure 6. The curated datasets for four TNBC cell lines with training source code are available at https://www.imdlab.net/JavaDL/.

3.3. Discussion

In this study, we present a new deep-learning method for predicting the efficacy of small-molecule anti-cancer therapeutic agents. The novelty of this implementation lies in multiple aspects. The first is that prediction of drug response has been challenging due to the intrinsic complexity of cancer cells and various unknown cell survival mechanisms (Hu et al., 2016; Tan and Zhang, 2016); DNN is exactly designed to handle this type of "black-box" problems through a continuous self-learning process. Second, we have implemented a variety of strategies to address several frequent issues including data cleaning, overfitting, outliers, and program complexity. Third, we designed a new scoring activation function and cost function. Fourth, JavaDL is able to predict the response of cancer cells to drugs using only structural information from compounds, and it can be easily expanded to include cell gene profile change upon drug treatment for predictions. Finally, DNN is combined with variable selection in this parallelized implementation and we could identify specific descriptors that are critical for activities. Such information is tremendously helpful to guide actual rational drug design.

Using two very well studied datasets (hERG and Caco-2) as benchmarks, we obtained robust results demonstrating the superiority of our multiple-layer DNN techniques to several machine learning techniques such as kNN and SVM. As expected, its predictive capability is comparable to some popular methods including Random Forest, XGBoost, and CNN. Of note, such observations are not only related to modeling methods and training parameters, but also highly dependent on the datasets used. In addition, our models could effectively capture the abstract relation between the structural features of chemical compounds and their activities. Also as shown, JavaDL was successfully employed to predict the response of aggressive TNBC cell lines to different drugs, which has been a challenge in personalized cancer therapy. Moreover, this indicates that JavaDL may potentially be used as a general screening tool to predict activities of novel compounds in different cancer cells, and thus helping to lower costs of anticancer

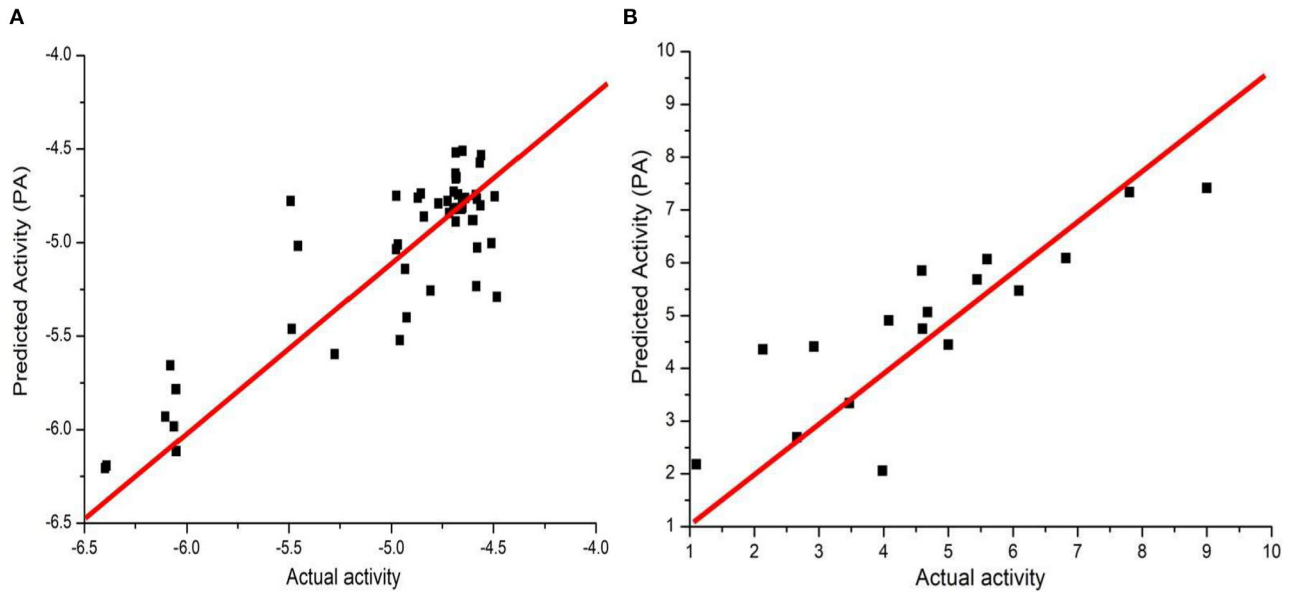

FIGURE 4
Predicted vs. actual activities obtained for test sets. **(A)** Prediction of Caco-2 permeability using JavaDL with five descriptors: BCUT_ PEOE_3, GCUT_SLOGP_1, mr, a_base, vsa_base. **(B)** Prediction of hERG activity with five descriptors: GCUT_PEOE_3, reactive, SlogP_V SA1, SlogP_V SA9, and vdw_area.

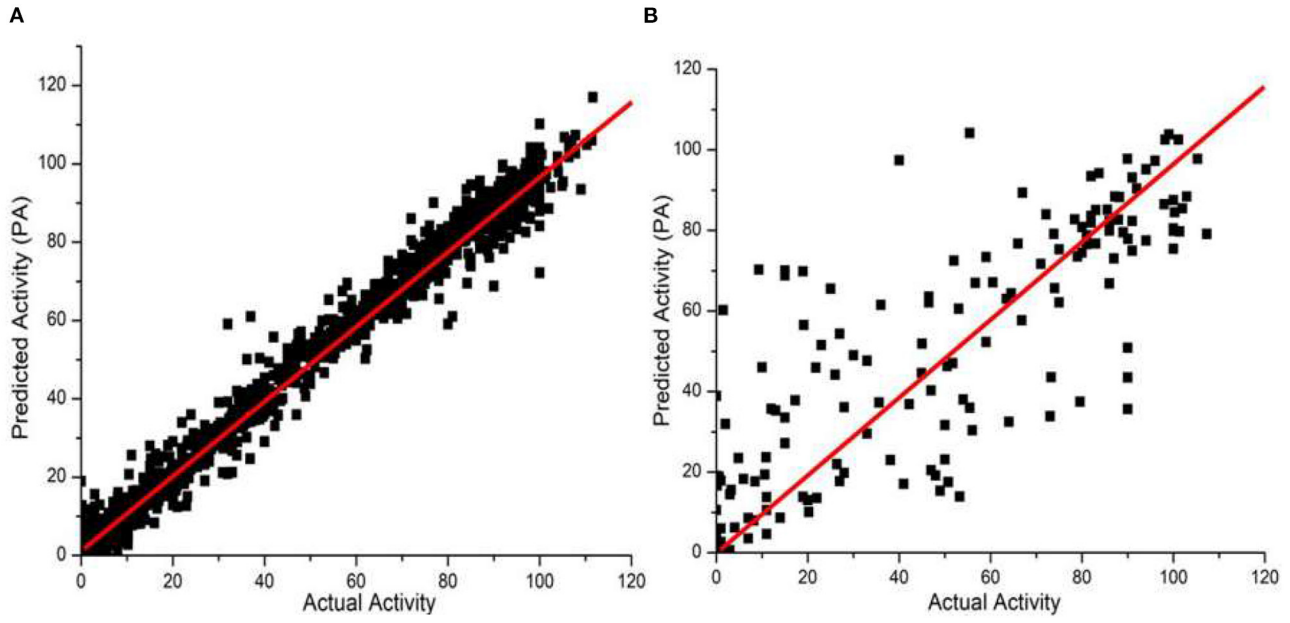

FIGURE 5
Predicted vs. actual biological activities for the Merck Molecular Activity Challenge big data set in Kaggle competition. The model was built using 1,569 compounds with 4,505 descriptors. Among them 250 compounds are picked out randomly for test predictions, and the best model derived **(A)** $q^2 = 0.99$, shown in the left panel, and **(B)** $r^2 = 0.65$, shown in the right panel.

therapeutics screening. It is mostly worth to mention that, as one of the signified features of deep learning, our prediction of the Merck Molecular Activity Challenge dataset demonstrated the capacity of JavaDL to handle big data problems with a large number of points in a high dimensional space.

As mentioned, the variable selection feature in this implementation is particularly useful for rational molecular design, and identification of selected descriptors, when combined with chemical structures, can be employed to interpret the molecular structure-activity relationship and guide lead optimization in drug development. For instance, the descriptors selected in the Caco-2 permeability model are PEOE Charge BCUT, GCUT logP, molecular refractivity, number of basic atoms, and Van der Waals basic surface area. It makes sense that all of these properties, in

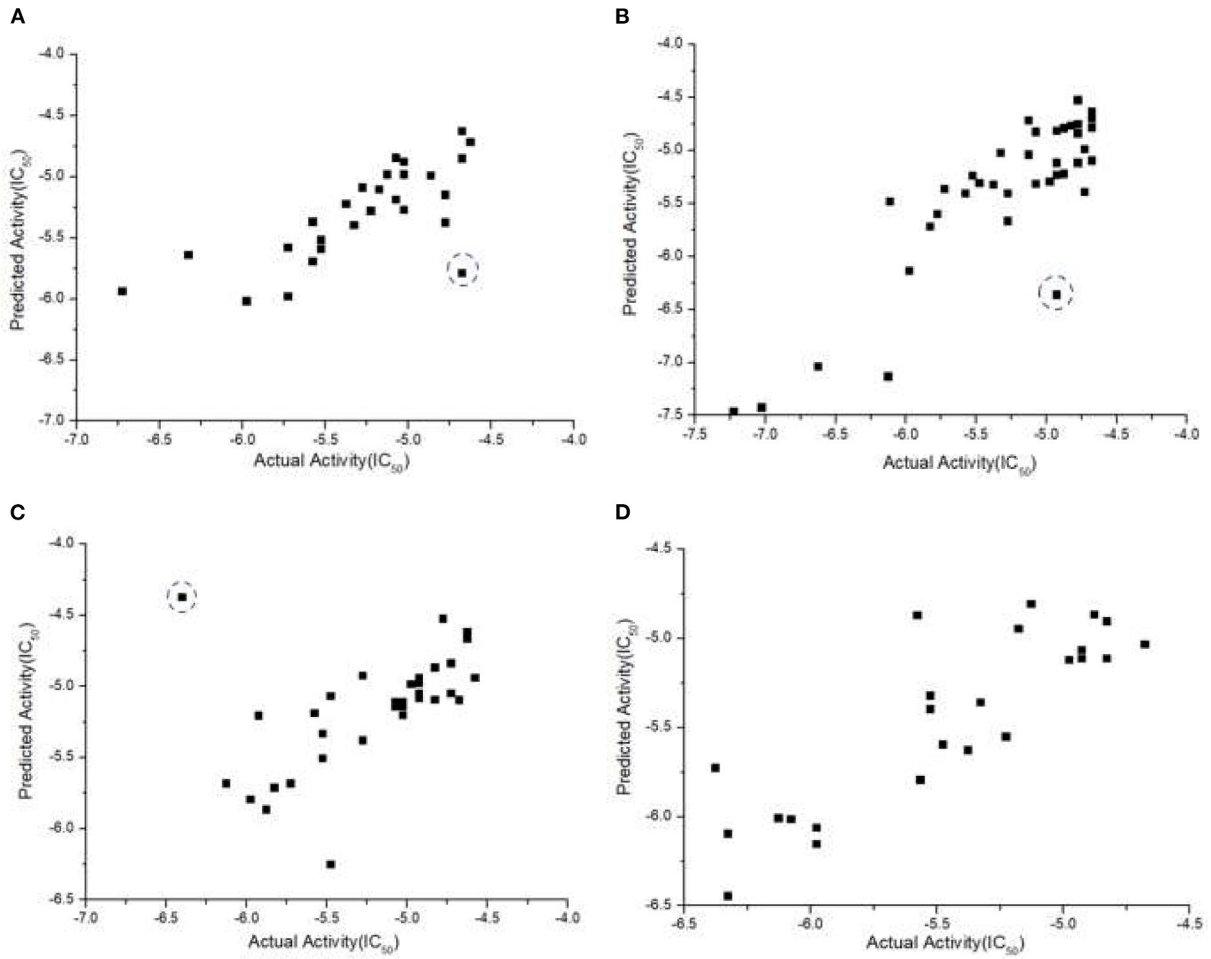

FIGURE 6
Predicted vs. actual activities (IC_{50}) of drugs in TNBC, including **(A)** HCC-1937, **(B)** MDA-MB-231, **(C)** MDA-MB-453, and **(D)** MDA-MB-436. The red-circled compound 3-Methyladenine as an outlier was not accurately predicted. Further analysis shows that no compound in the training set is similar enough to 3-Methyladenine while there are always similar compounds to the accurately predicted ones (blue-circled Afatinib and UNC-0638).

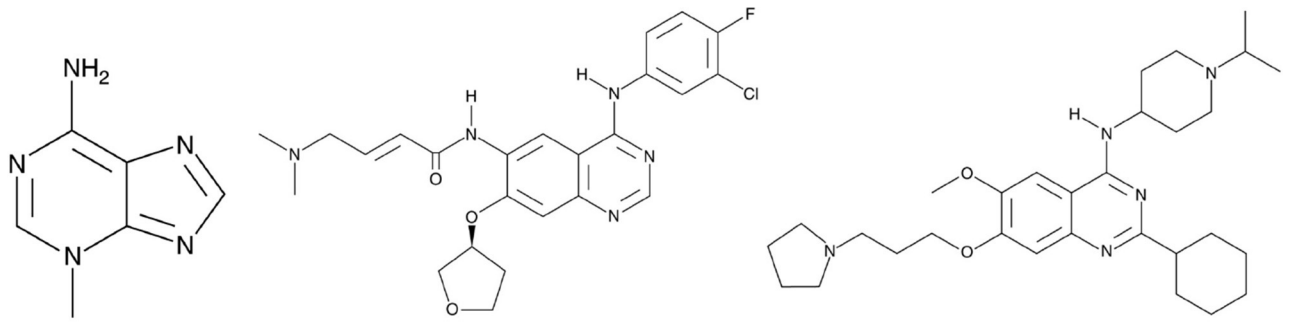

FIGURE 7
From left to right is 3-Methyladenine, Afatinib and UNC-0638, respectively. 3-Methyladenine was not accurately predicted while the other two compounds obtained accurate prediction of their activity in TNBC.

particular logP, charges, and molecular refractivity, are highly correlated with the permeability of the compounds. While GCUT logP contributes positively to permeability, the other four have negative impact on permeability. Such relationship is demonstrated by the comparison of quinidine (Caco-2 $P = -4.69$) and ranitidine (Caco-2 $P = -6.31$) where quinidine has much higher GCUT logP while the other four properties have lower values than ranitidine. Similar observations have been obtained in the case of hERG and TNBC studies.

Some cheminformatics practitioners contend that machine learning has not fulfilled its promise in predicting biological activity (Golbraikh and Tropsha, 2002; Kovalishyn et al., 2018).

Poor predictivity is a problem in most models, and there are a variety of possible reasons for this including the incorrect assignment of molecular properties, chance correlation, rough response surfaces, and overtraining (Robichaux et al., 2018; Jumper et al., 2021). With JavaDL we obtained significant improvement of prediction over other machine-learning models, with most of the prediction errors below one unit, which is usually acceptable in drug design and development. For a few compounds our prediction is not ideal. As shown in Figure 6C, the compound 3-methyladenine (structure shown in Figure 7) is identified as an outlier with the worst prediction in MDA-MB-453 cell lines. To elucidate the reason, we calculate the similarity of 3-methyladenine to other compounds in the training dataset based on their Euclidean distance. We found that the distance to the most similar compound (theobromine) is $SD_{3-methyladenine} = 1.49$. Similarly, we randomly selected two compounds (afatinib and UNC-0638 shown in Figure 7) that were accurately predicted (Figure 6C) and computed their corresponding Euclidean distance to the most similar compounds. We obtained $SD_{afatinib} = 0.68$ and $SD_{UNC-0638} = 0.80$, respectively. Using the Tanimoto Coefficient (TC), we observed the similar trend: for 3-methyladenine the highest TC = 0.40 (to theobromine); for afatinib and UNC-0638, TC = 0.61 (to pelitinib) and TC = 0.50 (to XL-647), respectively. This indicates that when there are similar compounds in the training set, the prediction is relatively accurate, in agreement with the general principle of QSAR that similar structures tend to have similar activities.

Our DNN-based JavaDL is not limited to drug response predictions; rather it can be very easily used for studies of large genomic and bioinformatics data to predict gene-disease association, identify biomarkers with whole genome profiling, and develop treatment algorithms based on patient response to drugs. Our promising results shown here suggest that JavaDL can be used as a general tool for the discovery and design of biologically active agents as well as for many other types of biomedical research.

4. Availability

The program is freely available at https://www.imdlab.net/JavaDL/.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary material, and further inquiries can be directed to the corresponding author.

Author contributions

SZ and BH conceived the idea and designed the algorithm. BH developed the program. SZ, BH, LF, and RC tested program and analyzed the data. All authors contributed to the article and approved the submitted version.

Funding

This work has been partially supported by CPRIT RP170333, NIH 1R01CA225955, DOD W81XWH-20-PCRP-IDA, and MD Anderson Cancer Center Bridge Funding Program to SZ.

Acknowledgments

We thank Eric Zhang for helping build and maintain the website. Special thanks to the Deeplearning4j and CDK toolkit development teams for providing us academic licenses. We also thank the high-performance computing (HPC) resources provided by MD Anderson RIS and Texas Advanced Computing Center (TACC).

Conflict of interest

Author RC is currently employed by the company Eurofins Beacon Discovery.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frai.2023.1069353/full#supplementary-material

References

Acharya, U. R., Oh, S. L., Hagiwara, Y., Tan, J. H., and Adeli, H. (2018). Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals. *Comput. Biol.Med.* 100, 270–278. doi: 10.1016/j.compbiomed.2017.09.017

Adam, G., Rampášek, L., Safikhani, Z., Smirnov, P., Haibe-Kains, B., and Goldenberg, A. (2020). Machine learning approaches to drug response prediction: challenges and recent progress. *NPJ Precis. Oncol.* 4, 19. doi: 10.1038/s41698-020-0122-1

Angermueller, C., Pärnamaa, T., Parts, L., Stegle, O. (2016). Deep learning for computational biology. *Mol. Syst. Biol.* 12, 878. doi: 10.15252/msb.20156651

Bajorath, J. (2017). Representation and identification of activity cliffs. *Expert Opin. Drug Discov.* 12, 879–883. doi: 10.1080/17460441.2017.1353494

Ballabio, D., Manganaro, A., Consonni, V., Mauri, A. (2009). Introduction to MOLE DB - on-line molecular descriptors database. *Match Commun. Math. Comput. Chemist.* 62, 199–207.

Baskin, I. I. (2020). The power of deep learning to ligand-based novel drug discovery. *Expert Opin. Drug Discov.* 15, 755–764. doi: 10.1080/17460441.2020.1745183

Bizzego, A., Bussola, N., Chierici, M., Maggio, V., Francescatto, M., Cima, L., et al. (2019). Evaluating reproducibility of AI algorithms in digital pathology with DAPPER. *PLoS Comput. Biol.* 15:e1006269. doi: 10.1371/journal.pcbi.1006269

Burden, F., and Winkler, D. (2008). Bayesian regularization of neural networks. *Methods Mol. Biol.* 458, 25–44. doi: 10.1007/978-1-60327-101-1_3

Carpenter, K., and Huang, X. (2017). Is it a prime time for ai-powered virtual drug screening? *EC Pharmacol. Toxicol.* SI, 16–17.

Chan Phooi M'ng, J., and Mehralizadeh, M. (2016). Forecasting east asian indices futures via a novel hybrid of wavelet-PCA denoising and artificial neural network models. *PLoS ONE* 11:e0156338. doi: 10.1371/journal.pone.0156338

Chaudhari, R., Tan, Z., Huang, B., and Zhang, S. (2017). Computational polypharmacology: a new paradigm for drug discovery. *Expert Opin. Drug Discov.* 12, 279–291. doi: 10.1080/17460441.2017.1280024

Du, Y., Pan, Y., Wang, C., and Ji, J. (2018). Biomedical semantic indexing by deep neural network with multi-task learning. *BMC Bioinformatics* 19, 502. doi: 10.1186/s12859-018-2534-2

Du-Cuny, L., Chen, L., and Zhang, S. (2011). A critical assessment of combined ligand- and structure-based approaches to HERG channel blocker modeling. *J. Chem. Inf. Model.* 51, 2948–2960. doi: 10.1021/ci200271d

Du-Cuny, L., Song, Z., Moses, S., Powis, G., Mash, E. A., Meuillet, E. J., et al. (2009). Computational modeling of novel inhibitors targeting the Akt pleckstrin homology domain. *Bioorg. Med. Chem.* 17, 6983–6992. doi: 10.1016/j.bmc.2009.08.022

Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T. F. G., Qin, C., et al. (2018). *De novo structure prediction with deep-learning based scoring. In Thirteenth Critical Assessment of Techniques for Protein Structure Prediction.*

Freedman, D. H. (2019). Hunting for new drugs with AI. *Nature* 576, S49–S53. doi: 10.1038/d41586-019-03846-0

Gawehn, E., Hiss, J. A., and Schneider, G. (2016). Deep learning in drug discovery. *Mol. Inform.* 35, 3–14. doi: 10.1002/minf.201501008

Golbraikh, A., Shen, M., Xiao, Z., Xiao, Y. –D., Lee, K. –H., Tropsha, A., et al. (2003). Rational selection of training and test sets for the development of validated QSAR models. *J. Comput. Aided Mol. Des.* 17, 241–253. doi: 10.1023/A:1025386326946

Golbraikh, A., and Tropsha, A. (2002). Beware of q2! *J. Mol. Graph. Model.* 20, 269–276. doi: 10.1016/S1093-3263(01)00123-1

Gurovich, Y., Hanani, Y., Bar, O., Nadav, G., Fleischer, N., Gelbman, D., et al. (2019). Identifying facial phenotypes of genetic disorders using deep learning. *Nat. Med.* 25, 60–64. doi: 10.1038/s41591-018-0279-0

Heffernan, R., Paliwal, K., Lyons, J., Dehzangi, A., Sharma, A., Wang, J., et al. (2015). Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning. *Sci. Rep.* 5, 11476. doi: 10.1038/srep11476

Hinton, G. (2014). Where do features come from? *Cogn. Sci.* 38, 1078–1101. doi: 10.1111/cogs.12049

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Netw.* 4, 251–257. doi: 10.1016/0893-6080(91)90009-T

Hu, B., Wang, Q., Wang, Y. A., Hua, S., Sauvé, C. -E. G., Ong, D., et al. (2016). Epigenetic activation of WNT5A drives glioblastoma stem cell differentiation and invasive growth. *Cell.* 167, 1281–1295. e18. doi: 10.1016/j.cell.2016.10.039

Humphreys, I. R., Pei, J., Baek, M., Krishnakumar, A., Anishchenko, I., Ovchinnikov, S., et al. (2021). Computed structures of core eukaryotic protein complexes. *Science.* 374:abm4805. doi: 10.1126/science.abm4805

Husby, J., Bottegoni, G., Kufareva, I., Abagyan, R., and Cavalli, A. (2015). Structure-based predictions of activity cliffs. *J. Chem. Inf. Model.* 55, 1062–1076. doi: 10.1021/ci500742b

Indraswari, R., Kurita, T., Arifin, A. N., Suciati, N., Astuti, E. R. (2019). Multi-projection deep learning network for segmentation of 3D medical images. *Pattern Recognit. Lett.* 125, 791–797. doi: 10.1016/j.patrec.2019.08.003

Jang, Y., Son, J., Park, K. H., Park, S. J., and Jung, K. -H. (2018). Laterality classification of fundus images using interpretable deep neural network. *J. Digit. Imaging* 31, 923–928. doi: 10.1007/s10278-018-0099-2

Jeyaraj, P. R., and Nadar, E. R. S. (2019). Computer-assisted medical image classification for early diagnosis of oral cancer employing deep learning algorithm. *J. Cancer Res. Clin. Oncol.* 145, 829–837. doi: 10.1007/s00432-018-02834-7

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., et al. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature* 596, 583–589. doi: 10.1038/s41586-021-03819-2

Kang, G., Li, J., and Tao, D. (2018). Shakeout: a new approach to regularized deep neural network training. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 1245–1258. doi: 10.1109/TPAMI.2017.2701831

Kovalishyn, V., Grouleff, J., Semenyuta, I., Sinenko, V. O., Slivchuk, S. R., Hodyna, D., et al. (2018). Rational design of isonicotinic acid hydrazide derivatives with antitubercular activity: Machine learning, molecular docking, synthesis and biological testing. *Chem. Biol. Drug Des.* 92, 1272–1278. doi: 10.1111/cbdd.13188

Lee, G., Park, C., and Ahn, J. (2019). Novel deep learning model for more accurate prediction of drug-drug interaction effects. *BMC Bioinformatics* 20, 415. doi: 10.1186/s12859-019-3013-0

Lo, Y.-C., Rensi, S. E., Torng, W., and Altman, R. B. (2018). Machine learning in chemoinformatics and drug discovery. *Drug Discov. Today* 23, 1538–1546. doi: 10.1016/j.drudis.2018.05.010

Ma, J., Sheridan, R. P., Liaw, A., Dahl, G. E., and Svetnik, V. (2015). Deep Neural Nets as a Method for Quantitative Structure-Activity Relationships. *J. Chem. Inf. Model.* 55, 263–274. doi: 10.1021/ci500747n

Ma, J., Song, Y., Tian, X., Hua, Y., Zhang, R., and Wu, J. (2019). Survey on deep learning for pulmonary medical imaging. *Front. Med.* 14, 450–469. doi: 10.1007/s11684-019-0726-4

Maggiora, G. M. (2006). On outliers and activity cliffs - why QSAR often disappoints. *J. Chem. Inf. Model.* 46, 1535–1535. doi: 10.1021/ci060117s

Murphy, R. F. (2011). An active role for machine learning in drug development. *Nat. Chem. Biol.* 7, 327–330. doi: 10.1038/nchembio.576

Reddy, A. S., Tan, Z., and Zhang, S. (2014). Curation and analysis of multitargeting agents for polypharmacological modeling. *J. Chem. Inf. Model.* 54, 2536–2543. doi: 10.1021/ci500092j

Robichaux, J. P., Elamin, Y. Y., Tan, Z., Carter, B. W., Zhang, S., Liu, S., et al. (2018). Mechanisms and clinical activity of an EGFR and HER2 exon 20-selective kinase inhibitor in non-small cell lung cancer. *Nat. Med.* 24, 638–646. doi: 10.1038/s41591-018-0007-9

Segler, M. H. S., Preuss, M., and Waller, M. P. (2018). Planning chemical syntheses with deep neural networks and symbolic AI. *Nature,* 555, 604–610. doi: 10.1038/nature25978

Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., et al. (2020). Improved protein structure prediction using potentials from deep learning. *Nature*, 577, 706–710. doi: 10.1038/s41586-019-1923-7

Smalley, E. (2017). AI-powered drug discovery captures pharma interest. *Nat. Biotechnol.* 35, 604–605. doi: 10.1038/nbt0717-604

Smith, J. S., Roitberg, A. E., and Isayev, O. (2018). Transforming computational drug discovery with machine learning and AI. *ACS Med. Chem. Lett.* 9, 1065–1069. doi: 10.1021/acsmedchemlett.8b00437

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15, 1929–1958. Available online at: http://jmlr.org/papers/v15/srivastava14a.html

Steventon, A., Deeny, S. R., Keith, J., and Wolters, A. T. (2019). New AI laboratory for the NHS. *BMJ* 366, l5434. doi: 10.1136/bmj.l5434

Tan, Z., Chaudhai, R., and Zhang, S. (2016b). Polypharmacology in drug development: a minireview of current technologies. *Chem. Med. Chem.* 11, 1211–1218. doi: 10.1002/cmdc.201600067

Tan, Z., Chen, L., and Zhang, S. (2016a). Comprehensive modeling and discovery of mebendazole as a Novel TRAF2- and NCK-interacting Kinase Inhibitor. *Sci. Rep.* 6, 33534. doi: 10.1038/srep33534

Tan, Z., and Zhang, S. (2016). Past, present, and future of targeting ras for cancer therapies. *Mini Rev. Med. Chem.* 16, 345–357. doi: 10.2174/1389557515666151001154111

Tropsha, A., Gramatica, P., and Gombar, V. K. (2003). The importance of being earnest: validation is the absolute essential for successful application and interpretation of QSPR models. *Qsar Comb. Sci.* 22, 69–77. doi: 10.1002/qsar.200390007

Tsao, A. S., Wistuba, I., Xia, D., Byers, L., Diao, L., Wang, J., et al. (2017). Response of germline and somatic smoothened (SMO) mutations in non-small cell lung cancer (NSCLC) to hedgehog inhibitor vismodegib. *J. Clin. Oncol.* 2017, 35. doi: 10.1200/JCO.2017.35.15_suppl.9062

Vamathevan, J., Clark, D., Czodrowski, P., Dunham, I., Ferran, E., Lee, G., et al. (2019). Applications of machine learning in drug discovery and development. *Nat. Rev. Drug Discovery* 18, 463–477. doi: 10.1038/s41573-019-0024-5

Watanabe, C., Watanabe, H., Fukuzawa, K., Parker, L. J., Okiyama, Y., Yuki, H., et al. (2017). Theoretical analysis of activity cliffs among benzofuranone-class pim1 inhibitors using the fragment molecular orbital method with molecular mechanics poisson-boltzmann surface area (FMO+MM-PBSA) approach. *J. Chem. Inf. Model.* 57, 2996–3010. doi: 10.1021/acs.jcim.7b00110

Winkler, D. A., and Le, T. C. (2017). Performance of deep and shallow neural networks, the universal approximation theorem, activity cliffs, and QSAR. *Mol. Inform.* 36, 160018–160024. doi: 10.1002/minf.201781141

Woo, M. (2019). An AI boost for clinical trials. *Nature* 573, S100–S102. doi: 10.1038/d41586-019-02871-3

Xue, Y., Li, Z. R., Yap, C. W., Sun, L. Z., Chen, X., and Chen, Y. Z. (2004). Effect of molecular descriptor feature selection in support vector machine classification of pharmacokinetic and toxicological properties of chemical agents. *J. Chem. Inf. Comput. Sci.* 44, 1630–1638. doi: 10.1021/ci049869h

You, J. Y., McLeod, R. D., and Hu, P. Z. (2019). Predicting drug-target interaction network using deep learning model. *Comput. Biol. Chem.* 80, 90–101. doi: 10.1016/j.compbiolchem.2019.03.016

Zeng, X., Zhu, S., Liu, X., Zhou, Y., Nussinov, R., and Cheng, F. (2019). deepDR: a network-based deep learning approach to in silico drug repositioning. *Bioinformatics* 35, 5191–5198. doi: 10.1093/bioinformatics/btz418

Zhang, R., Tao, J., Lu, R., and Jin, Q. (2018). Decoupled ARX and RBF neural network modeling using PCA and GA optimization for nonlinear distributed parameter systems. *IEEE Trans. Neural Netw. Learn Syst.* 29, 457–469. doi: 10.1109/TNNLS.2016.2631481

Zhang, S., Golbraikh, A., Oloff, S., Kohn, H., and Tropsha, A. (2006a). A novel automated lazy learning QSAR (ALL-QSAR) approach: Method development, applications, and virtual screening of chemical databases using validated ALL-QSAR models. *J. Chem. Inf. Model.* 46, 1984–1995. doi: 10.1021/ci060132x

Zhang, S., Golbraikh, A., and Tropsha, A. (2006b). Development of quantitative structure-binding affinity relationship models based on novel geometrical chemical descriptors of the protein-ligand interfaces. *J. Med. Chem.* 49, 2713–2724. doi: 10.1021/jm050260x