Exploration and Augmentation of Pharmacological Space via Adversarial Auto-encoder Model for Facilitating Kinase-centric Drug Development

Xinyu Bai (✉ xybai@bjmu.edu.cn)

Peking University Health Science Centre <https://orcid.org/0000-0001-7526-7461>

Yuxin Yin

Peking University Health Science Centre <https://orcid.org/0000-0003-4504-8035>

Research article

Keywords: Data augmentation, AAE, cheminformatics, deep learning, kinase

Posted Date: November 17th, 2020

DOI: <https://doi.org/10.21203/rs.3.rs-105889/v1>

License: This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Abstract

Predicting drug-protein interactions (DPIs) is of great importance for drug discovery and repositioning, yet still challenging mainly due to the sparse nature of DPI matrixes, resulting in poor generalization performance. Hence, unlike typical DPI prediction models which focused on representation learning or model selection, we propose a deep neural network-based strategy, PCM_AAE, that re-explores and augments the pharmacological space of kinase inhibitors by introducing adversarial auto-encoder model (AAE) to improve the generalization of the prediction model. To complete the pharmacological space, we constructed Ensemble of PCM-AAE (EPA), an ensemble model that quickly and accurately yields quantitative predictions of binding affinity between any human kinase and inhibitor. In rigorous internal validation, EPA showed excellent performance, consistently outperforming the model trained with the imbalanced set, especially for targets with relatively fewer training data points. Improved prediction accuracy of EPA to external datasets again demonstrated enhanced generalization ability of EPA that could gracefully handle previously unseen kinases or inhibitors. Further analysis showed promising potential when EPA was directly applied to virtual screening and off-target prediction, exhibiting the practicality of the EPA model in hit prediction. Our strategy is expected to facilitate kinase-centric drug development, as well as to solve more challenging prediction problems with insufficient data points.

Introduction

Determining the drug-protein interactions (DPI) is one key aspect in drug development, contributing to both understanding complicated mechanism of action (MoA) of drugs and discovering novel inhibitors of proteins [1]. In particular, for the homologous proteins with high genetic conservation and similar structures (eg. Protein kinase family), it is necessary and challenging to detect the potential protein-inhibitor profiling which may bring therapeutic effects or side effects. However, the number of activity matrixes is still sparse due to the time-consuming and cost-intensive efforts for establishing and conducting functional assays [2]. To increase the efficiency of this process, various structure- and ligand-based computational approaches have been developed for pre-screening [3–5], yet their predictive power are limited in many cases partly because the accuracy of these approaches depend heavily on decent number of active compounds towards the target [6, 7].

In recent years, a revival of deep learning (DL)-derived methods with the state-of-art application in the fields of natural language processing, speech recognition and computer vision [8] has attracted academic and industrial attention and been increasingly applied to address various challenges. In chemistry, a lot of efforts based on deep learning have also been devoted to accelerate the drug development process, such as predicting compound properties, planning chemical syntheses, drug repositioning and generating novel molecules for *de novo* drug design [9–13]. Apart from the model where representation of small molecules derives from pre-defined descriptors (eg. ECFP [14]), lately there is also some work on learnable representations of molecules, such as Mol2vec [15], GraphConv [16], and druGAN [17].

DPI prediction is usually approached as a classification task, the main efforts have been focusing on novel prediction algorithm establishment or representation of molecules and proteins [18, 19]. It should be noted that training dataset is also vitally important, as experimental uncertainty and the sparse nature of DPI annotations set limits for DL models greatly [20, 21]. However, to our best knowledge, studies on improving the quality of training data and supplementing activity annotations are few, especially for regression tasks. Encouragingly, new strategies for data augmentation based on generative models lead to clearer margin among different data categories, and have been applied widely to image recognition, anomaly detection, emotion classification and other fields [22–24]. Inspired by GAN and AAE models, here we proposed two semi-supervised strategies PCM-GAN and PCM-AAE to expand data space, which were then successfully applied to protein kinase-inhibitor dataset. We then trained ensemble of PCM-AAE (EPA) for ensemble of expanded data space using random forest, and tested the model on internal and external datasets. The results showed that EPA boosted the performance of models that directly fed with imbalanced data in rigorous internal validations and external validations. Moreover, the reliable quantitative predictions of EPA to new kinases and new molecules suggested that EPA can support predictions across large-scale data, and thus aid kinase-profiling prediction of inhibitors or serve as an efficient alternative of molecular docking algorithm for virtual screening.

Methods

Data collection, preprocessing and analysis

Seven kinase bioactivity datasets were collected for generating and validating the prediction models: Christmann-Franck’s dataset [25], Kinase SARfari [26], PKIS1 [27], PKIS2 [28], MRC (MRC PPU, <https://www.ppu.mrc.ac.uk/>), Metz’s dataset [29] and Davis’s dataset [30]. The number of molecules and kinases in particular dataset is summarized in Table 1. The proteins in the datasets were grouped by their Uniprot Identifier, while compounds were integrated based on their canonical SMILES. The sequence of kinase domain was extracted from the full sequence of each kinase.

Table 1
Statistics of datasets

Datasets	Compounds	Kinases	Data points
Christmann-Franck’s	2000	196	85958
Kinase SARfari	9977	20	20508
PKIS1	354	187	66197
PKIS2	333	302	1437
MRC	217	116	27154
Metz’s	1450	161	99480
Davis’s	70	336	6642

Models were developed based on Christmann-Franck's dataset and validated on other 6 datasets. Since the data are heterogeneous and obtained by various experimental methods, we conducted data standardization: (1) we first selected the data with units of pKi, pKd, Kd, Ki, and POC (Percent of Control) to retrieve more reliable data representing the protein-ligand binding affinity; (2) we then applied data standardization rules as described previously to the curated datasets [25]; (3) Finally, all measured values were $-\log_{10}$ transformed. The cutoff to distinguish active compounds from inactive ones was set at 6 (corresponding to measured value of 1 μM). Additionally, the coefficient of variation (CV) of repeated measurements were assessed for each dataset to remove duplicates. Measurements with CV value over 0.05 were removed or else one of the measurements was kept randomly.

Data representations

Mol2Vec and ProtVec models were combined to obtain vector embeddings of ligands and proteins respectively [15, 31]. The ligand corpus was obtained from ZINC database version 15 [32], where the canonical SMILES representation of each ligand was transformed to a list of ordered atom identifiers as a "molecule sentence" through Morgan algorithm. To embed the "molecule sentence" of a specific molecule, skip-gram was used to convert each atom identifier to a 100- or 300- dimensional vector. Molecules were represented as summed vectors of its atom identifiers. The protein corpus of 554,241 sequences was downloaded from Swiss-Prot [33]. Each protein sequence was represented as three sequences of 3-grams. 1,662,723 ($554,241 \times 3$) sequences were generated for training Protvec model, where each 3-amino acid phrase was converted to a 100 or 300-dimensional vector. For each protein-ligand pair, we explored three different combinations of ligand embedding and protein sequence embedding methods (**Figure S1**). The combination of 300-dimensional ligand representation and 300-dimensional protein representation showed slightly better performance than the other 2 combinations and was used for feature extraction of all datasets.

PCA and t-SNE algorithms for data exploration

To explore the data space, the high-dimensional embeddings mentioned above were first reduced to 50 dimensions using principal component analysis (PCA) [34], a traditional technique to orthogonal-linearly transform data to their low-dimensional representatives. The top 50 principal components were selected, with a cumulative explained variance percentage of 89.65% (**Figure S2**). We then converted the 50-dimensional representations to two-dimensional map by Barnes-Hut t-distributed Stochastic Embedding (t-SNE) algorithm [35], a popular algorithm that captures the non-linear relationship of data. The t-SNE algorithm considers that samples are Gaussian-distributed in high dimensional space while Student t-distributed in low dimensional space, it learns the transformed embeddings with minimum information loss, and minimizes the Kullback-Leibler divergence of sample distributions in high and low dimensions simultaneously [36]. PCA and t-SNE were both implemented by Python Scikit-learn [37].

Model building

Prediction model selection. Random forest (RF) and deep neural network (DNN) regressors are implemented in Scikit-learn. In order to select the most suitable combination of parameters, grid search

was performed. In the final RF model, the number of estimators was set at 500 and the maximum number of features was set at 200. In the final DNN model, three hidden layers are activated by rectified linear unit (ReLU), where each layer is comprised of 200 neurons. The output layer is activated by sigmoid function. ADAM is used for weight optimization.

Generative model. GAN and AAE were established in Tensorflow (version 1.3.0) [38]. Fully connected layers were adopted in all modules. ADAM with learning rate of 0.01 was used as the optimization algorithm during the training for both generator and discriminator. The epoch number was set at 30, and the batch size was set at 64. The details of tuning to obtain model stability will be discussed in the results section. The GAN was trained by the combination of generator, Discriminator 1 and Discriminator 2. The AAE was trained by the combination of auto-encoder, generator, Discriminator 1 and Discriminator 2.

Model evaluation

Here, we adopted a rigorous four-level (CV1 – CV4) validation for model evaluation (**Figure S3**). In CV1, the datasets were randomly partitioned into five equal subsets with the same ratio of negative samples to positive samples for 5-fold cross-validation. In CV2, the kinases included in the training set were removed from the test set in order to test the ability to predict new target. Similar to CV2, inhibitors in the training set were excluded from the test set in CV3 in order to test the ability to predict new drugs. CV4 tests unseen inhibitors and targets by leaving out kinases or inhibitors in the training set. For model validation in CV2 and CV3, instead of using 5-fold cross validation, we randomly divided data points into 80% of training data and 20% of test data, and the process was repeated 20 times for testing.

As a regression problem, prediction error was measured using root mean squared error (RMSE) and Pearson's correlation coefficient between experimental measurements and predicted values to assess model quality. Area under ROC (ROC-AUC), AP (Average Precision) or the area under PRC (Precision Recall Curve), sensitivity and specificity were calculated to characterize the capability of the model in distinguishing different classes of the samples.

Results And Discussion

This work is focused on augmentation of kinase-inhibitor matrixes by AAE model to improve the generalization of the regression model that fed with imbalanced dataset.

The overall pipeline of our strategy is shown in Fig. 1. By means of proteochemometrics (PCM), the PCM2vec features combining the inhibitor features and protein features were extracted and fed into the PCM-AAE models to expand the positive data space [39]. New data space was then ensembled to further attenuate the bias towards negative samples. Finally, the model for prediction of binding affinity trained by ensembled data space, yielding more accurate predictions. In this section, we will elaborate on how our strategy was implemented and was evaluated from various perspectives. It is worth noting that each

prediction model in our research was assessed with a rigorous four-level (CV1 – CV4) validation strategy [40] (Figure S3).

Baseline methods comparison

To assess how effective the strategy we proposed, we firstly trained the baseline predictor which was fed with imbalanced dataset (Non-balanced model/NB model) using three popular machine learning algorithms, including random forest (RF), XGBoost [41] and deep neural network (DNN). To speed up the convergence of DNN, each feature was standardized with zero mean and unit variance prior to feeding feature vectors. As expected, DNN models fed with standardized data (ST-DNN) achieved more rapid convergence after 30 iterations than those fed with non-standardized data (NST-DNN), with RMSE of 0.06 and 0.17 respectively. Moreover, the performance of ST-DNN is far more superior to NST-DNN in almost all metrics (Figure S4). Specifically, (1) in CV1 (known drug-known target pairs prediction), ST-DNN is evidently surpass the other methods in almost all metrics; (2) In CV2 (unknown target prediction) and CV3 (unknown drug prediction), RF and ST-DNN showed comparable performance that outperformed XGboost; (3) In CV4 (unknown drug-unknown target pairs prediction), all models demonstrated no significant difference in metrics of AUC, AP, correlation and specificity; (4) From CV1-CV4, the sensitivity of ST-DNN models on average was higher than that of other models, indicating the stronger capability of ST-DNN to capture positive samples. Overall, ST-DNN predictor was of highest quality among the models evaluated and was employed further.

Implementation and training of PCM-GAN and PCM-AAE

The key assumption in this work is that a predictive model with the positive sample space being well-portrayed will have an enhanced generalization ability. To verify this hypothesis, two popular generative models based on DNN architectures, Generative Adversarial Network (GAN) and Adversarial Auto-encoder (AAE), were introduced to generate novel PCM2vec representations, so that the positive sample space would enlarge and alienate from the distribution of the negative samples. The resultant models were termed as PCM-GAN and PCM-AAE respectively (Fig. 2).

Generally, the typical GAN trains generator (G) and discriminator (D) jointly until G generates fake data that match the distributions of real data and these fake data cannot be distinguished from real data by D [42]. In PCM-GAN setting, to generate rational positive samples, the discriminator D was split as two discriminators, Discriminator 1 (D1) and Discriminator 2 (D2). D1 was used to distinguish generated data between real positive representations while D2 was used to judge if generated data obey the distribution of negative representations (Fig. 2a). One generator was trained jointly with two discriminators, which enables the model to minimize the loss of D1 prediction and generated data while maximize the loss of D2 prediction and generated data.

Based on auto-encoder and GAN, AAE is jointly trained by minimizing the reconstruction error of auto-encoder and the adversarial loss for matching the aggregated posterior distribution of outputs from encoder to the stochastic prior distribution [43]. Compared to the typical AAE model, PCM-AAE

architecture employed two discriminators in the section of GAN to match the distribution of the latent layer vector of the auto-encoder model, and generated data by generator. As shown in Fig. 2b, although the positive samples and negative samples were encoded to the latent vector by the same auto-encoder, they matched different latent distributions. Therefore, discriminator D1 was trained to match latent vector encoded by positive data and generated data, while discriminator D2 was trained to distinguish latent vector encoded by negative data and generated data. In the end, a specific sampled distribution was first converted to positive latent vector by the generator, which was then fed into decoder to output new positive PCM2vec data.

It is noteworthy that the continuous labels corresponding to the generated samples were learned by the generative model, which means that in our models, the dimension of input training data is 601 [600-dimensional feature vector + 1-dimensional experimental measurement (binding affinity)], and the outputs of the generator of GAN or of the decoder of AAE is a 600-dimensional new feature vector data point + 1-dimensional label (**Figure S5**). Therefore, to generate possible valid samples, we defined three criteria for acceptance of the generator: (1) whether the model trained with only the training set (imbalanced data) was improved by addition of generated data; (2) whether the model (GAN/AAE) can converge; (3) whether the generator could generate valid distribution of labels, in this case valid distribution means at least 80% of predicted labels distributes between 5.3 and 11 (corresponding to activity of 0.01 nM to 5 mM).

As shown in **Figure S6**, we plotted errors of PCM_GAN models against iterations. When training PCM-GAN, we found that it is difficult to converge for generator and discriminator simultaneously, that is, one side was completely victorious while the other side continuously had large error at any arbitrary point, even with addition of dropout layer and batch-normalization layer. Interestingly, although the PCM-AAE model seems more complicated than PCM-GAN, PCM-AAE can achieve a win-win situation of discriminator and generator with converged loss during confrontation (Fig. 3a). A potential reason is that compared with 601-dimensional data, it is easier to capture the distribution of the low-dimensional data reduced by auto-encoder; or that auto-encoder is good at learning more effective representation in a low dimensional [44].

Previous research showed that batch-normalization were crucial to AAE training on different datasets [43]. In our PCM-AAE settings, the batch-normalization used in encoder layers of auto-encoder did help convergence of models, so as to generating possible valid samples to balance training dataset (Fig. 3a). At the end, 50 generators were accepted and utilized to generate new samples for reconstruction of DNN predictors. Here, to test the robustness and effectiveness of the generators, CV4 cross-validation was iterated 20 times for each generator to yield predictor. Overall, the models with data complemented by the generators showed more superior performance than the NB models (Fig. 3b).

Exploration of PCM-AAE data space

To illustrate the effectiveness of PCM-AAE for data augmentation, we applied the PCA and t-SNE algorithms to explore data space, which mapped PCM2vec to low-dimensional space for data visualization.

Firstly, to explore the data space of NB models, we split the entire data into a set of CV4 training data and test data, and analyze its distribution. As shown in **Figure S7**, the data space of test and training set had little overlap for both negative data and positive data, explaining that NB models showed significant difference of prediction accuracy between training set and test set at CV4 level is because data space of training set cannot cover the test data space, so that NB models cannot fit test data well. Such overfitting can also occur more or less from CV1 to CV3, which cannot be reduced by tuning the parameters of models (**Table S1**).

Next, to investigate the reason why PCM-AAE could reduce overfitting of NB models, we further explored data space of PCM-AAE. Specifically, to reconstruct DNN predictor, each of 50 accepted generators derived from PCM-AAE model generated new positive samples until the training set is balanced. On that account, the test set was predicted by 50 DNN predictors respectively. As expected, the performance of the predictors was enhanced to various degrees compared to the NB model, possibly ascribed to diverse space expanded by different generators. The best generator was selected and the corresponding merged data were fed into PCA and t-SNE for further dimensionality reduction and visualization (Fig. 4). From the resultant maps, we can draw the following conclusions: (1) the newly generated data have less overlap with training data, demonstrating the unpopulated space was expanded by PCM-AAE; (2) data space of PCM-AAE generated data overlaps largely with positive samples of test data, and thus improved predictive power of the predictor to test set; (3) In general, expansion of the data space resulted in more separate distribution of the positive and negative samples. Encouragingly, the conclusions were consistent when the training set and the test set were randomly regenerated and subjected to the analysis above. The results partially indicated the intuitive idea of PCM-AAE, that is, for imbalanced dataset, the generalization ability of prediction model can be enhanced with expanded data space.

Training and Evaluation of EPA

Although PCM-AAE showed favorable performance, as described above, we found single generator was not enough to complete data space in every case. To address this, an ensemble strategy, ensemble of PCM-AAE (EPA), was then proposed to enable the different generated space to be complemented by each other and thus give more accurate and robust prediction. Training regime of EPA and ENB (ensemble of NB models) is shown in Fig. 5a. Firstly, we split datasets as training set (N data points) and test set at CV1-CV4 levels respectively. After that, for each accepted generator, the training set was complemented until it balanced, then the balanced set was used to train DNN predictor to re-predict data points of the original training set. The 50 predictions was yielded by generators in 50 re-trained DNN models for the original training set ($N \times 50$ dimensions), which was then fed into RF model to obtain final predictions. To select the best RF model, five RF models were trained and evaluated on corresponding test sets, since the set was split in the manner of five-fold cross validation. The procedure mentioned above was repeated 20 times for robustness. To be fair, the best ENB model (ensemble of non-balanced models) was also trained, evaluated and selected in the same manner as EPA with imbalanced training datasets.

Encouragingly, as shown in Fig. 5b, EPA consistently outperformed ENB from CV1 to CV4 levels in almost all metrics considered. Especially in CV4, compared to ENB, EPA reached an average improvement in AUC

of 0.11, AP of 0.10, correlation of 0.15 and sensitivity of 0.19, an average decline in RMSE of 0.22, as well as a slight drop in specificity of 0.04 on average. Especially, the sensitivity was enhanced notably with minimum sacrifice of specificity, indicating that more positive samples were recalled in EPA without a declined ability to capture negative samples. The improvement from ENB to EPA demonstrates better fits can be obtained through augmentation of positive data space. It is worth mention that the performance of EPA enhanced the most at CV4 level, a potential explanation is that since PCM2vec is the combination of Mol2vec and ProtVec, the assumption that expanding data space improves generalization of predictor becomes relative invalid at CV1-CV3 levels, where the feature space of test data was already included in the training data space.

Table 2
Summary of ENB and EPA performance on external datasets.

Dataset	Method	ROC_AUC	PRC_AP	Pearson correlation coefficient (r^2)
Metz	ENB	0.81	0.45	0.49
	EPA	0.84	0.49	0.54
Davis	ENB	0.68	0.76	0.39
	EPA	0.70	0.78	0.43
PKIS1	ENB	0.75	0.17	0.28
	EPA	0.79	0.23	0.34
PKIS2	ENB	0.68	0.70	0.36
	EPA	0.74	0.75	0.45
KinaseSafari	ENB	0.69	0.71	0.34
	EPA	0.74	0.76	0.44
MRC	ENB	0.69	0.42	0.38
	EPA	0.74	0.46	0.45

To further examine the superiority of EPA model over ENB, assessments were conducted on six external independent datasets, and the performance such as ROC_AUC, PRC_AP and correlation of EPA was more favorable than ENB (Table 2). It should be noticed that the data points presented in both the previous training set and external sets were excluded when constructing validation datasets. Generally, better performance goes with higher ROC_AUC and higher PRC_AP. However, in terms of our models, the trends of ROC_AUC and PRC_AP were not consistent among different external datasets. For example, ROC_AUC of Metz’s dataset ranked the first among the 6 datasets, while its PRC_AP was relatively low. One possible explanation is data inconsistency among datasets that derived from different biochemical assays, another explanation is different extent of dataset balance, because PRC_AP is more sensitive to imbalanced datasets than ROC_AUC. To verify these two points, the Pearson correlation coefficient

between the experimental measurements from every two datasets was calculated with the data points overlapping with the training data (**Figure S8a**), and positive and negative samples in test set were counted. As expected, the data more relevant with the training set yields better predictions, and balanced data enabled higher PRC_{AP} (**Figure S8b**).

Applicability of EPA on kinase profile prediction and virtual screening

Detecting off-target effects or the kinase profiling of kinase-specific inhibitors is vital for portraying attribution of drugs. To quantify the model's capability to predict off-target effects of compounds, the selectivity score was defined as the ratio of the number of hits to that of all tested kinases [45]. Intuitively, for a given inhibitor, lower selectivity score indicates better selectivity. Hence, we re-predicted the training set by EPA, generating predicted selectivity of each compound for correlation analysis with the compound selectivity measured by experiments. Another two sets, Davis's and Metz's, with relative complete kinase panels for different compounds, were also analyzed in the same way. As presented in Fig. 6a, experimental selectivity scores from 3 datasets correlate closely with corresponding predictions by the EPA model, with Spearman correlation coefficient of 0.96, 0.89 and 0.74 respectively, which again outperformed ENB (**Figure S9**). To further validate whether EPA could accurately predict subfamily selectivity, Li's dataset that contains the kinase profile of five inhibitors absent from the datasets we employed so far was analyzed. Meanwhile, OR (odds ratio) described previously was also introduced to quantify subfamily selectivity [46]. The major targeted subfamilies of the compounds predicted by EPA accord well with experimental results (Fig. 6b).

Another important approach to drug discovery is structure-based virtual screening. Given desirable performance of EPA in CV3, we examined its effectiveness to recall active compounds for a given kinase and the potential of becoming an efficient alternative to conventional structure-based virtual screening methods. Firstly, we analyzed the training set grouped by targets. In each subset, PRC_{AP} was adopted to evaluate the classification accuracy with cut-off of 1 μ M, while RMSE was used to assess the predictive power of the regressor. Theoretically, EPA can address the machine learning problem even when training data were insufficient. Thus, we next analyzed the relationship between model performance and data composition including number of data points and degree of data imbalance. As presented in Fig. 7, on the same training set, the number of well-fitted kinases by EPA was far more than that of ENB, especially for kinases with few or imbalanced data points.

Next, we delved into the kinases in Metz's dataset to further investigate the capability of ENB and EPA to screen active compounds (Fig. 8a and S10). The set was divided into two subsets, where one subset included the kinases presented in the training set (presented as circles) and the kinases in the other subset were absent from training set (presented as squares). Although both ENB and EPA generated considerable predictions to most kinases that had been seen by the model, EPA topped ENB on most kinases that previously unseen by the models, further suggesting the effectiveness of our strategy. Encouraged by these results, we evaluated EPA with 26 kinases in the Directory of Useful Decoys-

Enhanced (DUD-E) database [47] to see if EPA discriminates true binders from decoy molecules. As shown in Fig. 8b, EPA achieved superior predictions over ENB again with an average AUC of 0.77, which is quite high for a non-docking based predictive method so far to the best of our knowledge. In addition, the performance of EPA was comparable with classical docking algorithms including GOLD, Glide, SurFlex, and FlexX in terms of BEDROC scores at alpha of 80.5 based on the benchmark work of Chaput *et al* [48] (Fig. 8c).

Conclusion

In this study, EPA were devised to improve predictive power of kinase-inhibitor interactions with an enlarged feature space. In baseline model comparison, we found that DNN model with standardized PCM2vec embeddings achieved supreme performance, even surpassing the well-performed RF models, especially in the metric of sensitivity which representing good recall for positive points. Based on this DNN predictor, we developed two semi-supervised models termed PCM-GAN and PCM-AAE to improve the generalization capability, which expands the feature space by generative models to separate the distribution of positive and negative samples. As expected, both PCM-GAN and PCM-AAE models outperformed classical non-balanced PCM models at rigorous fourth cross validation level. To obtain more complete data space and thus more reliable predictions, multiple PCM-AAE models were integrated as EPA model, which outperformed the other imbalanced approaches in internal validation and exhibited the most robust performance. Further test of EPA in external datasets and its practicality in kinase selectivity profiling and virtual screening also verified superior performance of EPA to predict interaction of kinase-inhibitors.

To further demonstrate the practicability of the strategy, EPA could be applied to the discovery of novel kinase-inhibitors and solve more challenging prediction problems with few data points. Encouragingly, our strategy is a proof-of-concept implementation of generative model in feature space exploration, which is expected to be applicable in various drug discovery scenarios where abundant training data are not available.

Abbreviations

DPI

drug-protein interaction

DL

deep learning

PCM

proteochemometrics

RF

random forest

DNN

deep neural network

ECFP

extended-connectivity circular fingerprints

POC

percent of control

PCA

principal component analysis

t-SNE

t-distributed Stochastic Embedding

ReLU

rectified linear unit

GAN

generative adversarial network

AAE

adversarial auto-encoder

PDB

protein data bank

AUC

area under the ROC curve

ROC

receiver operating characteristic

PRC

precision recall curve

AP

average precision

RMSE

root mean squared error

Declarations

Availability of data and materials

The code for EPA is available at <http://github.com/xybai-dev/EPA>. Dataset of Kinase SARfari was downloaded from <http://ftp.ebi.ac.uk/pub/databases/chembl/>. MRC dataset was downloaded from <https://www.ppu.mrc.ac.uk/>. The other data used in this paper are supported by the corresponding reference.

Competing interests

The authors declare no competing interests.

Funding

This work was supported by grants to Y. Yin, including the National Key Research and Development Program of China (2016YFA0500302); the National Natural Science Foundation of China (81430056, 31420103905, 81621063); The Beijing Natural Science Foundation (7161007); and the Lam Chung Nin Foundation for Systems Biomedicine.

Authors' contributions

Bai Xinyu designed and performed the research, Yin Yuxin supervised the study. Both authors read and approved the final manuscript.

Acknowledgements

Not applicable.

Authors' information (optional)

Department of Pathology, School of Basic Medical Sciences, Peking University Health Science Center, Beijing 100191, China.

Yu-Xin Yin & Xin-Yu Bai

Institute of Systems Biomedicine, School of Basic Medical Sciences, Peking University Health Science Center, Beijing 100191, China.

Yu-Xin Yin

Peking-Tsinghua Center for Life Sciences, Peking University Health Science Center, Beijing 100191, China.

Yu-Xin Yin

References

1. M Keiser, V Setola, J Irwin et al (2009) Predicting new molecular targets for known drugs. *Nature* 462:175-81.
2. JW Scannell and J Bosley (2016) When quality beats quantity: decision theory, drug discovery, and the reproducibility crisis. *PLoS One* 11:e0147215.
3. G Zhang, J Xing, Y Wang et al (2018) Discovery of novel inhibitors of indoleamine 2, 3-dioxygenase 1 through structure-based virtual screening. *Frontiers in pharmacology* 9:277.
4. L Dong, S Shen, W Chen et al (2019) Discovery of Novel Inhibitors Targeting Human O-GlcNAcase: Docking-Based Virtual Screening, Biological Evaluation, Structural Modification, and Molecular Dynamics Simulation. *Journal of chemical information and modeling* 59:4374-4382.
5. F Kilchmann, MJ Marcaida, S Kotak et al (2016) Discovery of a selective aurora a kinase inhibitor by virtual screening. *Journal of medicinal chemistry* 59:7188-7211.

6. T Scior, A Bender, G Tresadern et al (2012) Recognizing pitfalls in virtual screening: a critical review. *Journal of chemical information and modeling* 52:867-881.
7. Y Li, L Han, Z Liu et al (2014) Comparative assessment of scoring functions on an updated benchmark: 2. Evaluation methods and general results. *Journal of chemical information and modeling* 54:1717-1736.
8. Y LeCun, Y Bengio and G Hinton (2015) Deep learning. *nature* 521:436-444.
9. H Chen, O Engkvist, Y Wang et al (2018) The rise of deep learning in drug discovery. *Drug discovery today* 23:1241-1250.
10. E Gawehn, JA Hiss and G Schneider (2016) Deep learning in drug discovery. *Molecular informatics* 35:3-14.
11. MH Segler, T Kogej, C Tyrchan et al (2018) Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS central science* 4:120-131.
12. T Blaschke, M Olivecrona, O Engkvist et al (2018) Application of generative autoencoder in de novo molecular design. *Molecular informatics* 37:1700123.
13. X Liu, K Ye, HW van Vlijmen et al (2019) An exploration strategy improves the diversity of de novo ligands using deep reinforcement learning: a case for the adenosine A_{2A} receptor. *Journal of Cheminformatics* 11:35.
14. D Rogers and M Hahn (2010) Extended-connectivity fingerprints. *Journal of chemical information and modeling* 50:742-754.
15. S Jaeger, S Fulle and S Turk (2018) Mol2vec: unsupervised machine learning approach with chemical intuition. *Journal of chemical information and modeling* 58:27-35.
16. DK Duvenaud, D Maclaurin, J Iparraguirre et al (2015) Convolutional networks on graphs for learning molecular fingerprints. pp 2224-2232.
17. A Kadurin, S Nikolenko, K Khrabrov et al (2017) druGAN: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico. *Molecular pharmaceutics* 14:3098-3104.
18. T Unterthiner, A Mayr, G Klambauer et al (2014) Deep learning as an opportunity in virtual screening. pp 1-9.
19. I Wallach, M Dzamba and A Heifets (2015) AtomNet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *arXiv preprint arXiv:1510.02855*
20. C Kramer, T Kalliokoski, P Gedeck et al (2012) The experimental uncertainty of heterogeneous public K_i data. *Journal of medicinal chemistry* 55:5165-5173.
21. GV Paolini, RH Shapland, WP van Hoorn et al (2006) Global mapping of pharmacological space. *Nature biotechnology* 24:805-815.
22. X Zhu, Y Liu, Z Qin et al (2017) Data augmentation in emotion classification using generative adversarial networks. *arXiv preprint arXiv:1711.00648*

23. L Sixt, B Wild and T Landgraf (2018) Rendergan: Generating realistic labeled data. *Frontiers in Robotics and AI* 5:66.
24. SK Lim, Y Loo, N-T Tran et al (2018) Doping: Generative data augmentation for unsupervised anomaly detection with gan. pp 1122-1127.
25. S Christmann-Franck, GJ van Westen, G Papadatos et al (2016) Unprecedentedly Large-Scale Kinase Inhibitor Set Enabling the Accurate Prediction of Compound–Kinase Activities: A Way toward Selective Promiscuity by Design? *Journal of chemical information and modeling* 56:1654-1675.
26. A Gaulton, LJ Bellis, AP Bento et al (2012) ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic acids research* 40:D1100-D1107.
27. D H Drewry, T M Willson and W J Zuercher (2014) Seeding collaborations to advance kinase science with the GSK Published Kinase Inhibitor Set (PKIS). *Current topics in medicinal chemistry* 14:340-342.
28. DH Drewry, CI Wells, DM Andrews et al (2017) Progress towards a public chemogenomic set for protein kinases and a call for contributions. *PloS one* 12:e0181585.
29. JT Metz, EF Johnson, NB Soni et al (2011) Navigating the kinome. *Nature chemical biology* 7:200-202.
30. MI Davis, JP Hunt, S Herrgard et al (2011) Comprehensive analysis of kinase inhibitor selectivity. *Nature biotechnology* 29:1046-1051.
31. E Asgari and MR Mofrad (2015) Continuous distributed representation of biological sequences for deep proteomics and genomics. *PloS one* 10:e0141287.
32. T Sterling and JJ Irwin (2015) ZINC 15–ligand discovery for everyone. *Journal of chemical information and modeling* 55:2324-2337.
33. E Boutet, D Lieberherr, M Tognolli et al, *Uniprotkb/swiss-prot*, in *Plant bioinformatics*. 2007, Springer. p. 89-112.
34. S Wold, K Esbensen and P Geladi (1987) Principal component analysis. *Chemometrics and intelligent laboratory systems* 2:37-52.
35. GC Linderman, M Rachh, JG Hoskins et al (2017) Efficient algorithms for t-distributed stochastic neighborhood embedding. *arXiv preprint arXiv:1712.09005*
36. H Zhou, F Wang and P Tao (2018) t-Distributed Stochastic Neighbor Embedding Method with the Least Information Loss for Macromolecular Simulations. *Journal of chemical theory and computation* 14:5499-5510.
37. F Pedregosa, G Varoquaux, A Gramfort et al (2011) Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12:2825-2830.
38. M Abadi, P Barham, J Chen et al (2016) Tensorflow: A system for large-scale machine learning. pp 265-283.
39. GJ van Westen, JK Wegner, AP IJzerman et al (2011) Proteochemometric modeling as a tool to design selective compounds and for extrapolating to novel targets. *MedChemComm* 2:16-30.

40. FA Sorgenfrei, S Fulle and B Merget (2018) Kinome-wide profiling prediction of small molecules. *ChemMedChem* 13:495-499.
41. T Chen and C Guestrin (2016) Xgboost: A scalable tree boosting system. pp 785-794.
42. I Goodfellow, J Pouget-Abadie, M Mirza et al (2014) Generative adversarial nets. pp 2672-2680.
43. A Makhzani, J Shlens, N Jaitly et al (2015) Adversarial autoencoders. arXiv preprint arXiv:1511.05644
44. GE Hinton and RR Salakhutdinov (2006) Reducing the dimensionality of data with neural networks. *science* 313:504-507.
45. MW Karaman, S Herrgard, DK Treiber et al (2008) A quantitative analysis of kinase inhibitor selectivity. *Nature biotechnology* 26:127-132.
46. X Li, Z Li, X Wu et al (2019) Deep learning enhancing kinome-wide polypharmacology profiling: model construction and experiment validation. *Journal of Medicinal Chemistry*
47. N Huang, BK Shoichet and JJ Irwin (2006) Benchmarking sets for molecular docking. *Journal of medicinal chemistry* 49:6789-6801.
48. L Chaput, J Martinez-Sanz, N Saettel et al (2016) Benchmark of four popular virtual screening programs: construction of the active/decoy dataset remains a major determinant of measured performance. *Journal of cheminformatics* 8:1-17.

Figures

Figure 1

Overall pipeline. The protein sequence and molecules were transformed into a set of features based on Mol2vec and Protvec model respectively. The PCM2vec features was fed into PCM-AAE to generate new PCM2vec features, complementing the PCM space for training predictors. Generated predictors were integrated as a new model to predict binding affinity.

Figure 2

The architecture of (a) PCM-GAN and (b) PCM-AAE.

Figure 3

(a) Loss plot of PCM-AAE. (b) Performance comparison of sensitivity among NB model (Non-Balanced model) and 50 reconstructed models fed with data augmented by 50 generators.

Figure 4

Visualization by t-SNE of the distributions of datasets, including raw datasets and new data points generated by PCM-AAE. The generated positive data space (green) not only separated clearly from negative data space in training set (blue) or test set (orange), but also expanded positive data space of the training set (red), and has similar distribution with positive data space of the test set (purple).

Figure 5

(a) Training process of EPA and (b) the performance in different metrics compared with ENB (ensemble of non-balanced models). Paired Wilcoxon signed-ranked test was used to determine whether the performance between EPA and ENB was statistically significant. ns: $p > 0.05$; *: $p < 0.05$; **: $p < 0.01$; ***: $p < 0.001$; ****: $p < 0.0001$.

Figure 6

(a) Scatterplots of EPA-predicted selectivity score and experimental selectivity score of inhibitors in various sets. (b) Comparison of experimental odd ratio (OR) and EPA-predicted OR for five drugs.

Figure 7

The ratio of positive samples to negative samples for each kinase plotted against total number of data points in training set, based on the predictions by ENB and EPA.

Figure 8

(a) Phylogenetic tree to display the performance of EPA on kinases from the Metz's set. Circles represent the kinases included in training set. Squares represent the kinases absent from training set. (b) ROC analysis of EPA and ENB predictions to DUD-E dataset. (c) Distribution of BEDROC ($\alpha=80.5$) covering 26 kinase targets in DUD-E for EPA and docking programs.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [Supplementary.docx](#)