Bioinformatics, 38, 2022, ii75–ii81
https://doi.org/10.1093/bioinformatics/btac496
Proceedings of ECCB2022

OXFORD

Insights into performance evaluation of compound–protein interaction prediction methods

Adiba Yaseen[1,*], Imran Amin[2], Naeem Akhter[1], Asa Ben-Hur[3] and Fayyaz Minhas [4,*]

[1]Department of Computer and Information Sciences (DCIS), Pakistan Institute of Engineering and Applied Sciences (PIEAS), Islamabad 45650, Pakistan, [2]National Institute for Biotechnology and Genetic Engineering, Faisalabad 38000, Pakistan, [3]Department of Computer Science, Colorado State University, Fort Collins, CO 80523, USA and [4]Department of Computer Science, University of Warwick, Coventry CV4 7AL, UK

*To whom correspondence should be addressed.

Abstract

Motivation: Machine-learning-based prediction of compound–protein interactions (CPIs) is important for drug design, screening and repurposing. Despite numerous recent publication with increasing methodological sophistication claiming consistent improvements in predictive accuracy, we have observed a number of fundamental issues in experiment design that produce overoptimistic estimates of model performance.

Results: We systematically analyze the impact of several factors affecting generalization performance of CPI predictors that are overlooked in existing work: (i) similarity between training and test examples in cross-validation; (ii) synthesizing negative examples in absence of experimentally verified negative examples and (iii) alignment of evaluation protocol and performance metrics with real-world use of CPI predictors in screening large compound libraries. Using both state-of-the-art approaches by other researchers as well as a simple kernel-based baseline, we have found that effective assessment of generalization performance of CPI predictors requires careful control over similarity between training and test examples. We show that, under stringent performance assessment protocols, a simple kernel-based approach can exceed the predictive performance of existing state-of-the-art methods. We also show that random pairing for generating synthetic negative examples for training and performance evaluation results in models with better generalization in comparison to more sophisticated strategies used in existing studies. Our analyses indicate that using proposed experiment design strategies can offer significant improvements for CPI prediction leading to effective target compound screening for drug repurposing and discovery of putative chemical ligands of SARS-CoV-2-Spike and Human-ACE2 proteins.

Availability and implementation: Code and supplementary material available at https://github.com/adibayaseen/HKRCPI.

Contact: Fayyaz.Minhas@warwick.ac.uk or adibayaseen@gmail.com

1 Introduction

Compound–protein interaction (CPI) prediction is an important task in target compound screening for identifying protein targets of compounds, drug design and drug repurposing studies (Schirle and Jenkins, 2016). Affinity chromatography (Broach and Thorner, 1996) and protein microarrays (Lee and Lee, 2016; Zhao *et al.*, 2021) are among the most frequently used experimental methods for the identification of CPIs. However, such wet-lab approaches can be expensive and time-consuming (Zhang *et al.*, 2017). The emergence of pandemics such as Ebola and COVID-19 and the global challenge of antimicrobial resistance have highlighted the need of improving efficiency and throughput in drug design (Thafar *et al.*, 2019). Consequently, CPI prediction using computational methods has become an attractive area of research (Chen *et al.*, 2016) as such approaches can improve the cost, time and efficiency of drug discovery in contrast to experimental methods (Mazandu *et al.*, 2018).

1.1 Methods for CPI prediction

Conventionally, structure-based and ligand-based virtual screening is the most well-researched areas of drug discovery (Lim *et al.*, 2021) but such techniques require tertiary structure of the protein of interest. As a consequence, machine learning (ML)-based methods that use sequence characteristics of proteins and chemical structural representations of compounds for interaction prediction have been developed (Bleakley and Yamanishi, 2009; Bredel and Jacoby, 2004). Classical ML approaches in this domain range from similarity-based methods (Chen *et al.*, 2018) to feature representation and kernel-based approaches (Bleakley and Yamanishi, 2009; Gönen, 2012); pairwise kernels (Jacob and Vert, 2008), etc. Comparative analysis by Ding *et al.* (2014) has shown that pairwise kernels outperform other approaches. In recent years, researchers have developed multiple deep learning models for CPI prediction. DeepDTA (Öztürk *et al.*, 2018) extracts real-valued sparse feature representations of proteins as well as compounds using convolutional neural networks (CNNs) and appends these features through the final fully connected layer. WideDTA (Öztürk *et al.*, 2019) and Conv-DTI (Wang *et al.*, 2020) also used an analogous idea with additional features, ligand structural similarity and information about protein domains and motifs to enhance model accuracy. For the representation of compound structures, CPI–NN (Tsubaki *et al.*, 2019) and Graph-DTA (Nguyen *et al.*, 2021) used novel graph neural networks (GNNs) (Zhang *et al.*, 2021) as an alternative to CNNs resulting in state-of-the-art prediction accuracy.

1.2 Issues in performance assessment of CPI models

Despite increasing sophistication of CPI models through deep learning, the generalization performance of existing approaches on independent or real-world datasets is still not perfect (Riley, 2019). One of the fundamental issues behind this is biased and overly optimistic performance assessment strategies arising from the use of unsuitable datasets, poor non-redundancy control in train-test data splitting in cross-validation (CV), improper procedures for generation of negative example, lack of independent test sets and choice of performance metrics. Here, we discuss each of these issues in further detail.

A number of ML-based CPI prediction models have used the MUV (Rohrer and Baumann, 2009), DUD-E (Mysinger *et al.*, 2012) and Human-CPI datasets (Liu *et al.*, 2015; Tsubaki *et al.*, 2019) for model training and performance evaluation. However, most datasets in this domain do not contain true or experimentally verified negative examples and may have a large degree of redundancy between proteins and compounds which can lead to biased ML models (Chen *et al.*, 2019, 2020; Sieg *et al.*, 2019).

Another issue associated with the performance assessment of ML CPI models is the protocol used for generating negative examples. As there are no standardized datasets of negative examples for CPI prediction, researchers in this domain resort to one of two approaches for the generation of 'synthetic' negative examples for training and performance assessment: *Random pairing* and *Inter-class similarity-controlled negative example generation*. In random pairing, proteins and compounds in the positive set are simply randomly paired for generating synthetic negative examples after exclusion of known positive pairs as in the dataset used in CPI-NN (Tsubaki *et al.*, 2019). However, researchers have argued that random-pairing can produce examples that are highly similar to positive examples and this can add labeling noise in training (Ding *et al.*, 2014). As a consequence, they have proposed that negative examples should be generated with controls over inter-class similarity. This process first creates a candidate negative set through random pairing of compounds and proteins. Then a similarity function is used to calculate the degree of similarity between a candidate negative example and the given set of positive examples. Only those candidate negative examples are added to the final negative set whose similarity score with positive examples is lower than a pre-specified threshold resulting in negative examples that are sufficiently dissimilar to known positive examples (Ding *et al.*, 2014). However, as in the case of protein–protein interaction prediction models (Ben-Hur and Noble, 2006), the use of similarity controlled negative example generation in model evaluation can result in overly optimistic performance results with a high likelihood of generalization failure on real-world test sets.

Many existing approaches also use an equal number of positive and negative examples even though the number of compounds that can be expected to bind to a given protein can be significantly smaller in comparison to the size of the universe of possible compounds. This results in the generation of a large number of false positives in real-world applications. Furthermore, CV protocols employed in most existing ML CPI models also do not consider protein sequence and compound similarity in generating training and test folds resulting in overly optimistic performance estimates as the training set can contain examples that are very similar to test examples. Ideally, the examples in the test folds should be sufficiently different from training examples to reflect real-world use cases.

Lastly, existing methods report areas under the receiver operating characteristic or precision–recall curves (Area under the Receiver Operating Characteristic Curve (AUCROC)/Area under the Precision-Recall Curve (AUC-PR)) as performance metrics. However, given that such approaches are typically used for screening interactions from a large number of candidate compound–protein pairs for wet-lab validation, these metrics do not provide a directly interpretable estimate of how good a method is at ranking interacting compounds of a protein.

1.3 Contributions of this work

In this work, we highlight the issues discussed above with a number of experiments using existing state-of-the-art CPI prediction model (CPI-NN) (Tsubaki *et al.*, 2019) and Graph-DTA (Nguyen *et al.*, 2021) as well as a simple heterogeneous kernel-based approach. We suggest improvements in the evaluation protocol used for performance assessment of such models in terms of negative example generation as well as performance metrics. We report the prediction results of the proposed approach for screening candidate compounds for a number of test proteins not included in the datasets used in model construction including SARS-CoV-2 Spike and Human-ACE2 proteins.

2 Materials and methods

In this section, we discuss details of our datasets, experiments and ML methods for CPI prediction.

2.1 Datasets

2.1.1 Non-redundant Liu et al. human CPI dataset

We use the human protein–compound interaction dataset originally proposed by Liu *et al.* (2015) and employed in a number of existing methods such as CPI-NN (Tsubaki *et al.*, 2019). In this dataset, positive examples consisting of protein–compound pairs were collected from two experimentally verified databases: DrugBank 4.1 (Wishart *et al.*, 2008) and Matador (Günther *et al.*, 2008). This dataset has 3364 positive examples of interacting protein–compound pairs constituting 852 unique proteins and 1179 unique compounds. It also contains an equal number of negative examples obtained by randomly pairing proteins and compounds in the positive set provided as part of the CPI-NN code repository (Tsubaki *et al.*, 2019). We found that the aforementioned dataset by Liu *et al.* used in CPI-NN (Tsubaki *et al.*, 2019) contained duplicated examples. We removed these duplicated examples from the positive set resulting in 2633 unique positive examples that constitute our non-redundant Liu *et al.* human CPI (NR-HCPI) dataset together with negative examples obtained by randomly pairing proteins and compounds in the positive set excluding any pairs already included in the positive set. We generated different ratios of positive-to-negative examples (P:N = 1:1, 1:3, 1:5 and 1:7) for the evaluation of predictive performance under more realistic evaluation scenarios with high-class imbalance. In conjunction with this dataset, we also utilized a non-redundant CV (NRCV) protocol which is detailed in the performance evaluation section.

2.1.2 Binding DB dataset

As discussed in Section 1, one of the fundamental issues with protein–compound interaction datasets is the lack of experimentally verified negative examples. For performance assessment of CPI prediction methods and for studying the impact of various approaches for generating synthetic negative examples, we have used the binding affinity values of protein–compound pairs in the latest version (June 2021) of Binding DB (Gilson *et al.*, 2016) with a total of 22 782 226 examples. For this purpose, we applied a number of data filtering steps (detailed in GitHub Supplementary Material) such as using only single-chain protein targets with experimentally verified inhibition constant values that are sufficiently high to ensure very low probability of interaction to select our final dataset of 3657 negative examples.

2.1.3 Superdrug bank for drug repurposing

For drug repurposing analysis, we use the SuperDRUG2 (version 2) database (Siramshetty *et al.* 2018) of approved and commercially available drugs with a total of 3633 unique small molecules. We have also used the Superdrug bank molecules for screening potential targets of SARS-Cov-2 Spike protein and the human-ACE2 protein.

2.2 ML models

For performance analysis, we have used the available implementations of Graph-DTA (Nguyen *et al.*, 2021) and CPI-NN methods which give state-of-the-art results (Tsubaki *et al.*, 2019). CPI-NN has been validated for human and *Caenorhabditis elegans* proteins with high AUC–ROC (0.95) and under different class ratio settings over the same datasets. Similarly, Graph-DTA is a state-of-the-art approach for predicting binding affinity of drugs and proteins which can be used for CPI prediction. We have used the publicly available codes of CPI-NN and Graph-DTA for conducting experiments with various CV and assessment strategies after verifying the reproducibility of the results using the experimental settings as reported in the original papers.

As a baseline, we have also developed a simple kernel-based approach for CPI prediction (see Fig. 1). For this purpose, we model CPI prediction as a classification problem in which every example $x \equiv (c, p)$ consists of a protein p and a compound c with corresponding feature representations $\psi(p)$ and $\phi(c)$, respectively. Each example in the training dataset $D = \{((p_i, c_i), y_i) | i = 1 \dots N\}$ is associated with a binary label $y_i \in \{-1, \ +1\}$ indicating whether the corresponding protein and compound interact $(+1)$ or not (-1).

2.2.1 Protein and compound features

In order to capture amino-acid-specific binding characteristics of proteins with their target compounds in the predictive model, we have used the amino acid composition (AAC) of protein (denoted by $\psi_{\mathrm{AAC}}(p)$) which is a 20-dimensional vector representation of a protein sequence containing the frequency of occurrence of various amino acids in the protein sequence. For modeling the physiochemical similarity across amino acids, we used grouped k-mer composition of proteins as a feature vector. In this approach, each amino acid in a protein is assigned one of seven predetermined groups based on its physicochemical characteristics (Hashemifar *et al.*, 2018) and the counts of all possible group-level k-mers are used as a feature vector. For $k = 2$ and $k = 3$, this results in $7^2 = 49$- and $7^3 = 343$-dimensional features of a protein denoted by $\psi_2(p)$ and as $\psi_3(p)$, respectively.

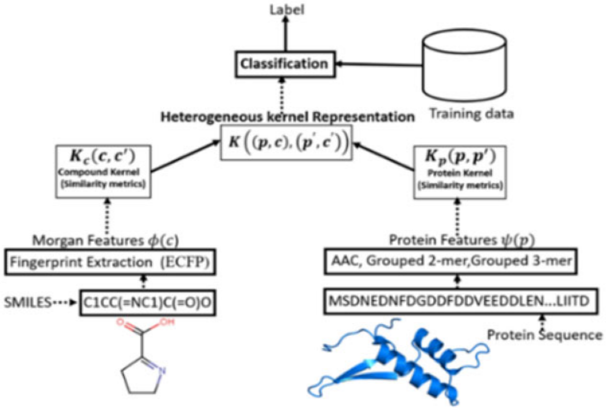

Fig. 1. Concept diagram of kernel CPI (kernel-CPI) prediction. Protein sequence and SMILES are given as input, AAC and grouped k-mer ($k=2$, $k=3$) features are extracted from protein sequence and concatenated into a single feature vector $\psi(p)$ for computing a protein-level kernel $K_p(p, p')$. The Morgan fingerprint $\phi(c)$ is extracted from SMILES representation of a compound to calculate kernel $K_c(c, c')$. These kernels are combined into a kernel representation $K\left((p, c), (p', c')\right)$ for CPI prediction

For modeling compound characteristics, we extract features from SMILES of compounds in the CPI pair using its extended-connectivity fingerprint (ECFP) (also known as Morgan fingerprint) (Veselinovic *et al.*, 2015) using RDKit (Cao *et al.*, 2013). This fingerprint is a topological feature of a chemical compound and captures its structural confirmation within a given radius. The feature dimension of this representation is 1024 for a radius of three atoms.

2.2.2 Heterogeneous kernel representation

We have developed a simple kernel method for CPI prediction. As each classification example in this problem comprises a protein and compound, we first construct non-linear kernel representations of proteins and compounds separately which are then merged to form a heterogeneous feature space kernel for classification as shown in Figure 1.

2.2.2.1 Compound similarity kernel. We use the compound feature representation $\phi(c)$ to construct a radial basis function (RBF) similarity kernel between pairs of compounds as follows: $K_c(c_i, c_j) = \exp(-\gamma_c \|\phi(c_i) - \phi(c_j)\|^2)$. In this equation, $\phi(c_i)$ and $\phi(c_j)$ are Morgan fingerprint feature vectors as described in the previous section. The kernel $K_c(c_i, c_j)$ essentially measures the degree of similarity of two compounds in the feature space in a non-linear manner with a single hyper-parameter $\gamma_c > 0$.

2.2.2.2 Protein similarity kernel. For a protein p, all three feature vectors of protein sequence p are concatenated in a feature representation $\psi(p)$ resulting in a 412-dimensional column vector of the protein features as $\psi(p) = \left[\ \psi_{\mathrm{AAC}}(p) \mid \psi_2(p) \mid \psi_3(p)\right]$.

This feature representation is then used to generate a protein–protein similarity kernel as follows: $K_p(p_i, p_j) \ = \exp(-\gamma_p \ \|\psi(p_i) - \psi(p_j)\|^2)$.

2.2.2.3 Heterogeneous kernel representation and classification. Based on protein and compound similarity kernels, we construct a heterogeneous feature space kernel between pairs of examples (p, c) and (p', c') each consisting of a protein and a compound as follows.

$$\begin{aligned} K((p,c),(p',c')) &= \langle \varphi(p,c), \varphi(p',c') \rangle = (K_p(p,p') + K_c(c,c'))^2 \\ &= K_p(p,p')^2 + K_c(c,c')^2 + 2K_p(p,p')K_c(c,c'). \end{aligned}$$

This joint kernel essentially measures the degree of similarity between pairs of examples with each example being a protein–compound pair. Note that the joint kernel is a quadratic sum of the protein and compound kernels which gives rise to an abstract and nonlinear joint feature space $\varphi(p, c)$ for compound–protein pairs with the kernel K being an implicit generalized dot product between $\varphi(p, c)$ and $\varphi(p', c')$. The product $K_p(p, p')K_c(c, c')$ in the above formulation implicitly corresponds to the tensor-product of the protein and compound feature spaces. It is also important to note that two examples will have a high kernel score if the corresponding proteins and compounds in the two examples are similar. The joint kernel over the training dataset $D = \{((p_i, c_i), y_i) | i = 1 \dots N\}$ is used for training a kernelized support vector machine (SVM) which is then used to infer the prediction score $f(p, c)$ for a given test example (p, c). This approach is in line with the work by Jacob and Vert (2008) with major differences in the choice of constituent kernels and construction of the joint kernel (see GitHub Supplementary Material for comparative results between these kernel methods).

3 Performance comparison and screening

We have designed multiple experiments to identify issues in performance evaluation and generalization of CPI predictors which are described below.

3.1 Cross-validation

For direct comparison with previous methods, we have used stratified 5-fold CV which is typically used for reporting CPI prediction results. Each CV experiment is repeated 10 times to obtain the average and standard deviation of different performance metrics such as AUCROC and AUC-PR.

One of the limitations of 5-fold CV is that very similar proteins or compounds may end up in different folds resulting in an overly optimistic assessment of prediction performance. To estimate the generalization performance in a real-world setting where test proteins may not share very high sequence similarity with proteins in the training set, we have performed a more stringent NRCV analysis which has not been performed in previous studies. For this purpose, proteins in the NR-HCPI dataset are first clustered based on a given sequence identity threshold through CD-HIT (Huang *et al.*, 2010). These clusters are then divided into 5-folds such that no 2-folds have examples from the same cluster while ensuring that the number of examples in every fold remains approximately the same. This guarantees that the sequence similarity of proteins in examples in a test-fold is always less than a specified threshold with proteins in the training set. We used two different sequence similarity thresholds (40% and 90%) in our analysis.

3.2 Validation over experimentally verified negative examples

We have also analyzed the prediction quality of different CPI predictors on an external set containing experimentally verified negative examples from Binding DB as described in the dataset section. In this experiment, the ML models are trained on 4-folds of NRCV as described above. However, the original negative examples in the test fold are replaced with experimentally verified negative examples from Binding DB. This process is repeated by alternating across different folds and then multiple runs to generate mean and standard deviation values of performance metrics.

3.3 Analysis of negative example generation strategies

As discussed in Section 1, there are two strategies used in the literature for generating negative examples: random pairing and similarity controlled negative example generation. In this work, we systematically compare these strategies for training and performance assessment of the proposed model. For this purpose, we have developed the algorithm shown in Algorithm 1 to generate negative examples at different inter-class similarity thresholds using kernel-based calculations. This algorithm can be used to generate a desired number of synthetic negative examples by controlling their degree of similarity to examples in a given positive set based on an inter-class similarity threshold α ∈ [0,1]. For our systematic comparison, we first pick a value of α and then generate synthetic negative examples through this algorithm for training and performance evaluation. It is important to note that for sufficiently high values of α (α→1), this algorithm essentially generates randomly paired negative examples which can be similar to known positive examples whereas for low values (α→0), the generated negative examples are highly dissimilar to known positive examples. The resulting data of positive and synthetic negative examples are then divided into 5-folds in a stratified manner as for NRCV. Similar to NRCV, training is performed on 4-folds followed by testing on examples in the held-out set in two different ways: first by using the held-out set of positive and synthetic negative examples and, secondly, by using the held-out set of positive examples and 'true' negative examples from Binding DB. The process is then repeated for different values of α. Differences in predictive performance of a given method between the CV protocol and the evaluation with true negative examples from Binding DB indicate systematic biases resulting from synthetic negative example generation strategies.

Algorithm 1. Algorithm for negative example generation with inter-class similarity α

```
Inputs:
Set of positive examples $\wp = \{(p_i, c_i) | i = 1 \ldots P\}$
Set of unique proteins $P_U$ ($P_U = \{p | (p, c) \in \wp\}$)
Set of unique compounds $C_U$ ($C_U = \{c | (p, c) \in \wp\}$)
Desired number of negative examples N (based on P:N ratio)
Desired inter-class similarity threshold $\alpha \in [0, 1]$
Output: Set $\aleph$ of N negative examples with similarity to positive
    examples $\wp$ less than $\alpha$
Algorithm:
Initialize $\aleph \leftarrow \{\}$
While $|\aleph| < N$:
    Randomly select a protein–compound pair $(p, c)$ from $P_U \times C_U$
    Calculate similarity of candidate negative example with positive set
    as follows:
            $\alpha_{pc} = \max_{p' \in P_U - \{p\}} K_p(p, p') \max_{c' \in C_U - \{c\}} K_c(c, c')$
    If $\alpha_{pc} < \alpha$ and $(p, c) \notin \wp \cup \aleph$: $\aleph \leftarrow \aleph \cup \{(p, c)\}$
Return $\aleph$
```

3.4 Target compound screening

In a practical setting, CPI prediction approaches are used for screening a large number of compounds for potential binding with a target protein of interest. Ideally, interacting compounds of a given protein should rank close to the top in comparison to non-interacting compounds in the screening library based on prediction scores of all test examples from the predictor. However, CV experiments used in previous works do not model this 'screening' use case as they are restricted to a fixed evaluation dataset and do not analyze how a predictor would rank known interacting partners in a setting in which all compounds are paired with all proteins. In this work, we have performed *in silico* screening of all unique compounds against all proteins in a given test set. This all-versus-all pairwise screening is useful for drug discovery and repurposing studies and is carried out by computing the prediction score of all possible pairs of proteins and compounds in a test set using a prediction model and calculating how often a predictor ranks a known interacting pair in its top predictions. We have performed multiple screening experiments for comparison between CPI prediction models:

3.4.1 Screening with NRCV

In this experiment, we train a model using training folds of the NRCV dataset and then compute prediction scores of all-versus-all compound–protein pairs in the test fold using the trained model (see GitHub Supplementary File for an illustration of the experimental setup). This process is repeated for all 5-folds of the dataset to compute a rank-based performance metric [rank of first positive prediction (RFPP)] described in the next subsection.

3.4.2 Screening SuperDRUG2 for drug-repurposing

For drug-repurposing analysis with the proposed model, we used the SuperDRUG2 dataset containing 3633 FDA-approved drugs. In this experiment, the model is first trained on all examples in training folds of the NRCV dataset and then used to generate prediction scores for all proteins in the test fold paired with all compounds in the SuperDRUG2 database (see Supplementary Material on GitHub for an illustration of the experimental setup). These scores are used to rank known interacting compounds of each protein in the test set relative to the compounds in SuperDRUG2 to compare the predictive performance of different models and identify putative compounds in SuperDRUG2 that can bind test proteins in the NRCV dataset.

We have also used Kernel-CPI to generate predictions for interactions of SARS-CoV-2 Spike protein and human-ACE2 protein across all compounds in SuperDrug2. We performed a literature search for any experimental evidence of interaction of the top-scoring compound with these proteins or their association with SARS-CoV-2 treatment effects.

3.5 Using ranks for performance evaluation

For quantifying the prediction quality of CPI predictors in screening experiments, we have developed an interpretable performance metric called RFPP inspired from our previous work on protein–protein interactions (Minhas *et al.*, 2014). It essentially shows the expected number of compounds that will need to be screened in the wet lab to identify at least one true interacting partner of a protein. For a given protein in the test set, RFPP is obtained by first pairing all possible test compounds with the protein and computing the prediction scores of all such examples using the CPI model under evaluation. Then the rank of the highest scoring compound that is a known interacting partner of the test protein is used as the RFPP value of that protein (see GitHub Supplementary Material for an illustration of this experimental setup). Note that for an ideal predictor, the RFPP for all test proteins should be 1, that is, the top-ranked compound of each test protein should be an interaction partner of that protein. In order to quantify the predictive quality of a CPI model across all test proteins, we first compute RFPP for all test proteins and then calculate percentiles of the RFPP values across all proteins. The percentile values across all proteins can be used to compare the predictive performance of screening models based on their ability to rank putative CPIs. The rth percentile of RFPP of a predictor will be q (denoted as RFPP(r) = q) if r% test proteins have at least one known interacting compound in the top q predictions from the predictor. For an ideal predictor, the RFPP value for all proteins in the test set should be 1, that is, RFPP (100) =1. We have generated the RFPP percentile plots of different CPI predictors. As a baseline, we have also plotted the RFPP percentiles of a random predictor which generates random prediction scores given a protein and compound. These values provide more directly interpretable estimates of prediction quality for such screening experiments.

4 Results and discussion

4.1 NRCV analysis is essential for realistic performance evaluation

Previous approaches have used 5-fold CV or multiple bootstrap runs for performance evaluation. In order to provide a direct comparison between different methods, we have performed stratified 5-fold CV on the original Liu *et al.* dataset as well as after removing duplicated examples from it (Table 1). This analysis shows that the predictive performance in terms of AUROC CPI-NN (94%) and GraphDTA (97%) is comparable to kernel-CPI baseline (99%). As expected, removal of duplicated examples reduces the prediction accuracy of these methods. In order to get a more realistic estimate of the generalization performance of these methods, we have performed 5-fold NRCV analysis with 90% sequence identity threshold as discussed in the previous section. As expected, the predictive performance of the predictors decreases significantly with the removal of redundancy between training and test sets through NRCV. These experiments clearly show that it is very important to analyze prediction performance through NRCV. Results at 40% thresholds are reported in the Supplementary Material (on GitHub) and follow a similar trend.

4.2 Validation over true negative examples from binding DB allows realistic performance evaluation

As outlined in Section 3.2, we have used a set of experimentally verified negative examples from the Binding-DB dataset to analyze the generalization performance of CPI predictors. For this purpose, these models were first trained on the NR-HCPI dataset with a balanced (1:1) class ratio. The results of this analysis are given in Table 2 which shows that, upon using true negative examples from Binding-DB in testing, both CPI-NN (AUC–ROC of 76.8%) and Graph-DTA (AUC–ROC of 61.5%) perform significantly worse than the simple kernel-CPI approach (AUC–ROC of 89.9%). This shows generalization failure of these approaches and is line with the NRCV results discussed in the previous subsection. As expected, increasing the ratio of negative examples in training for the proposed method improves the prediction performance over the binding DB test set further.

4.3 Random pairing for generating negative examples yields more realistic and better generalization performance

We have analyzed the impact of different strategies of generating synthetic negative examples (random-pairing versus inter-class similarity controlled negative example generation) on estimation of prediction quality of a CPI model through CV and its generalization performance on an external dataset containing experimentally verified negative examples from Binding-DB. For this purpose, we have used the procedure discussed in Section 3.3 that allows us to generate synthetic negative examples by controlling their degree of similarity with a given positive set through an inter-class similarity threshold α. The AUC-PR values of CPI-NN, Graph-DTA and the proposed model for CV and the external test set for different values of α are plotted in Figure 2. It shows that, as expected, as the similarity between the synthetic negative examples and the positive set is increased, the AUC-PR values of all three methods obtained from CV decrease. This is inline with the findings by Ding *et al.* (2014) and support similarity controlled generation of negative examples. However, if models trained over such 'easy' negative examples that are significantly different from the positive set are tested on an external set containing experimentally verified negative examples, the generalization performance is significantly lower. On the other hand, generalization performance over experimentally verified negative test examples improves as the value of α is increased. This experiment clearly shows that using random pairing of proteins and compounds (corresponding to $\alpha \rightarrow 1$) can be a superior strategy for generating synthetic negative examples as it not only gives more realistic estimates of predictive quality but can improve the performance of CPI models over unseen test sets in comparison to strictly controlling the degree of inter-class similarity in model training (corresponding to $\alpha \rightarrow 0$).

4.4 RFPP for interpretable performance evaluation in screening experiments

Figure 3 shows the RFPP percentiles across all proteins resulting from the all-versus-all screening experiment over the NR-HCPI dataset with NRCV detailed in Section 3.4. In this experiment, a CPI model is first trained over examples in the training folds of the NRCV dataset and then used to rank all possible pairs of proteins and compounds in the test set to see how good is the method at ranking known interacting compounds for all proteins through the RFPP metric. The total number of all such possible combinations in

Table 1. Mean and standard deviation (in brackets) of AUCs (expressed as percentage) of different CPI methods over NR-HCPI dataset with stratified 5-fold and non-redundant (NR) CV

Strategy	CPI-NN		Graph-DTA		Kernel-CPI	
	ROC	PR	ROC	PR	ROC	PR
5-Fold CV (Liu et al.)	94.41 (1.19)	94.01 (2.21)	96.51 (0.37)	95.08 (1.09)	**98.89 (0.14)**	**99.03 (0.16)**
5-fold CV Duplicates removed	93.1 (1.06)	91.44 (0.64)	87.1 (0.74)	85.60 (1.10)	**93.84 (2.35)**	**94.56 (1.31)**
5-Fold NR CV	62.58 (1.10)	72.52 (5.20)	68.21 (1.90)	67.3 (1.00)	**69.98 (5.70)**	**77.30 (1.44)**

Note: Highest values shown in bold.

Table 2. Performance analysis over true negative examples from binding-DB for different training class ratios (P:N)

Method	P:N	AUC–ROC	AUC-PR
CPI-NN	1:1	76.81 ± 9.80	47.48 ± 5.14
Graph-DTA	1:1	61.53 ± 2.55	24.52 ± 2.6
Proposed (Kernel-CPI)	1:1	89.88 ± 2.59	72.0 ± 2.02
	1:3	91.19 ± 3.56	84.3 ± 4.01
	1:5	**91.74 ± 3.35**	88.34 ± 3.09
	1:7	91.15 ± 2.15	**88.96 ± 1.86**

Note: Bold values indicate the best mean AUC percentage ± standard deviation.

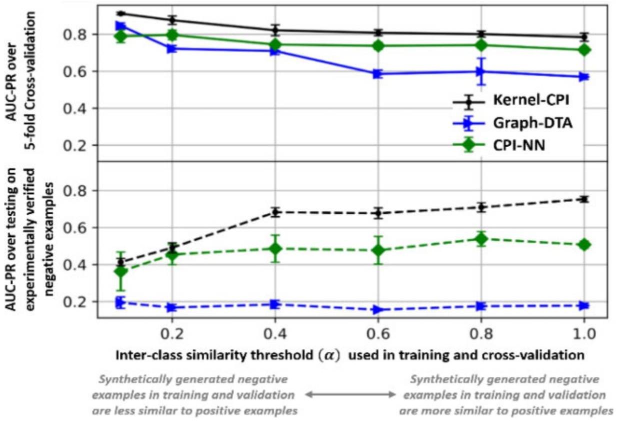

Fig. 2. Analysis of the impact of negative example generation strategies. AUC-PR (with error bars) for different CPI predictors (Kernel-CPI, Graph-DTA and CPI-NN) over different values of the inter-class similarity threshold α for CV (solid lines) and testing over experimentally verified negative examples from binding-DB

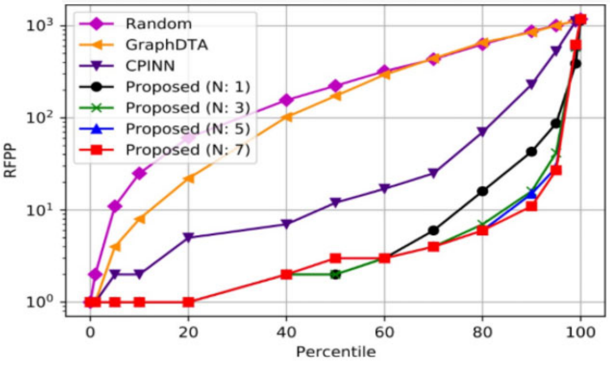

Fig. 3. Percentiles of RFPP across proteins in the NRCV screening experiment for a random predictor, Graph-DTA, CPI-NN and Kernel-CPI (proposed) for different P:N

this dataset is ∼292 500. It shows that for 85% of test proteins, the kernel-CPI baseline is able to find at least one known interacting compound of those proteins in its top 10 hits [i.e. RFPP(85) = 10] whereas for CPI-NN and Graph-DTA only 50% and 12% proteins, respectively, have at least one known hit in their top 10 predictions for each protein. In contrast, a random predictor can be expected to have at least one interacting compound in its top 10 predictions for only 5% of proteins in this test set. This clearly shows the efficacy of the proposed approach as well as the ease of interpreting results of model evaluation through RFPP in screening experiments. As expected, adding more randomly paired negative examples to training improves RFPP further.

4.5 Drug repurposing analysis using SuperDRUG2

In order to evaluate the prediction performance for possible drug-repurposing studies, we have conducted a virtual screening experiment using the FDA approved drugs in the SuperDRUG2 dataset. For this purpose, we score all possible (∼908 250) pairs of proteins from the NR-HCPI with compounds from SuperDRUG2. All these predictions from the kernel-CPI model are made available to the community as Supplementary Results. As an additional step, we have also calculated the RFPP percentiles across all proteins from different models for this screening experiment which are given in the Supplementary File. These results show that for a random predictor we can expect to find at least one true interacting compound in the top 10 hits for only 3% of the proteins in this analysis. However, CPI-NN and kernel-CPI models are able to identify at least one interacting compound for 50% and 75% of proteins, respectively.

The results of *in silico* screening of compounds in the SuperDRUG2 dataset for Human ACE2 (Uniprot ID: Q9BYF1) and SARS-Cov-2 Spike (Uniprot ID: P59594) proteins through the proposed method are given in the Supplementary File (on GitHub) which shows the top 100 predictions of our model for ACE2 and Spike protein along with evidence from the literature supporting the predicted interaction. We have found that the proposed model is able to identify a number of compounds as potential interaction partners of these proteins even though these were not included in its training. Specifically, we have identified Trandolapril, dimethyl sulfoxide (DMSO), Remdesivir, Ramipril, *N*-acetylglucosamine, perindopril, sunitinib and glutathione in our top hits for ACE2 binding with strong support from experiments and *in silico* studies in the literature. Similarly, *N*-acetylglucosamine, DMSO, Remdesivir, Sunitinib, Nilotinib, Dasatinib and Sorafenib show binding potential with the spike protein of SARS-Cov-2 with strong support in recent literature (references added in Supplementary Material).

5 Conclusions

In this work, we have identified a number of shortcomings in experiment design approaches for CPI prediction. We hope that the insights, performance assessment strategies and baselines discussed in this work will enable researchers to address these issues so that future CPI models are more effective in prediction of CPIs with higher generalization performance. Further investigation into the role of surface accessible residues in proteins and other protein feature representations can help improve prediction performance.

Funding

This paper was published as part of a special issue financially supported by ECCB2022. This work was supported by the Pakistan HEC NRPU 6085.

Conflict of Interest: none declared.

References

Ben-Hur,A. and Noble,W.S. (2006) Choosing negative examples for the prediction of protein–protein interactions. *BMC Bioinformatics*, **7**, S2. https://doi.org/10.1186/1471-2105-7-S1-S2

Bleakley,K. and Yamanishi,Y. (2009) Supervised prediction of drug–target interactions using bipartite local models. *Bioinformatics*, **25**, 2397–2403. https://doi.org/10.1093/bioinformatics/btp433

Bredel,M. and Jacoby,E. (2004) Chemogenomics: an emerging strategy for rapid target and drug discovery. *Nat. Rev. Genet.*, **5**, 262–275. https://doi.org/10.1038/nrg1317

Broach,J.R. and Thorner,J. (1996) High-throughput screening for drug discovery. *Nature*, **384**, 14–16. https://doi.org/10.1038/384014a0

Cao,D.-S. *et al.* (2013) ChemoPy: freely available python package for computational biology and chemoinformatics. *Bioinformatics*, **29**, 1092–94. https://doi.org/10.1093/bioinformatics/btt105

Chen,L. *et al.* (2019) Hidden bias in the DUD-E dataset leads to misleading performance of deep learning in structure-based virtual screening. *PLoS ONE* **14**, e0220113.

Chen,L. *et al.* (2020) TransformerCPI: improving compound–protein interaction prediction by sequence-based deep learning with self-attention mechanism and label reversal experiments. *Bioinformatics (Oxford, England)*, **36**, 4406–4414. https://doi.org/10.1093/bioinformatics/btaa524

Chen,R. *et al.* (2018) Machine learning for drug–target interaction prediction. *Molecules*, **23**, 2208. https://doi.org/10.3390/molecules23092208

Chen,X. *et al.* (2016) Drug–target interaction prediction: databases, web servers and computational models. *Brief. Bioinform.*, **17**, 696–712. https://doi.org/10.1093/bib/bbv066

Ding,H. *et al.* (2014) Similarity-based machine learning methods for predicting drug–target interactions: a brief review. *Brief. Bioinform.*, **15**, 734–747. https://doi.org/10.1093/bib/bbt056

Gilson,M.K. *et al.* (2016) BindingDB in 2015: a public database for medicinal chemistry, computational chemistry and systems pharmacology. *Nucleic Acids Res.*, **44**, D1045–D1053. https://doi.org/10.1093/nar/gkv1072

Gönen,M. (2012) Predicting drug–target interactions from chemical and genomic kernels using bayesian matrix factorization. *Bioinformatics*, **28**, 2304–2310. https://doi.org/10.1093/bioinformatics/bts360

Günther,S. *et al.* (2008) SuperTarget and matador: resources for exploring drug–target relationships. *Nucleic Acids Res.*, **36**, D919–D922. https://doi.org/10.1093/nar/gkm862

Hashemifar,S. *et al.* (2018) Predicting protein–protein interactions through sequence-based deep learning. *Bioinformatics*, **34**, i802–i810. https://doi.org/10.1093/bioinformatics/bty573

Huang,Y. *et al.* (2010) CD-HIT suite: a web server for clustering and comparing biological sequences. *Bioinformatics (Oxford, England)*, **26**, 680–682. https://doi.org/10.1093/bioinformatics/btq003

Jacob,L. and Vert,J.-P. (2008) Protein–ligand interaction prediction: an improved chemogenomics approach. *Bioinformatics (Oxford, England)*, **24**, 2149–2156. https://doi.org/10.1093/bioinformatics/btn409

Lee,H. and Lee,J.W. (2016) Target identification for biologically active small molecules using chemical biology approaches. *Arch. Pharm. Res.*, **39**, 1193–1201. https://doi.org/10.1007/s12272-016-0791-z

Lim,S. *et al.* (2021) A review on compound–protein interaction prediction methods: data, format, representation and model. *Comput. Struct. Biotechnol. J.*, **19**, 1541–1556.

Liu,H. *et al.* (2015) Improving compound–protein interaction prediction by building up highly credible negative samples. *Bioinformatics*, **31**, i221–i229. https://doi.org/10.1093/bioinformatics/btv256

Veselinovic,M. *et al.* (2015) Application of SMILES notation based optimal descriptors in drug discovery and design. *Curr. Top. Med. Chem.*, **15**, 1768–1779.

Mazandu,G.K. *et al.* (2018) Large-scale data-driven integrative framework for extracting essential targets and processes from disease-associated gene data sets. *Brief. Bioinform.*, **19**, 1141–1152. https://doi.org/10.1093/bib/bbx052

Minhas,FuAA. *et al.* (2014) PAIRpred: partner-specific prediction of interacting residues from sequence and structure. *Proteins*, **82**, 1142–1155. https://doi.org/10.1002/prot.24479

Mysinger,M.M. *et al.* (2012) Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking. *J. Med. Chem.*, **55**, 6582–6594. https://doi.org/10.1021/jm300687e

Nguyen,T. *et al.* (2021) GraphDTA: predicting drug–target binding affinity with graph neural networks. *Bioinformatics*, **37**, 1140–1147. https://doi.org/10.1093/bioinformatics/btaa921

Öztürk,H. *et al.* (2018) DeepDTA: deep drug–target binding affinity prediction. *Bioinformatics*, **34**, i821–i829.

Öztürk,H. *et al.* (2019) WideDTA: prediction of drug–target binding affinity. *ArXiv*:1902.04166, February. http://arxiv.org/abs/1902.04166

Riley,P. (2019) Three pitfalls to avoid in machine learning. *Nature*, **572**, 27–29. https://doi.org/10.1038/d41586-019-02307-y

Rohrer,S.G. and Baumann,K. (2009) Maximum unbiased validation (MUV) data sets for virtual screening based on PubChem bioactivity data. *J. Chem. Inform. Model.*, **49**, 169–184. https://doi.org/10.1021/ci8002649

Schirle,M. and Jenkins,J.L. (2016) Identifying compound efficacy targets in phenotypic drug discovery. *Drug Discov. Today*, **21**, 82–89. https://doi.org/10.1016/j.drudis.2015.08.001

Sieg,J. *et al.* (2019) In need of bias control: evaluating chemical data for machine learning in structure-based virtual screening. *J. Chem. Inform. Model.*, **59**, 947–961. https://doi.org/10.1021/acs.jcim.8b00712

Siramshetty,V.B. *et al.* (2018) SuperDRUG2: a one stop resource for approved/marketed drugs. *Nucleic Acids Res.*, **46**, D1137–D1143. https://doi.org/10.1093/nar/gkx1088

Thafar,M. *et al.* (2019) Comparison study of computational prediction tools for drug–target binding affinities. *Front. Chem.*, **7**, 782. https://doi.org/10.3389/fchem.2019.00782

Tsubaki,M. *et al.* (2019) Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. *Bioinformatics (Oxford, England)*, **35**, 309–318.

Wang,S. *et al.* (2020) LDCNN-DTI: a novel light deep convolutional neural network for drug–target interaction predictions. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, pp. 1132–1136. https://doi.org/10.1109/BIBM49941.2020.9313585

Wishart,D.S. *et al.* (2008) DrugBank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Res.*, **36**, D901–D906. https://doi.org/10.1093/nar/gkm958

Zhang,W. *et al.* (2017) Computational multitarget drug design. *J. Chem. Inform. Model.*, **57**, 403–412. https://doi.org/10.1021/acs.jcim.6b00491

Zhang,X.-M. *et al.* (2021) Graph neural networks and their current applications in bioinformatics. *Front. Genet.*, **12**, 690049. https://doi.org/10.3389/fgene.2021.690049

Zhao,T. *et al.* (2021) Identifying drug–target interactions based on graph convolutional network and deep neural network. *Brief. Bioinform.*, **22**, 2141–2150. https://doi.org/10.1093/bib/bbaa044