Hierarchical Clustering Split for Low-Bias Evaluation of Drug-Target Interaction Prediction

Peizhen Bai[1], Filip Miljković[2], Yan Ge[1], Nigel Greene[3], Bino John[3*], and Haiping Lu[1*]

[1]Department of Computer Science, University of Sheffield, Sheffield, UK
[2]Imaging and Data Analytics, Clinical Pharmacology & Safety Sciences, R&D, AstraZeneca, Gothenburg, Sweden
[3]Imaging and Data Analytics, Clinical Pharmacology & Safety Sciences, R&D, AstraZeneca, Waltham, USA
{pbai2, yge5, h.lu}@sheffield.ac.uk, *{filip.miljkovic, nigel.greene, bino.john}@astrazeneca.com*

***Abstract*—Drug-target interaction (DTI) prediction is important in drug discovery and chemogenomics studies. Machine learning, particularly deep learning, has advanced this area significantly over the past few years. However, a significant gap between the performance reported in academic papers and that in practical drug discovery settings, e.g. the random-split-based evaluation strategy tends to be too optimistic in estimating the prediction performance in real-world settings. Such performance gap is largely due to hidden data bias in experimental datasets and inappropriate data split. In this paper, we construct a low-bias DTI dataset and study more challenging data split strategies to improve performance evaluation for real-world settings. Specifically, we study the data bias in a popular DTI dataset, BindingDB, and re-evaluate the prediction performance of three state-of-the-art deep learning models using five different data split strategies: random split, cold drug split, scaffold split, and two hierarchical-clustering-based splits. In addition, we comprehensively examine six performance metrics. Our experimental results confirm the overoptimism of the popular random split and show that hierarchical-clustering-based splits are far more challenging and can provide potentially more useful assessment of model generalizability in real-world DTI prediction settings.**

***Index Terms*—Drug-target interaction, data bias, data splitting strategy, performance evaluation**

I. Introduction

Predicting drug-target interaction (DTI) plays an important role in the drug discovery process, where drugs are chemical compounds and targets are usually proteins. The availability of high-confidence DTI pairs is crucial for screening candidate compounds in novel drug development and providing insights on the causes of side effects between existing drugs. Traditional DTI prediction via in vitro experiments is reliable but has high monetary cost and long development cycle, preventing their usage on a large scale. Therefore, machine learning-based in silico approaches have gained more attention and developed rapidly over the past few years.

Recently, machine learning, particularly deep learning, has advanced many areas such as knowledge discovery and natural language processing. Machine learning methods can learn powerful representations for discrete data, such as words and entities. We can embed such discrete data into a low-dimensional and dense vector space to capture their relationships [16]. Therefore, we can embed discrete drugs and targets into vector spaces similarly. Furthermore, chemogenomics [1] integrates the compound chemical space and protein genomic space into a unified framework, enabling the development of many machine learning-based methods for DTI prediction [2,6,7,9]. Not only 3D structure information, but also various chemical and genomic features have been introduced to train predictive machine learning models, e.g. simplified molecular-input line-entry system (SMILES) for compounds, and amino acid sequence for proteins. These common input features accelerated the development of machine learning methods on large-scale DTI databases.

Despite the growing interests in machine learning for drug discovery and the reported progress, a large performance gap still exists between academic research and industrial application, where academic results tend to be over-optimistic for industrial settings [17]. Similar to the three pitfalls in machine learning pointed out in [14], we identify three common pitfalls that cause high-bias DTI performance evaluation.

Inappropriate data splitting. A common practice in machine learning research is to split training and test sets at random and evaluate model performance by the accuracy on the test set (under the assumption that the training and test data have the same distribution). In the context of drug discovery, such random split tends to overestimate model performance in real-world settings. One important reason is that drug compounds in the same series share the same scaffold or large substructure, which is easy to learn as long as a few molecules of this series are contained in the training set. However, in real applications, chemists often need DTI prediction on a new compound series different from known compounds, which is more challenging and makes random split inappropriate.

Low-confidence negative samples. Although many public databases exist for generating DTI data, researchers often ignore hidden bias during model training. The lack of highly confident negative samples is one of them. Machine learning is a data-driven technique and model performance depends heavily on data quality. DTI papers often randomly generate negative samples from unobserved pairs for training and evaluation, which leads to low confidence because they may include some unknown true positive samples.

*Corresponding author

978-1-6654-0126-5/21/$31.00 ©2021 IEEE

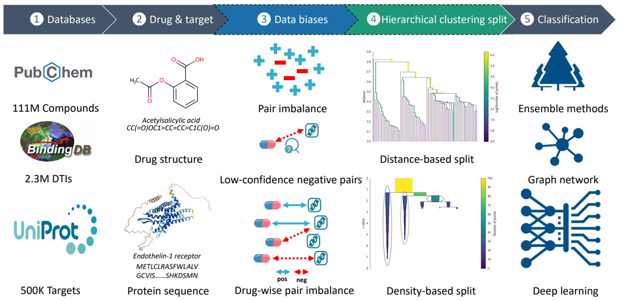

Fig. 1: Bias control in a general DTI prediction workflow.

Drug-wise pair imbalance. The final important pitfall is the hidden imbalance between positive and negative pairs for each drug. In DTI datasets [4, 8, 18], it is common that most drugs have only one type of pairs (positive or negative). For these drugs, models can make correct prediction using only drug information without learning appropriate DTI patterns, leading to significant drug-wise pair imbalance. Subsequently, the evaluation result is over-optimistic and the model has poor generalization.

This paper focuses on addressing the three pitfalls above to reduce bias in training and evaluating machine learning models. We compare five different DTI data split strategies, one of which is our newly proposed density-based hierarchical clustering split. The results show that different data splits can significantly affect model performance comparison. Clustering-based splits lead to more challenging tasks that can better reward generalization rather than memorization. In experiments, we strictly control the hidden data bias in a benchmark dataset and experimentally validate all negative DTI pairs to ensure both interaction types exist for each drug. This leads to a low-bias dataset to encourage learning correct interaction information for better DTI prediction.

II. Materials and Methods

The DTI prediction task can be viewed as a binary classification of whether a drug forms biological interaction with a target of interest or not. Drugs are commonly encoded as 1D sequences (SMILES) or 2D molecular graphs, and target proteins are typically represented as amino acid sequences. Figure 1 shows a general DTI prediction workflow and our bias-controlled evaluation.

A. Low-bias dataset construction

Most DTI datasets are not originally designed for training machine learning models. They have hidden data bias and tend to produce over-optimistic results. We propose to reconstruct the experimental datasets following two bias removal guidelines:

- *High-confidence negative samples should be used.* The best option is to select experimentally validated pairs. We can set a safe margin in measured binding affinity to select negative samples [3]. We can also employ some similarity-based DTI negative sampling algorithms [10].
- *The number of drugs containing only one interaction type should be removed or reduced.* Many DTI experiments only consider the imbalance between positive and negative pairs across the whole dataset, e.g., by keeping a fixed ratio without considering the pair imbalance for individual drugs, leading to prediction based only on drug features rather than drug-target interaction.

B. Classic data split strategies

We introduce three classical and one clustering-based data split strategies: random split, cold drug split, scaffold split and single-linkage split [11].

- **Random split** is the most popular data split strategy. DTI pairs are randomly split into train, validation, and test with given ratios.
- **Cold drug split** first randomly splits drugs into train/validation/test, and then puts all DTI pairs associated with individual drugs in corresponding sets as the final splits.
- **Scaffold split** is based on 2D molecular structures that partition drugs into different bins according to their Murcko scaffolds. These bins are then randomly split into train/validation/test sets so that all drugs associated with a bin are part of the same set. Next, all DTI pairs associated with the drugs in a bin are assigned to the corresponding sets.
- **Single-linkage split** is a clustering-based strategy to ensure that the distances between clusters are always larger than a pre-defined threshold. This strategy can cluster drugs by their chemical fingerprints such as ECFP4 [15] in this study, and then apply Jaccard distance on binarized ECFP4 to measure the pairwise distance between drugs. For single-linkage hierarchical clustering, each cluster of drugs will only be assigned to one of the train, validation and test sets.

C. HDBSCAN for data split

However, single-linkage split uses a single distance threshold that cannot separate clusters of different densities, which are common in drug compound series. Therefore, we propose a Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) strategy [12] to split data. HDBSCAN is a hierarchical clustering method that transforms the original distances between data points to density. It is designed for clusters of varying densities. We investigate the performance differences between HDBSCAN split and other split strategies. We review HDBSCAN briefly below.

1) Calculate the mutual reachability distance (MRD): To find clusters, HDBSCAN first computes the MRD between data points a and b as:

$$d_k(a,b) = \max\left\{\mathrm{core}_k(a), \mathrm{core}_k(b), d(a,b)\right\}, \tag{1}$$

where k is a hyperparameter indicating the number of nearest neighbors, $\mathrm{core}_k(a)$ is the core distance between the core a and its k-th nearest neighbor, and $d(a,b)$ is the distance between a and b with the original metric. This MRD metric allows dense points with low core distance to remain at the same distance from each other while sparse points with high core distance are pushed away.

2) Build a minimum spanning tree (MST): After getting MRD, an MST is built from a weighted graph. In this graph, vertices are the data points and the weight of an edge between any two points is their MRD. Then a tree is built one edge at a time by adding the edge with the lowest weight. Meanwhile this added edge needs to bridge the current tree and a vertex that is not in this tree. Given the MST, the next step is to convert it into cluster hierarchy. HDBSCAN sorts edges in MST by distance with ascending order and then iterates to create a new merged cluster for each edge.

3) Condense the cluster tree: A minimum cluster size is introduced as a hyperparameter. At each hierarchy split, the sizes of newly generated clusters are compared with the minimum cluster size. If a new cluster has fewer points than the minimum cluster size, HDBSCAN declares them to be "points falling out of a cluster". If the size of a new cluster is equal to or larger than the minimum cluster size, HDBSCAN treats it as a true cluster split to persist in the tree.

4) Compute the stability and extract clusters: To extract clusters from the condensed cluster tree, HDBSCAN defines a stability, which aims to choose clusters that persist and have a longer lifetime. First, for a given cluster in hierarchy, HDBSCAN defines a measure λ inversely proportional to the distance threshold, so λ_{birth} denotes the λ when the cluster is split from its parent cluster. Each falling point p in the cluster has a value λ_p so that each cluster c has a stability s:

$$s(c) = \sum_{p \in c} \left(\lambda_p - \lambda_{\text{birth}} \right). \tag{2}$$

Now traverse the condensed cluster tree from all leaf nodes to root. If the sum of the stabilities of child clusters is greater than the stability of their parent cluster, the parent cluster stability is set to be the sum of the child stabilities. Otherwise, the parent cluster is selected to be one of the final clusters.

III. Experiment and Results

A. Experimental setup

1) Dataset: we construct a low-bias version of binary BindingDB [4, 5] dataset in this experiment. Following the IC_{50} threshold used by Gao et al. [3], we consider a drug-target pair to be positive if its IC_{50} is less than 100 nm, and negative if its IC_{50} is greater than 10,000 nm, which is a 100-fold difference.

2) Bias-reducing preprocessing: Due to the drug-wise pair imbalance, 91% of drugs only have one type of pairs (positive or negative) in the binary BindingDB dataset. This implies that we can train a model to make right classification without considering protein information for DTI pairs associated with the 91% of drugs. High classification accuracy does not indicate successful learning of correct DTI patterns.

Therefore, we further process the data by removing all DTI pairs of drugs containing only one pair type. This gives us a low-bias dataset with 29,674 positive samples and 32,752 negative samples. Figure 2a shows the drug probability distribution in terms of log ratios of positive to negative samples in the dataset, which is calculated as:

$$ln^i_{ratio} = ln \frac{N^i_{pos}}{N^i_{neg}}, \tag{3}$$

where N^i_{pos} is the number of positive interactions for drug i, and N^i_{neg} is the number of negative interactions. Following the steps above, our constructed dataset addresses common pitfalls and has the following benefits:

- The number of positive and negative samples is balanced.
- All negative samples are experimentally validated and highly confident.
- The drug-wise pair imbalance in DTI pairs is removed.

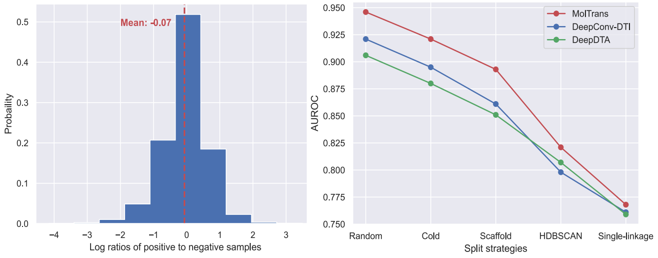

Fig. 2: (**a**) Drug probability distribution in terms of $ln(N^i_{pos}/N^i_{neg})$ in the constructed low-bias dataset. The red line indicates the mean log ratio for all drugs (mean=-0.07). $ln(N^i_{pos}/N^i_{neg}) = 0$ when the number of positive and negative interactions are equal for drug i. (**b**) Comparison of three models using five different split strategies.

3) Metrics: We use AUROC, AUPRC, and accuracy as the major metrics to measure model performance. We also report the sensitivity and specificity metrics at the best F1 score.

4) Split strategies: We study five split strategies in model performance evaluation: random split, cold drug split, scaffold split, single-linkage split, and HDBSCAN split, as detailed in Section II.

We set the distance threshold to 0.5 for single-linkage split and minimum cluster size to 5 for HDBSCAN split. Each split strategy keeps a 7:1:2 ratio for training/validation/test sets. We conduct five independent runs with different random seed for each split. We compare three algorithms below on the same splits.

5) Learning algorithms: Three state-of-the-art deep learning DTI models are selected for performance comparison:

- **DeepConv-DTI** [9] models DTI using convolutional neural network (CNN) and one global max-pooling layer to learn protein sequence features, and one fully connected layer to encode drug fingerprints ECFP4.
- **DeepDTA** [13] uses CNN on both drug SMILES string and protein amino acid sequence to extract local residue patterns. As the original DeepDTA is a regression model to predict binding affinity. A sigmoid function is added after the last layer of decoder for binary classification.
- **MolTrans** [6] adapts transformer architectures to encode drug and protein information, and introduces an interaction map module with a CNN layer to learn the interaction between molecular substructures.

Split strategy	AUROC	AUPRC	Accuracy	Sensitivity	Specificity	Test loss
Random	**0.946±0.003**	**0.935±0.004**	**0.874±0.003**	**0.838±0.01**	**0.914±0.007**	**0.469±0.019**
Cold drug	0.921±0.003	0.909±0.006	0.841±0.004	0.798±0.009	0.889±0.005	0.67±0.052
Scaffold	0.893±0.006	0.874±0.004	0.804±0.005	0.736±0.012	0.882±0.014	0.797±0.099
HDBSCAN	0.821±0.024	0.778±0.031	0.724±0.037	0.581±0.091	0.891±0.03	0.936±0.328
Single linkage	0.768±0.024	0.717±0.025	0.676±0.032	0.483±0.077	0.894±0.023	0.959±0.224

TABLE I: MolTrans performance comparison with five different data split strategies (**Best**, Worst).

6) Implementation: We follow the same model hyperparameter settings described in the original papers. The batch size is 64 and each model is allowed to run 100 epochs in each independent training. The learning rate is set to $1e^{-5}$ with the Adam optimizer. The test model is selected at the epoch giving the best AUROC on the validation set. The selected model is evaluated on the test set and metrics are reported.

B. Performance gap between different split strategies

To investigate whether there is a significant performance gap for the same method with different data split strategies, Table I shows the performance of MolTrans with different metrics on the test sets generated by different strategies. As expected, random split has significant information overlap between training and test sets, so it achieves the best performance across all metrics. However, random split's good performance on the test set does not imply the same good performance in real drug discovery, where it is unlikely to have so much prior information while predicting a novel drug-target interaction pair in reality. The performance declines differently in other split strategies. Compared with random split, single-linkage split has the largest performance drop of 18.8% in AUROC and 23.3% in AUPRC.

For other split strategies, the performance drop is also evident. Cold drug split has just a slight drop since it randomly selects drugs without considering drug similarities. Scaffold split further divides drugs by their shared scaffold, so it is more challenging than the cold split. However, as scaffold split considers only well-defined cyclic substructures connected by the linkers, scaffolds that share the topology may still be considered dissimilar. Thus, scaffolds proximal in the fingerprint space can cause information leakage and model bias if not found in either training or test set. As clustering-based split strategies cluster similar compounds irrespective of their scaffolds, and as part of the same subset, the potential for data leakage is reduced.

We further study the influence of data split strategy on performance of different models. Figure 2b plots AUROC of MolTrans, DeepConv-DTI and DeepDTA on the five split strategies. Although MolTrans always outperforms the other two models, their performance gap gradually decreases with the change of split strategies. Comparing random and single-linkage split, the improvement of MolTrans over DeepDTA drops from 4.4% to 1.1%. Moreover, DeepConv-DTI outperforms DeepDTA with random split, but DeepDTA outperforms DeepConv-DTI with HDBSCAN split. This shows that the better performance on random split strategy is over-optimistic because the test set rewards more memorization rather than generalization.

IV. Conclusion

This work studied low-bias evaluation of machine learning models in DTI prediction. Experimental results showed that traditional split strategies tend to overestimate predictive performance and exaggerate performance gaps between different models. We constructed a low-bias dataset, and adopted two clustering-based split strategies toward more realistic evaluation in drug discovery. Clustering-based splits created the most challenging prediction tasks for evaluating real-world DTI prediction performance.

References

[1] M. Bredel and E. Jacoby. Chemogenomics: an emerging strategy for rapid target and drug discovery. *Nature Reviews Genetics*, 5(4):262–275, 2004.

[2] L. Chen, X. Tan, D. Wang, et al. TransformerCPI: improving compound-protein interaction prediction by sequence-based deep learning with self-attention mechanism and label reversal experiments. *Bioinformatics*, 2020.

[3] K. Y. Gao, A. Fokoue, H. Luo, et al. Interpretable drug target prediction using deep neural representation. In *IJCAI*, pages 3371–3377, 2018.

[4] M. K. Gilson, T. Liu, M. Baitaluk, et al. BindingDB in 2015: a public database for medicinal chemistry, computational chemistry and systems pharmacology. *Nucleic Acids Research*, 44(D1):D1045–D1053, 2015.

[5] K. Huang, T. Fu, W. Gao, et al. Therapeutics Data Commons: Machine learning datasets and tasks for drug discovery and development. *NeurIPS Datasets and Benchmarks*, 2021.

[6] K. Huang, C. Xiao, L. M. Glass, et al. MolTrans: Molecular interaction transformer for drug–target interaction prediction. *Bioinformatics*, 37(6):830–836, 2021.

[7] M. Karimi, D. Wu, Z. Wang, et al. DeepAffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks. *Bioinformatics*, 35(18):3329–3338, 2019.

[8] S. Kim, J. Chen, T. Cheng, et al. PubChem 2019 update: improved access to chemical data. *Nucleic Acids Research*, 47(D1):D1102–D1109, 2019.

[9] I. Lee, J. Keum, and H. Nam. Deepconv-DTI: Prediction of drug-target interactions via deep learning with convolution on protein sequences. *PLoS Computational Biology*, 15(6):e1007129, 2019.

[10] H. Liu, J. Sun, J. Guan, et al. Improving compound–protein interaction prediction by building up highly credible negative samples. *Bioinformatics*, 31(12):i221–i229, 2015.

[11] A. Mayr, G. Klambauer, T. Unterthiner, et al. Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. *Chemical Science*, 9(24):5441–5451, 2018.

[12] L. McInnes, J. Healy, and S. Astels. HDBSCAN: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11):205, 2017.

[13] H. Öztürk, A. Özgür, and E. Ozkirimli. DeepDTA: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, 2018.

[14] P. F. Riley. Three pitfalls to avoid in machine learning. *Nature*, 572:27–29, 2019.

[15] D. Rogers and M. Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010.

[16] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. *NeurIPS*, pages 3104–3112, 2014.

[17] I. Wallach and A. Heifets. Most ligand-based classification benchmarks reward memorization rather than generalization. *Journal of Chemical Information and Modeling*, 58 5:916–932, 2018.

[18] D. S. Wishart, Y. D. Feunang, A. Guo, et al. DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Research*, 46:D1074 – D1082, 2018.