ABSTRACT

Structure-based virtual screening is a promising in silico technique that integrates computational methods into drug discovery. The most extensively used method in structure-based virtual screening is molecular docking. However, the docking process is not computationally efficient and simultaneously accurate due to classic mechanics-based scoring functions. These can only approximate, but not reach, quantum mechanics precision. In order to reduce the computational cost of the protein-ligand scoring process and use data-driven approaches to boost the scoring function accuracy, deep learning non-docking methods can be used by utilizing 3D structure or 1D sequence information of the protein target. This method can minimize the error inherited from molecular docking methods and avoid the extensive computational cost of docking. Furthermore, these two methods are integrated into an easy-to-use framework, CarbonAI, that provides both choices for researchers. Graph neural network (GNN) is employed in the 3D version and BiLSTM has been adopted in the sequence version of CarbonAI, respectively. To verify our approaches, different experiments were performed on two datasets, an open dataset Directory of Useful Decoys: Enhanced (DUD.E) and an in-house proprietary dataset without computer generated artificial decoys (NoDecoy). On DUD.E we achieved a state-of-the-art AUC of 0.981 and on NoDecoy we achieved an AUC of 0.974 whereas on the conventional docking program, the respective AUC performance is less than 0.8. The CarbonAI engine also reaches a state-of-the-art enrichment factor at top 2 percent for 36.2 folds. We have also retrospectively validated the CarbonAI models with various wet lab experimental data, and the results demonstrated a consistently accurate performance. Furthermore, the inference speed of the engine was benchmarked using the openly available 2021 Enamine REAL Database (RDB), that comprises over 1.36 billion molecules in 4050 core-hours using our CarbonAI non-docking method (CarbonAI-ND). The inference speed of CarbonAI-ND is about 36000 molecule / core-hour, compared to typical docking methods’ speed of 20, which is about 16000 times faster than conventional docking method. Overall, the experiments indicate that CarbonAI is accurate and computationally efficient with good generalization to different molecular targets for virtual screening.

Keywords Virtual Screening · Biomolecules · Machine Learning

1 Introduction

Drug discovery process is a well known time-consuming and expensive task. The discovery of new drugs goes through several development stages, each of which could give out thousands of millions of molecules for screening[16]. Formerly, random screening and empirical advice were utilized for this task. Then, high-throughput screening (HTS) improves the drug discovery process by offering automated and much quicker screening of large chemical libraries against a molecular target[17]. However, HTS is still considered to be time-consuming, laborious, and expensive with a relatively low screening hit rate that range from 0.001% to 0.1%[6][29].

With the rapid development of computer science, computational methods are extensively used in drug design now. Specifically, virtual screening (VS) is one of the most promising in silico techniques that works as a filter, and structure-based virtual screening (structure-based virtual screening) is a robust and powerful VS that predicts the
Inference speed which is a common problem in structure-based virtual screening that the performance drops notably if the testing with sequences of proteins and simplified molecular-input line-entry system (SMILES) [39] strings of ligands as inputs.

A conventionally used methodology in structure-based virtual screening is molecular docking [13, 8, 20]. It is a Gold 5118 CPU @ 2.30GHz (GNN), and they also brought attention mechanism [34] into their model to mitigate the problem of generalizability, both methods on two large datasets, an open dataset Directory of Useful Decoys: Enhanced (DUD.E) [25] and an in-house experimental data, and establish a framework containing both of them, as illustrated in Figure 1. We evaluate interaction between a compound and a molecular target protein [13, 5]. structure-based virtual screening utilizes the three-dimensional (3D) structure information of molecular targets, hence, it normally produces a better performance than ligand-based VS, which does not contain any structure information [1].

Furthermore, deep learning (DL) is becoming a prominent tool in VS. Compared to traditional machine learning (ML) algorithms [16, 6], such as neural network [9], support vector machine (SVM) [19] and random forest (RF) [2], deep learning models require minimum feature engineering. Ideally speaking, if enough data is provided, representations or features can be learned directly from the dataset without any human design [3, 4, 22]. An early successful DL example is Quantitative Structure-Activity Relationship (QSAR) [36], which operates over molecular fingerprints. Wallach et al. [37] find a way to apply 3D convolutional neural network (3D-CNN) to a protein-ligand complex which is represented on a 3D grid. Ragoza et al. [28] further extended it to include active and inactive binding poses classification. These 3D-CNN methods successfully outperform previous works and also improve the accuracy of predicting absolute binding affinities [38, 2, 10].

Nevertheless, docking-based VS is not a perfect answer. The main problems of docking are that most docking processes are time and computing power consuming, docking results are not highly reliable, and balanced and high-quality data is limited. In order to resolve them, non-docking methods that avoid using any docking programs are proposed. A common framework is to integrate representations of ligands and proteins into a single neural network without knowing 3D binding structures. Karimi et al. [18] introduced their DeepAffinity to predict the pIC50 of protein-ligand complexes with sequences of proteins and simplified molecular-input line-entry system (SMILES) [39] strings of ligands as inputs. Gao et al. [12] developed a siamese network consists of a recurrent neural network (RNN) and a graph neural network (GNN), and they also brought attention mechanism [34] into their model to mitigate the problem of generalizability, which is a common problem in structure-based virtual screening that the performance drops notably if the testing ligand and protein are not seen in the training set. Lee et al. [23] also reported their models working on sequences of proteins and SMILES strings of ligands. Nguyen et al. [26] and Lim et al. [24] further developed GNN based models for structure-based virtual screening tasks.

In this paper, we propose a docking-based (CarbonAI-Dock) and a Non-docking (CarbonAI-ND) method for structure-based virtual screening tasks, and establish a framework containing both of them, as illustrated in Figure 1. We evaluate both methods on two large datasets, an open dataset Directory of Useful Decoys: Enhanced (DUD.E) [25] and an in-house dataset, NoDecoy. We also evaluate our methods on DUD, a subset of DUD.E. On DUD.E, we achieved an $AUC_{\text{CarbonAI-Dock}} = 0.926$ and a state-of-the-art $AUC_{\text{CarbonAI-ND}} = 0.981$. On NoDecoy dataset, our results are $AUC_{\text{CarbonAI-Dock}} = 0.911$ and $AUC_{\text{CarbonAI-ND}} = 0.974$, proving the robustness and generalizability of our methods. We did not do any deep learning fine-tuning on all our experiments, which implies a potential of getting better performances. Besides, in order to demonstrate CarbonAI-ND is computationally efficiency and accurate, we performed inferences on databases such as Enamine REAL Database (RDB) [30] and in-house experimental data, and successfully boosted the inference speed more than 16000 times by circumventing the docking process.
2 Methods

Our goal is to develop both a docking-based and a non-docking structure-based virtual screening (VS) method. Given this goal, we briefly introduced a general GNN model and the GNN model used in-house. Following this overview, the structure-based virtual screening methods are discussed comprehensively.

2.1 Data Curation

The first step in building the neural network model is data acquisition and curation. The primary backbone of the dataset is derived from PDBbind (v2020)[42]. This version of PDBbind contains over 19,000 structures derived from PDB[43] databases. This dataset was directly used for training in the CarbonAI-Dock model after protein-ligand docking 3D representation was extracted. For CarbonAI-ND, additional protein and ligand conversion steps were performed. First, the base protein structure dataset was fixed to the 17,679 structures from PDBbind. All of these structures contain whole-protein and pocket 3D PDB file format structures. For this study, only the pocket file was used in the training step. In the CarbonAI-ND structure model, the pocket file was directly used for GNN modeling, whereas this data was converted into a 1D sequence string for the CarbonAI-ND sequence model.

For small molecule compounds, additional data was also used to boost the base dataset in both CarbonAI-ND cases. For this step, data from ChEMBL (V27, 2020)[45] and BindingDB (2020 version)[44] are extracted. Specifically, small molecule data in the SMILES format and assay IC50 data were extracted from the databases based on UniProt IDs which match with available PDB structures from the base protein structure dataset. In addition, for a more domain accurate model, only UniProt IDs from human species were kept during this data curation step. After the extraction process, data de-duplication and basic data cleansing operations based on previous studies[46] were performed on the extracted ligand SMILES data including removing salts and solvents from the final SMILES structures which are known as cleaned-smiles in the final dataset. For de-duplication of IC50 assay data, all compound-target-assay records were compiled together. The newest records had the highest priority, and if conflicts were found, the largest (least active) data record was used. Then, the assay data was split into two different data columns in the final training set. The standard-value was the normal scaled molar value; the pchembl-value represented the log-scale converted value. Finally, all errors were discarded from the final dataset, and each SMILES string contained one or more protein pocket records.

In total, CarbonAI-Dock used only the 17,679 structures from this version of PDBbind. CarbonAI-ND used structures from 1,379 human proteins from the CarbonAI-Dock dataset and over 500,000 corresponding ligands from the above data curation steps. This represents a 26-times boost in data compared to base models greatly increasing the chemistry coverage of this new dataset.

2.2 Graph Neural Network

Currently, graph neural networks (GNN) are widely used in drug discovery processes, as ligands and proteins can be modeled as graphs by nature[40]. A graph is a pair $G = (V, E)$, where $V$ is a set of nodes and $E$ is a set of edges, and
an edge is a set of paired nodes. GNNs model both ligands and proteins as graphs, in which the nodes are atoms and the
dges are defined simply by connecting atoms that lie within a predefined cutoff distance \( c \), then GNNs represent each
atom via an atom embedding layer \( u \in \mathbb{R} \) and each edge via an edge embedding layer \( v \in \mathbb{R} \).

Most GNNs in this area can be summarized as a two-stage architecture, as known as the message-passing framework [14]: propagate node information among each other by neighborhood aggregation on each layer. Then, form the whole graph representation by a read-out function. Each layer of such GNNs can be written as:

\[
\tilde{h}_v^{l+1} = f_{\theta}^{l}(\tilde{h}_v^l, \{\tilde{x}_e : e \in \mathcal{N}_E(v)\}, \{\tilde{h}_v^l : v' \in \mathcal{N}_V(v)\}).
\]

where \( \tilde{h}_v^l \) is the node feature vector of node \( v \) at layer \( l \). \( \mathcal{N}_V(v) \) and \( \mathcal{N}_E(v) \) denote sets of nodes and edges connecting to the node \( v \), and \( f_{\theta}^{l}(\cdot) \) is a parameterized function.

The readout function \( R \) pools node features from the final iteration \( K \) to obtain the entire graph’s representation \( h_G \):

\[
\tilde{h}_G = R(\{h_v^K | v \in G\}).
\]

In our implementation, our GNN model can be formally described as:

\[
\tilde{h}_v^{l+1} = \phi(\text{concat}(h_v^l, \sum_{u \in \mathcal{N}(v)} h_u^l + \max_{u \in \mathcal{N}(v)} h_u^l))
\]

where \( \phi \) is a multilayer perceptron (MLP) network and concat layer concatenates features.

2.3 CarbonAI-Dock

CarbonAI-Dock is a docking-based method that takes a docking result generated by Autodock Vina (Version 1.1.2) [33] as an input, and output the whole docked protein-ligand complex representation via a single GNN. The pseudo-codes are presented in Algorithm 1. First, using the output of docking programs, which is the 3D structure of a protein-ligand complex, to form a graph by setting atoms as nodes, and connecting each two nodes if they are within a cutoff distance \( c \). Instead of modeling the whole protein-ligand complex into a graph, we solely put the atoms of the ligand into a graph and ignore atoms of the protein for now. Then, the graph is extended by adding atoms of the protein whose distances to any atom of the ligand are within the cutoff distance \( c \). Finally, a GNN is applied to the extended graph. This is because the protein-ligand interaction is vital and informative for the final prediction.

**Algorithm 1:** CarbonAI-Dock. \( f \) is the graph neural network that will be
tained. \( \text{dist}(i, j) \) is the distance between node \( i \) and \( j \). \( d_j \) is the minimum distance of node \( j \) to the ligand, defined as \( d_j := \min_{i \in V \setminus j} \text{dist}(i, j) \)

**input:** ligand atoms \( V_l \), protein atoms from a docked protein-ligand complex \( V_p \), cutoff distance \( c \)

\( V := \{i | i \in V_l \} \cup \{j | j \in V_p, d_j < c\} \)

\( E := \{i, j | i, j \in V, \text{dist}(i, j) < c, \ i \neq j\} \)

\( G := (V, E) \)

\( y \leftarrow f(G) \)

return \( y \)

In this study, there were several reasons that the whole protein-ligand complex as a graph was not modeled. First, a
typical protein size is normally much larger than a ligand. Proteins typically contain over 2000 atoms, compared to
50 in a ligand. Even a protein pocket, which is significantly smaller than a protein, can easily contain more than 500
atoms, still ten times the size of a ligand. Therefore, a protein-ligand complex graph will be dominated by the protein
data, increasing the difficulty for GNNs to differentiate active and inactive compounds. Also, the number of proteins is
much smaller than the number of compounds in most structure-based virtual screening datasets, in which over 1000 or
even 10000 compounds are paired with a single protein, making most protein-ligand complex graphs in the training
set structurally similar to each other and further increases the difficulty of training. Finally, a docked protein-ligand
complex graph can be too large for a GNN to train, causing serious efficiency and computing power problems.

2.4 CarbonAI Non-docking

There are three main problems where docking-based methods typically suffer. First, most docking processes are time
consuming as well as computational resource intensive. As listed in Table 1, it took several months to finish the DUD.E
docking process with multiple parallel computing clusters. It is either too time-consuming or less approachable for those
who are not accessible to high-performance computing. Next, docking programs are not giving out analytical solutions.
Errors are introduced during the docking stage, then propagate through follow-up models, eventually harming the final performances. Finally, high-quality datasets containing 3D binding poses are hard to acquire. Thus, Non-docking methods can be a vital supplement to structure-based virtual screening tasks.
Algorithm 2: CarbonAI-ND. $f_c$ is the GNN we need to train for ligand and $f_p$ is the GNN for pocket, $u$ is the atom embedding layer.

**input:** ligand graph $G_c = (V_c, E_c)$, protein pocket graph $G_p = (V_p, E_p)$; cutoff distance $c$  

$h_u^p \leftarrow u(V_p)$  

$h_u^c \leftarrow u(V_c)$  

$h_p \leftarrow f_p(h_u^p, E_p)$  

$h_c \leftarrow f_c(h_u^c, E_c)$  

$h \leftarrow \text{concat}(h_p, h_c)$  

$y \leftarrow \text{MLP}(h)$  

**return** $y$

However, unlike ligand modeling, the best method to model a protein remains a main problem that is unclear and worth discussing in structure-based virtual screening tasks. Figure 4 shows the statistical analysis of this problem on our in-house NoDecoy dataset, which is described in Table 2 and Section 3 in detail. Coefficient of Variation (CoV) in Figure 4 is defined as the ratio of the standard deviation to the mean: $c_v = \sigma / \mu$. It is a standardized measure of dispersion of a probability distribution, a smaller $c_v$ indicates that the data concentrates more on a certain range and easier to be modeled.

Modeling between amino acids or atoms is an important distinction and area of debate. A protein is made up of amino acids, and an amino acids are made up of atoms. Therefore, a protein can be modeled with both methods. In this study, CarbonAI-ND features both structure-based and sequence-based models. The structure-based model features atom embeddings, while the sequence-based model uses only the amino acid sequence for protein modeling. After training the initial models, only one of the models were used for the virtual benchmarks based on the training metrics.

Another important modeling decision to make was to decide between the whole protein structure or just the pocket region. Figure 4.a and Figure 4.b report $c_v$ about 16% and Figure 4.c and Figure 4.d report $c_v$ about 32%. From Figure 4, it is clear that, whether we choose to model in amino acids or atoms, modeling pockets only instead of a full protein is easier for GNNs. Additionally, Figure 4 also indicates that a protein is about 4 to 6 times larger than a pocket, regardless of amino acids or atoms, resulting in more time and computing power consumption.

Pseudo-code of the CarbonAI-ND structure-based model is presented in Algorithm 2. Firstly, two individual graphs are formed via the same methodology stated in the docking-based method using the ligand and the protein separately. They are initiated by the same atom embedding layer, then two individual GNNs are applied to get the representations. Finally, the representations are concatenated as an input to a classifier neural network, which is a MLP in our implementation. Note that weights are not shared between the two GNNs (except the atom embedding layer), since ligands and proteins are different from each other in attributes, sizes and features. The framework is also displayed in Figure 3. While we seek to maintain it as simple as possible, multiple optional modules are available, such as replacing the MLP with a CNN or a RNN module, adding attention mechanisms to the MLP, etc. These extensions are displayed in Figure 3 where instead of modeling the protein structure, a protein sequence is modeled instead. In this case, all protein graph inputs are changed to be a sequence $S_p$; $f_p$ is replaced with a transformer, $t_p$, and there is no atom embedding layer.

### 3 Experimental Settings

**Datasets** We analyzed our models and methods on two datasets, a public dataset and a dataset combining both public and in-house proprietary data. With such a setting, we attempted to evaluate our approach’s generalizability and reproducibility. The number of proteins and ligands of both datasets are listed in Table 2. Additionally, we performed inference on RDB with both CarbonAI-Dock and CarbonAI-ND to compare the computational efficiency, as reported in Table 1 together with real-world case-studies comparing actual experimental data.
CarbonAI, A Non-Docking Deep learning based small molecule virtual screening platform

Figure 4: (a) and (b) denote the distributions of numbers of amino acids and atoms in a pocket; (c) and (d) denote the distributions of numbers of amino acids and atoms in a protein. Distributions in (c) and (d) are right-skewed with higher variation and higher CoV comparing to (a) and (b), thus, they are harder to be modeled by GNN.

- **DUD.E**: An enhanced and rebuilt version of DUD(Directory of Useful Decoys)\[25\]. It contains 102 protein targets. We trained and evaluated our model on both the full DUD.E dataset and its subset DUD, which contains 38 proteins out of 102 as shown in Table 2.
- **NoDecoy**: A combination of DUD.E, excluding its computer generated decoys, and our in-house proprietary dataset which contains biochemical data of 261987 compounds and 83 targets using IC$_{50} = 10$M as cutoff to differentiate active and inactive compounds. We excluded decoys from DUD.E in order to eliminate implicit computer generated patterns, and to balance the proportion of positive and negative samples. Then we combined the remaining compounds with our proprietary dataset, which contains no decoys by nature.
- **RDB**: Enamine REAL database (RDB)\[30\] is the largest enumerated database which comprises over 1.36 billion synthetically feasible molecules in the version tested in our benchmark. We used RDB for inference only in this paper to simulate a regular screening task, since it is such a large database that can reveal the computational efficiency of CarbonAI-ND.

**Cross Validation** The performances of CarbonAI are assessed in two types of cross validation (CV). First, Leave-One-Out (LOO) CV was used as a technique for assessing how the method will generalize to independent data. Specifically, the model will leave one protein and all compounds pairing with it out in rotation, and train on all other remaining proteins and compounds, then do model testing on the left-out protein and compounds. We compared our results with DeepVS\[27\] and Lim \[24\]. Next, k-fold CV was used where the dataset is randomly partitioned into k equal sized subsets. Of the k subsets, a single subset is retained as the validation set, and other k-1 subsets are used as the training set. The CV process is repeated k times, with each subset used as the validation set exactly once. In our experiments, k=5.

**Evaluation Metrics** To evaluate our approaches, we select two extensively used metrics for comparison: the area under the Receiver Operating Characteristics curve (AUC) which is generally one of the most important evaluation metrics for classification problems\[11\], and the enrichment factor (EF). EF is used to indicate how good the prediction is on the top x% ranked compounds. The EF at x% is computed as:

$$EF_{x\%} = \frac{\text{actives at } x\%}{\text{compounds at } x\%} \times \frac{\text{total compounds}}{\text{total actives}}$$  \(4\)
CarbonAI, A Non-Docking Deep learning based small molecule virtual screening platform

Figure 5: CarbonAI-Dock leave-one-out(LOO) cross validation(CV) performance, each point represents an AUC score from LOOCV, and proteins are sorted in ascending order by AUC.

Deep Learning Settings  We tried to maintain the training and testing procedure as simple as possible, avoiding too many tricks in order to verify the robustness and generalizability of our methods. Therefore, all models are trained from scratch without any pretraining. No grid-searches or other fine-tuning methods were used for this study. Better performances are expected to be achievable with more deliberate fine-tuning. Additionally, all deep learning models used the SGD optimizer with a constant learning rate and were all trained individually on a single P100 GPU.

4 Results

4.1 CarbonAI-ND 5-fold CV results

Both CarbonAI-Dock and CarbonAI-ND were trained using a 5-fold CV strategy. For the CarbonAI-ND models, an average AUC score was obtained for both the sequence-based model and the structure-based model. For the structure-based CarbonAI-ND model, an average AUC score of 0.988 was obtained. Similarly, an average AUC score of 0.939 was obtained for the sequence-based CarbonAI-ND model. This shows that although only amino acid residue-level features instead of atom-level features were modeled in this model, there was not a large dropoff in model sensitivity showing that the two independent Neural Network models can indeed build a successful model for structure-based virtual screening tasks. The initial 5-fold CV tests showed that the structure-based CarbonAI-ND model had better predicting power, justifying the use of this model as a benchmark. Furthermore, the decision to use this model in benchmarks is that a typical structure-based virtual screening dataset, including DUD.E and our in-house NoDecoy, contains much more compounds than proteins, as reported in Table 2, so looking at the ligand binding prediction scoring is of more importance than the protein itself.

4.2 DUD and DUD.E LOO experiment

The results of CarbonAI-Dock are reported in Figure 5. Figure 5 (left) shows the LOO CV AUC of each of 102 target proteins from DUD.E, most of which achieves an AUC >0.90, and only 10 AUCs are under 0.80. Figure 5 (right) shows each of 38 target proteins from DUD. Next, we tested CarbonAI-ND with the same settings, as shown in Figure 6. Comparing to CarbonAI-Dock, $AUC_{mean}$ has boosted from 0.926 to 0.981, and most AUCs are above 0.95 with $AUC_{min}$ improved from 0.628 to 0.831, revealing that CarbonAI-ND is more robust and of greater generalizability.

Comparisons of AUC and EF to methods listed in other literatures are reported in Table 3 and Table 4. They show that both CarbonAI-Dock and CarbonAI-ND perform well. Specifically, CarbonAI-ND acquires a mean EF@2% = 39.7 on DUD.E, mean EF@2% = 36.2 on DUD, and $AUC_{mean}$ = 0.981 on DUD.E and $AUC_{mean}$ = 0.974 on DUD, all are state-of-the-art based on existing works at the moment.

4.3 NoDecoy 5-Fold experiment

To evaluate our methods and framework at production stage, we eliminate decoys from DUD.E and integrate its true active ligands into our in-house dataset to form a new dataset, namely NoDecoy. Afterwards, we perform a 5-fold CV experiment by a random split. As illustrated in Figure 7, we acquired an $AUC_{mean}$ = 0.902 for CarbonAI-Dock and $AUC_{mean}$ = 0.937 for CarbonAI-ND. Considering the dataset contains no decoys and active ratio = 78.19%, which is much more balanced than DUD.E whose active ratio = 1.59% as listed in Table 2, we find the performance quite
Figure 6: CarbonAI-ND leave-one-out (LOO) cross validation (CV) performance, each point represents an AUC score from LOOCV, and proteins are sorted in ascending order by AUC.

Figure 7: CarbonAI-Dock vs CarbonAI-ND performance in the k-fold cross validation, where k=5 in our experiment. Metrics are averaged on each epoch by k folds.

satisfying. EF@2% maintains a constant number of 1.279 because it is the maximum value we can achieve, since in the equation below, we have: \( \frac{\text{total ligands}}{\text{total actives}} = \frac{1}{\text{Active Ratio}} = 1.279 \) In addition to AUC and EF, we also evaluate the prediction accuracy. Figure 7 reveals that: CarbonAI-ND outperforms CarbonAI-Dock in all metrics we assessed; performances of CarbonAI-ND raise and converge faster; the AUC curve of the CarbonAI-ND fluctuates less, indicating a better model stability; the AUC and accuracy curves of both methods are still raising, it is expected that better performances may be reached given more training epochs.

4.4 NoDecoy 5-Fold experiment

Table 1 reports that CarbonAI-Dock is faster than CarbonAI-ND by itself, but the docking process is too time-consuming that CarbonAI-Dock spends most of its running time on docking and becomes incomparable to CarbonAI-ND efficiency-wise. The RDB test is valuable for the reason that performing inference on thousands to millions of compounds is not a rare thing in the drug discovery process. Therefore, this is a good benchmark that measures the inference speed of one of the largest available datasets. The core-hours CarbonAI-ND took for RDB is a fairly good result that about 360,000 compounds can be screened in an hour with a single CPU, and the speed is scalable with parallel processing clusters.

Table 3: Retrospective study results comparing assay readouts with the two different CarbonAI-ND models.

<table>
<thead>
<tr>
<th>Target</th>
<th>Indication</th>
<th>Sequence based Accuracy (%)</th>
<th>Structure based Accuracy (%)</th>
<th>Sequence based Top 5 Recall (%)</th>
<th>Structure based Top 5 Recall (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLI family</td>
<td>Cancer</td>
<td>25</td>
<td>25</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>CDK family</td>
<td>Cancer</td>
<td>11.1</td>
<td>42.9</td>
<td>20</td>
<td>60</td>
</tr>
<tr>
<td>FGFR family</td>
<td>Cancer</td>
<td>25</td>
<td>25</td>
<td>40</td>
<td>40</td>
</tr>
</tbody>
</table>
4.5 Results from retrospective studies

Three retrospective cases have been applied using the protein and pocket-based models from CarbonAI-ND’s structure and sequence-based models. There were three different targets tested from different projects: glioma-associated oncogene (GLI), cyclin-dependent kinase (CDK), and fibroblast growth factor receptor (FGFR) families. These cases all aim to evaluate both approaches’ value using real-world assays from in-house wet-lab applications. Table 3 summarizes the results from retrospective studies using this methodology from three different targets, all oncology related. The GLI family target’s result were from cellular assays with IC50 readouts from 5-50 micromolars. Both CDK and FGFR targets are enzymatic assays with IC50 readout results in the nanomolar range.

Studies in Table 3 indicated in these internal assays, there are at least 2 discovered active compounds in the Top 5 ranked compounds from the scoring functions calculated using the structure-based or sequence-based model. These results show that in a typical primary screening assay, active hits can be discovered more efficiently from this approach compared to traditional high throughput screening methods as these assays all identified active compounds from 5-10 total compounds. Future studies with different targets in other therapeutic areas should also be evaluated to further investigate this methodology and evaluate metrics.

5 Discussion

In summary, the results from this study’s experiments led to several conclusions. The non-docking methods possess advantages over docking-based ones based on results from our experiments. GNNs can play a significant role in drug design, and an increasing number of methods have started to use GNN to model molecules now, especially in virtual screening. Datasets from virtual screening process can be fairly large, thus, it is vital for VS methods to balance performance and efficiency. Our RDB experiment and real-world retrospective studies showed that CarbonAI-ND had the capability to meet this expectation.

5.1 Is GNN a solution to structure-based virtual screening tasks?

Although GNN would not be the only answer, and there are definitely other models worth investigating. A review article from Li et al. in 2020 has listed multiple machine learning-based scoring functions for structure-based virtual screening tasks. Some other methods include support vector machines (SVM), random forest (RF), and convolution neural networks (CNN)[41]. However, since ligands and target proteins can be modeled as graphs in a straightforward way without need for additional feature engineering such as molecular fingerprints or other types of molecular interaction features, and theories of GNN are developing rapidly, we expect structure-based virtual screening tasks to benefit more from GNN models in future studies. Therefore, GNNs would likely be a good state-of-the-art solution as a virtual screening tool.

5.2 CarbonAI-ND vs. CarbonAI-Docking

Comparing non-docking compared to docking-based methods is still a question that is being investigated as more state-of-the-art models are released. In non-docking approaches, there are still a few problems worth noticing related to building efficient and effective models.

First the model size poses a major limitation. The CarbonAI-ND non-docking model is almost twice the size of the docking-based ones since it contains two independent GNNs. Although it can be imagined that reducing the model size by trimming models skillfully can be of help, the docking-based method would still be the preferred on for model-size sensitive tasks.

Next, in terms of computing power cost and speed, there is a strong motivation to reduce computing time and save computing power in drug discovery process. While CarbonAI-ND takes more time in both training and inference, things change when the full virtual screening pipeline is considered. This is because the actual docking process, a known time-consuming step, can be circumvented in the non-docking method. One docking tool, AutoDock vina[33], requires approximately 90-180 seconds per molecule based on in-house use-cases averaging out to around 20 molecules per core-hour. This is compared to the 36,000 molecules per core-hour based on the RDB benchmark shown in Table 1 representing an efficiency increase of over 16,000 times. Autodock vina also requires a 3D molecule input, whereas SMILES inputs are 1D strings, so conversion from 1D to 3D also requires additional computation time. These are important limitations in performing inference with docking-based models. In short, the docking-based method is faster only when docked protein-ligand complexes are provided in 3D file formats like mol2 or pdb when this work has been done ahead of time.
Finally, pocket detection poses a problem. The binding pockets from proteins were provided along with the dataset. However, this is not always the case, and in real-world applications, such as the GLI family protein listed in Table 3, the binding pocket is not necessarily known. If pockets are not provided, we have to implement pocket detection, which can be done by various methods\cite{21, 31, 7} mentioned in previous studies focusing on this task. The situation is similar to docking in that the analytical solution is not available, and computational tasks are required to calculate this information. Overall, although much more efficient and faster than docking, pocket detection still takes time for computation and may not necessarily give the most accurate output compared to co-crystal structures from PDB.

5.3 Conclusion

Drug discovery has long been a tough task. It has progressed significantly ever since computational methods got involved into the field. Hopefully, our work can inspire more thoughts about structure-based virtual screening, and the analysis and discussions about ligand and protein modeling can slightly bridge the gap between drug discovery and computer science.

References


