Graph neural networks and molecular docking as two complementary approaches for virtual screening: a case study on Cruzain

Adriano M. Luchi, J. Leonardo Gómez Chávez, Roxana N. Villafañe, Germán A. Conti,
E. Rafael Perez, Emilio L. Angelina, Nélida M. Peruchena**

Lab. Estructura Molecular y Propiedades, IQUIBA-NEA, Universidad Nacional del Nordeste,
CONICET, FaCENA, Av. Libertad 5470, Corrientes 3400, Argentina

ABSTRACT

The idea behind virtual screening is to first test compounds computationally in order to reduce the number of compounds that need to be screened experimentally, thus reducing the time and cost of physical experiments. Molecular docking is the most popular virtual screening technique, it predicts the binding of candidate compounds to the protein target by modeling the interactions at the binding pocket. Despite being widely used, docking accuracy is often low due to the difficulty of modeling inherently complex biological systems. On the other hand, state of the art deep neural networks, like Graph Convolutional Networks (GCNs) are able to capture the complex non-linear relationships between structural and biological data, but they lack the interpretability of structure-based modeling. In this work we took advantage of the activity data from a quantitative High Throughput Screen (HTS) of ~200K compounds against Cruzain (Cz) to retrospectively evaluate the ability of a docking algorithm and a Graph Convolutional Network for prioritizing the active compounds from the dataset. We then propose strategies to combine both techniques in a single virtual screening pipeline in order to exploit their orthogonal benefits. By plugging in the atomic embeddings learned by the GCN into the docking algorithm by means of pharmacophoric restraints, docking ability to retrieve the active ligands was enhanced. Moreover, by applying the GCN as a pre-docking filter, the compound's library was enriched in active molecules and subsequent docking of the filtered library achieved significantly higher hit rates. This work aims to be a proof of concept of the usefulness of combination strategies involving deep learning and classical molecular docking techniques, in the context of drug discovery.

1. Introduction

The main goal of the drug discovery process is to find a chemical compound that can fit both geometrically and chemically into a specific cavity on the molecular target. Conventional drug design methods include automatic High Throughput Screening (HTS) of small molecule libraries to identify, by biological testing and affinity assays, those capable of binding to a certain molecular target, typically a protein receptor or enzyme. Although HTS campaigns screen from tens of thousands to millions of compounds, since chemical space is so vast (~10^{60} molecules), any collection of compounds covers an insignificant portion of this space. This explains why about half of the HTS experiments fail. In addition, HTS is an expensive technique and requires a substantial investment in infrastructure and assay development. The aforementioned disadvantages, as well as the attractiveness of a more deterministic (i.e. more rational) approach to combat diseases gave rise to the Computer-Aided Drug Design (CADD).[1]

The idea behind virtual screening is to first test compounds computationally in order to reduce the number of compounds that need to be screened experimentally, thus reducing the time and cost of physical experiments.[2] Currently, molecular docking is the most popular virtual screening technique for prioritizing candidate compounds from large datasets. Docking algorithms consist basically in two components: a conformational search algorithm for binding mode prediction (usually referred to as pose) and a scoring function to assess the binding affinity of the poses.

Despite being widely used, docking accuracy is substantially low as a result of the trade-off for the speed required in virtual screening of large compound libraries. As a consequence, virtual screening outcomes are often plagued by high false-positive rates, namely many compounds that rank highly against a given target protein do not actually show activity.[3]

Back in 2010, Ferreira et al. undertook a parallel docking and HTS screen of 197861 compounds against Cruzain (AID 1478 dataset), a thiol protease target for Chagas disease, looking for reversible, competitive inhibitors.[4] Both techniques identified different classes of inhibitors from the library, suggesting that they complement each other and that docking's weaknesses are orthogonal to those of HTS.

Ideally, docking should be able to recover all the active chemotypes from the compounds library so that experimental testing needs to be performed only to confirm docking predictions. However, if docking were used in advance to prioritize the compounds to be tested, two of the five active chemotypes in the AID 1478 dataset, that were only recovered by HTS, would have been missed. These results make it evident that the promise of virtual screening replacing the HTS was yet to be achieved at that time.

Since the seminal work of Ferreira et al.[4] to date, machine learning (ML) methods have emerged, as a promising alternative to molecular docking virtual screenings. They have been employed for a while to develop novel scoring functions (SF) which outperformed standard docking SFs.[5] More recently, classical ML methods are being replaced by Deep Learning (DL) representations that can automatically learn features from data and achieve much higher expressive power due to their inherently deep network architectures.[6]

The purpose of training DL models in drug discovery-related problems is elucidating correct structure-activity relationships from existing data. In practical terms, this means to find a function f able to perform the mapping Y = f(X) between structure X and biological activities Y of chemical compounds. Due to the high expressivity of deep network architectures, this can in principle be done with a general-purpose fully connected Multi-Layer Perceptron (MLP) network. However, the representation of molecules for relevant information extraction poses a major challenge and makes it necessary to develop specialized DL frameworks to tackle domain-specific problems, as already exist in fields like the Recurrent Neural Networks (RNN) for speech recognition or the Convolutional Neural Networks (CNN) for image classification.

DL approaches based on fully connected neural networks have been widely used to code the molecular characteristics into vector-shaped data, but in practice, these approaches inevitably discard some part of the structural information.[7] Instead, Graph Convolutional Networks (GCN), a type of CNN designed to work directly on graphs, are the natural choice for processing information contained in chemical structures, since atoms and bonds in molecules can be represented as the nodes and edges of the graphs, respectively.

Graph neural networks (GNN) have gained a lot of attention in several chemistry applications such as molecular properties prediction,[8,9] molecular design,[10] chemical reactions, among others.[11] For instance, Ryu et al.[12] developed several flavors of GCNs for compound physicochemical property prediction, from a classical (vanilla) GCN to augmented versions that incorporate attention and gate mechanisms.

At first glance, prediction of physicochemical properties of molecules (i.e. octanol–water partition coefficient (logP), topological surface area (TPSA), etc.), which are encoded to a great extent on the ligand topology itself, seems to be a less challenging task than prediction of their pharmacological activities, which largely depends on protein-ligand interactions.

However, Sakai et al.[13] recently demonstrated that GCNs which rely solely on 2D structural information of compounds can predict not only physicochemical properties but even the activity of compounds against a particular molecular target of interest. They have shown that Graph Convolutional Network (GCN) models constructed solely from the two-dimensional structural information of compounds demonstrated a high degree of activity predictability against 127 diverse targets from the ChEMBL database. Therefore, if sufficient experimental data is available and there are enough nodes hidden layers, a simple 2D representation could quantitatively predict activity with satisfactory accuracy.

Considering the state-of-the-art results of DL models for activity prediction we decided to perform a comparative study of a Graph Convolutional Network (GCN) against a molecular docking screening, for compound activity prediction on Cruzain.

It is important to note that molecular docking is a structure-based approach that explicitly accounts for the molecular interactions of the ligand with the protein structure, while the GCN we have trained only considers the ligand topology (i.e. ligand-based approach). There are some "structure-based" GCN implementations that train two GCNs separately for both interaction partners, either protein-small molecule[14] or protein-protein complexes,[15] which are then passed to a Fully Connected Layer (FCL) to make the binding affinity prediction. This approach better resembles molecular docking solutions where a ligand is docked against the target structure and then a scoring function ranks the poses (in this analogy the FCL would be the scoring function). However, neither of those implementations takes into account the intermolecular interactions explicitly.

On the other hand, Lim et al.[6] proposed a GCN that can extract intermolecular interactions as graph features directly from the 3D structural information on the protein-ligand binding pose. While the GCN implementation by Lim et al. does consider the intermolecular interactions, it might require the availability of hundreds of known protein-ligand 3D structures to train a target-focused network. In the case of our target of study, Cruzain, there were only 37 structures deposited in the Protein Data Bank (as of May 2022), which is an insufficient amount for "data-hungry" deep learning architectures.

On the contrary, ligand-based GCN implementations rely only on the topology of the ligand molecules and their corresponding activity annotations against the protein target, both of which are vastly available for Cruzain in public databases like PubChem, ChEMBL, among others.

Consequently, in this work we carried out a retrospective virtual screening of a large dataset of compounds from a quantitative high-throughput screening assay for Cruzain inhibitors by both approaches, structure-based molecular docking, and ligand-based GCN.

Most previous benchmarking studies involving DL and docking methods were aimed to probe the superiority of DL over docking and to encourage the use of the trained DL models as a subrogate of classical docking methods. However, the main goal of this study, besides comparing the ability of both approaches for retrieving the active ligands from the dataset, was to explore how they might complement each other to increase hit rates in the context of virtual screening campaigns.

Combination approaches of both DL and docking, much like Ferreira et al.[4] have done in the past for HTS and docking, would exploit the benefits of both, the complexity of DL models and the intuitivity and interpretability of molecular docking.

2. Results and Discussion

Ferreira et al.[4] undertook a quantitative High Throughput Screen (qHTS) of a ~200K compound library against Cruzain (Cz) to search for reversible, competitive inhibitors of the enzyme. The results of the qHTS screen were deposited in PubChem (Assay ID 1478). In this work we exploited that information to retrospectively evaluate the ability of a docking algorithm and a Graph Convolutional Network for prioritizing the active compounds from the dataset. We then propose strategies to combine both techniques in a single virtual screening pipeline.

2.1. Retrospective Virtual Screening by Molecular Docking

Fig. 1 shows the binding cleft of Cruzain bound to a known vinyl sulfone inhibitor (PDB: 2OZ2). For docking calculations, the inhibitor was removed from the structure and the compounds from the AID 1478 were docked instead.

Fig 2 shows the performance of the docking algorith for discriminating between active and inactive compounds.

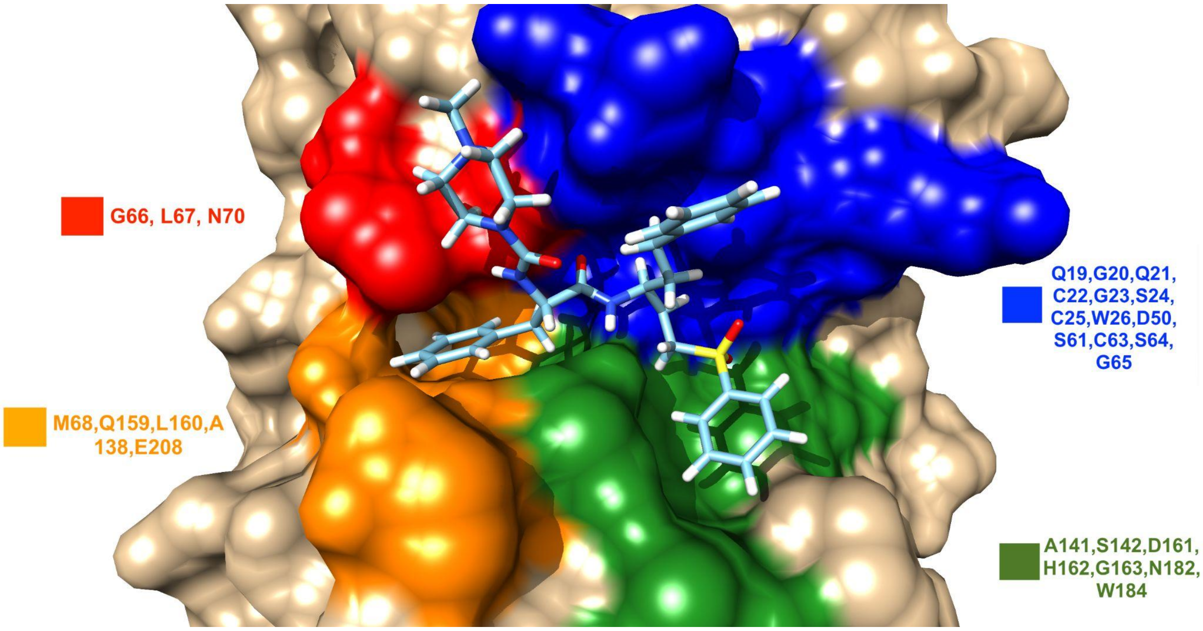

🟥 G66, L67, N70
🟧 M68,Q159,L160,A 138,E208
🟦 Q19,G20,Q21, C22,G23,S24, C25,W26,D50, S61,C63,S64, G65
🟩 A141,S142,D161, H162,G163,N182, W184

Figure 1. Cruzain surface with peptidelike vinyl sulfone inhibitor bound at the enzyme binding cleft (PDB 2OZ2). Residues that shape up the sub-pockets S1, S1’, S2 and S3 are depicted in blue, green, orange and red, respectively.

As evidenced by the low AUC in Fig 2 (orange curve), the docking algorithm struggles to classify the compounds correctly when the entire dataset is screened. This is likely because the AID 1478 dataset is a collection of compounds assayed for Cruzain inhibition that was not specifically designed for benchmarking molecular docking programs.

Visual inspection of inactive compounds misclassified as active (false positives) reveals that the high classification error rate might be due in part to the very high structural similarity between subsets of inactive and active compounds.

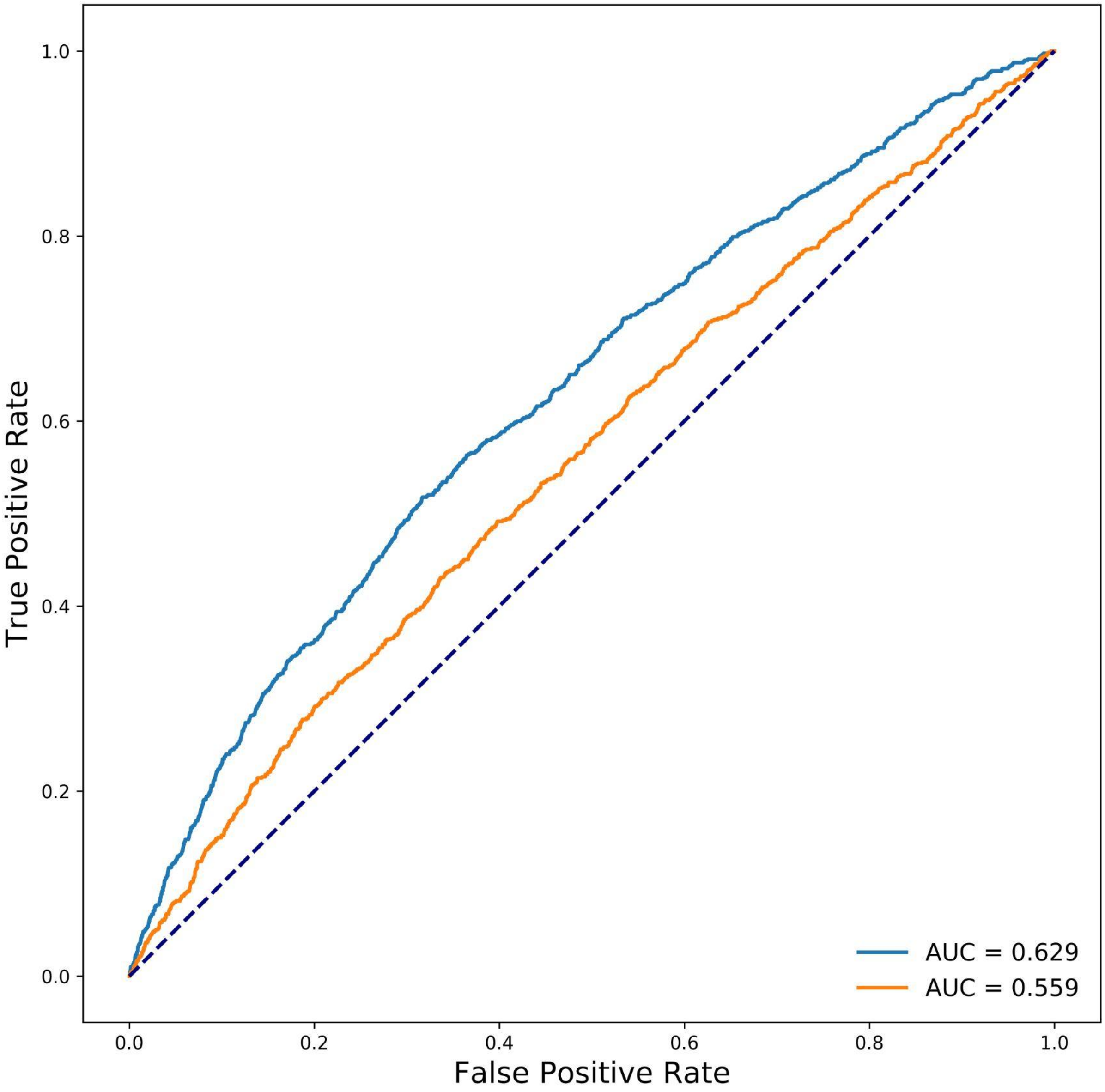


Figure 2. ROC Curve and Area Under the Curve (AUC) achieved by molecular docking of the full (orange curve) and similarity-filtered (blue curve) AID1478 dataset. The dashed diagonal line represents performance equal to random choice.

In a typical library for docking benchmarking, inactive compounds or "decoys" are randomly selected from large datasets of molecules in such a way that they will have similar physicochemical properties but different 2D topologies than actual inhibitors.

A major weakness of decoy libraries is that the difference between the two chemical spaces defined by the active compounds on one hand, and the decoy compounds on the other, may lead to an artificial overestimation of the enrichment. In other words, they may provide an overly optimistic assessment of the docking performance. Nevertheless, activity prediction on "real" qHTS datasets like the AID 1478 seems to be an extremely challenging task for docking algorithms, as judged by the very low AUC value achieved.

In view of these findings, we decided to slightly tune the AID 1478 dataset by filtering out inactive compounds whose structural fingerprints have a Tanimoto similarity coefficient (Tc) higher or equal than a given threshold with respect to active compounds. This procedure partially resembles common recipes to construct decoys libraries for docking benchmarking.[16]

By gradually decreasing the similarity threshold we found that at a Tc= 0.70 the docking algorithm reached an AUC ~ 0.63 which represents an improvement over the docking performance on the original dataset (Fig 2, blue curve). However, it is clear that these results are far from being optimal and that docking protocol requires some kind of readjustment to reach better performance.

2.2. Retrospective Virtual Screening by a Graph Convolutional Network

Encouraged by the results of Sakai et al.[13] which demonstrated the potential of Graph Convolutional Network for activity prediction, in this work we implemented a modified version of the GCN architectures (vanilla GCN, GCN+a, GCN+g and GCN+a+g) developed by Ryu et al.[12] to perform activity classification of compounds in the AID1478 dataset. Details of the dataset preparation and GCN architecture are presented in the computational methods section.

The learning curve in Fig 3 shows the training progress of the GCN augmented with both attention and gate mechanisms (GCN+a+g) which was the one that showed the best performance (see Table1 in computational details section). As can be seen in the figure, after 40 epochs, the validation curve has reached a plateau, namely increasing the number of epochs beyond that number of epochs would result in model overfitting. Therefore, we employed the strategy of “early stopping” for reducing overfitting, which is an effective and simple technique for regularization in deep learning. It is based on the fact that, on training a deep neural network, the training error will progressively decrease, but it does not necessarily happen the same with the validation error. The training of the model stops when the validation curve starts to rise again[17] or does not result in further error reduction. If the training continues beyond that step, the learned parameters will change and the model finally will overfit the data.

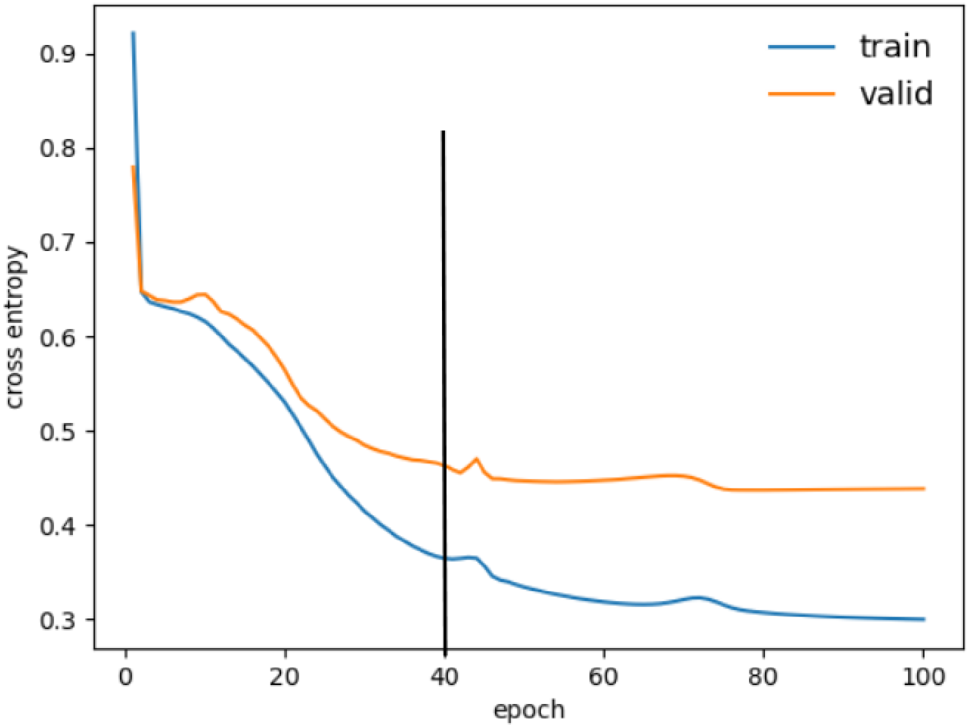


Figure 3. Learning curve showing the progress of the training process on train and validation sets. The black line shows when the training stops.

The performance of the final GCN model for classifying the examples was evaluated by the ROC-AUC metric, on a separate set reserved for testing (hold-out test set). As shown in figure 4 the augmented graph network (GCN+a+g) outperformed the docking algorithm in prioritizing active ligands from the AID 1478 dataset. It is important to point out that, unlike docking, in this case, there was no need to filter out the inactive compounds more topologically alike to the active ones, to achieve an acceptable performance of the GCN model (see fig 2). That is to say, the AUC achieved by the GCN+a+g corresponds to the testing on a stratified random sample from the entire dataset.

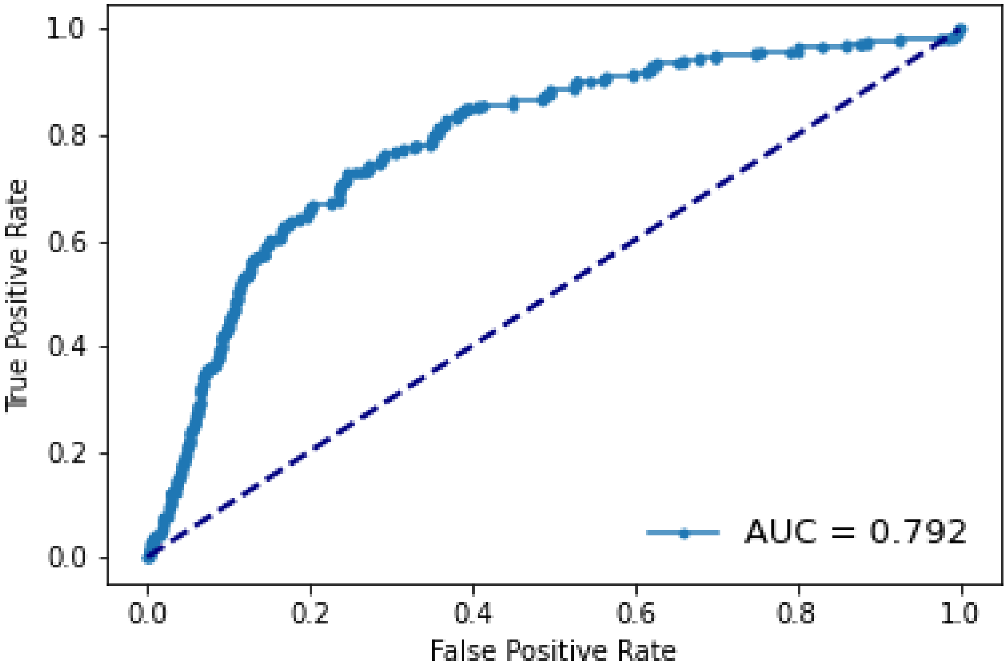


Figure 4. ROC Curve and Area Under the Curve (AUC) achieved by the GCN model on the test set. The dashed diagonal line represents a hypothetical model which does not perform better than chance.

As molecular modeling practitioners, we must admit that these results took us by surprise. Molecular docking is a structure-based approach that explicitly accounts for the molecular interactions of the ligand with the protein structure whereas the implemented GCN+a+g does not.

Therefore, we expected a better performance with the first technique.

Moreover, molecules are capable of taking various conformations depending on the number of their degrees of freedom, and only specific conformations are normally involved in their pharmacological mode of action. The trained GCN model does not take into account this conformational information and yet it performs better than molecular docking. Presumably, as argued by Sakai et al.[13] this is in part because preferred conformation is inherent to the chemical structure in many cases, namely, the 2D topologies already contain the key determinants of pharmacological actions.

Moreover, fig 5 shows the distribution of molecules in the validation set. Each point represents a molecule colored by its activity class (blue for active, red for inactive). The plot was constructed by performing a t-SNE dimensionality reduction on the output vectors from the GCN+a+g readout layer. t-SNE takes into account the local structure as well as the global structure and allows us to observe the presence of clusters.[18] In this case, we used the validation set instead of the test set due to the severe class imbalance in the latter, that make it difficult to visualize the distribution of active points in the t-SNE plot.

Active and inactive molecules are fairly well-separated (with some mixed examples), as evidenced in the figure by the two clusters of blue and red points, respectively.

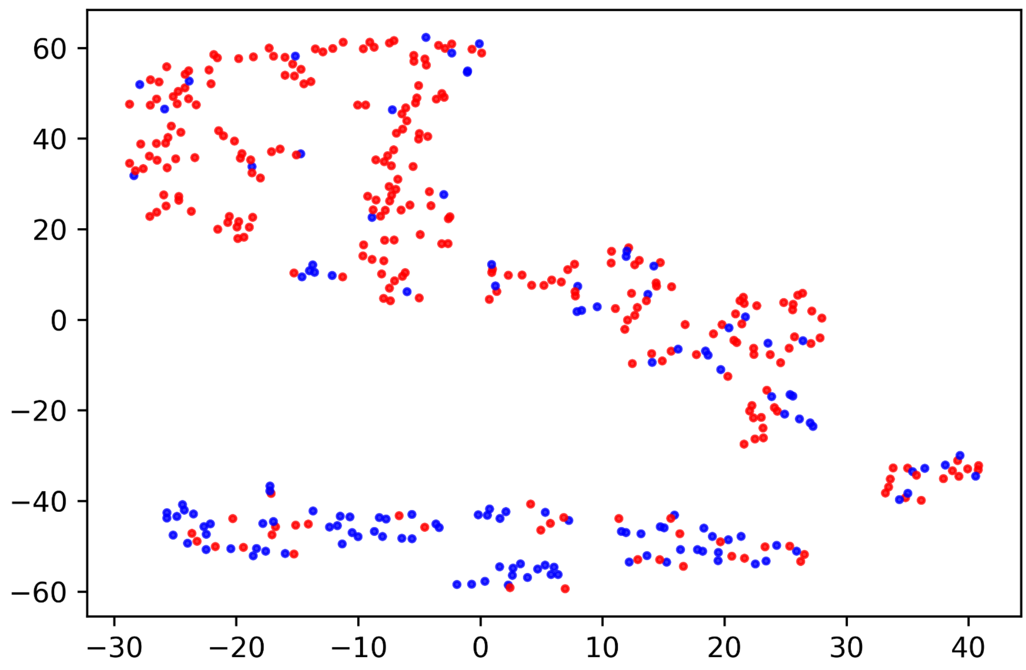


Figure 5. Distribution of active (blue) and inactive (red) molecules in the AID1478 dataset (validation set) obtained by t-SNE dimensionality reduction.

2.3. Interpretation of the GCN atom features for structure-activity relationships

In order to get some intuition about the structural features identified by the GCN as key determinants for Cruzain inhibition, we analyzed the node embeddings from the last graph convolution layer (Fig 6).

Due to the correct mapping between inputs and outputs, the GCN produces a high-dimensional feature space in which inputs from the same output class are closely located.[12] While at the graph-level embeddings we certainly observed a quite clear separation between active and inactive feature vectors (Fig 5), at the node-level embeddings we should not expect such a neat picture, because only a few atoms are responsible for the activity, the remaining nodes conform common scaffoldings present in both active and inactive molecules. Accordingly, a more mixed distribution of active and inactive nodes is observed in Fig 6. Even so, there is still a region enriched in active nodes (circled area), which presumably might contain useful information about the structural requirements for Cruzain inhibition.

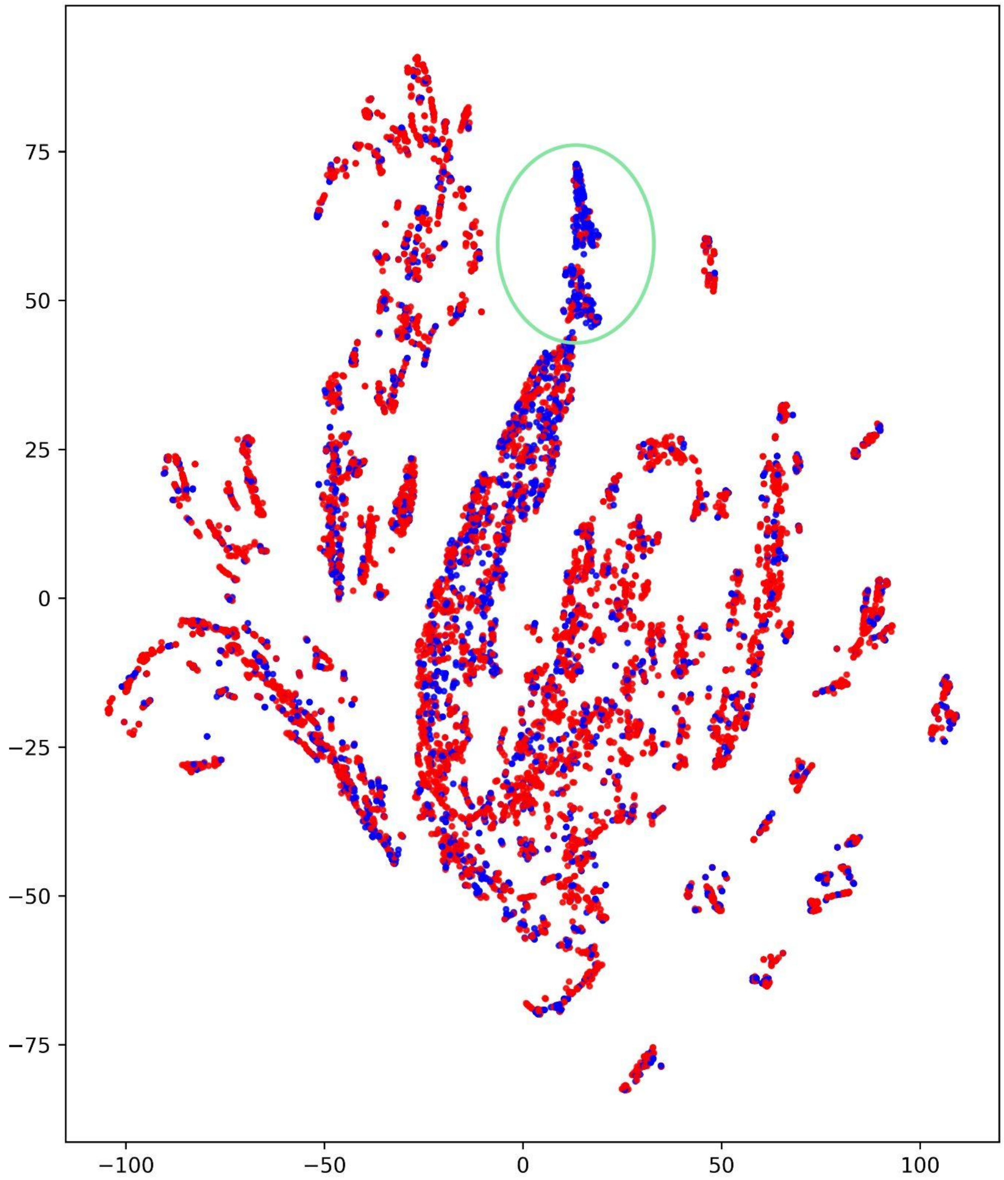


Figure 6. Distribution of nodes from active (blue) and inactive (red) molecules in the AID1478 dataset (validation set) obtained by t-SNE dimensionality reduction. The circled area is enriched in active nodes.

Fig 7 depicts different chemotypes among active compounds in the validation set, including both non-covalent inhibitors as well as electrophilic warheads that react covalently with the nucleophilic sulfur atom from the active-site cysteine residue, Cys 25. Despite the scaffold diversity, the GCN+a+g attempts to find a common set of node features that help distinguish active from inactive molecules. The set of nodes selected by the network as key determinants for activity are highlighted in red over the molecular topologies in Fig 7. These highlighted nodes belong to the region enriched in active nodes in Fig 6 (circled area).

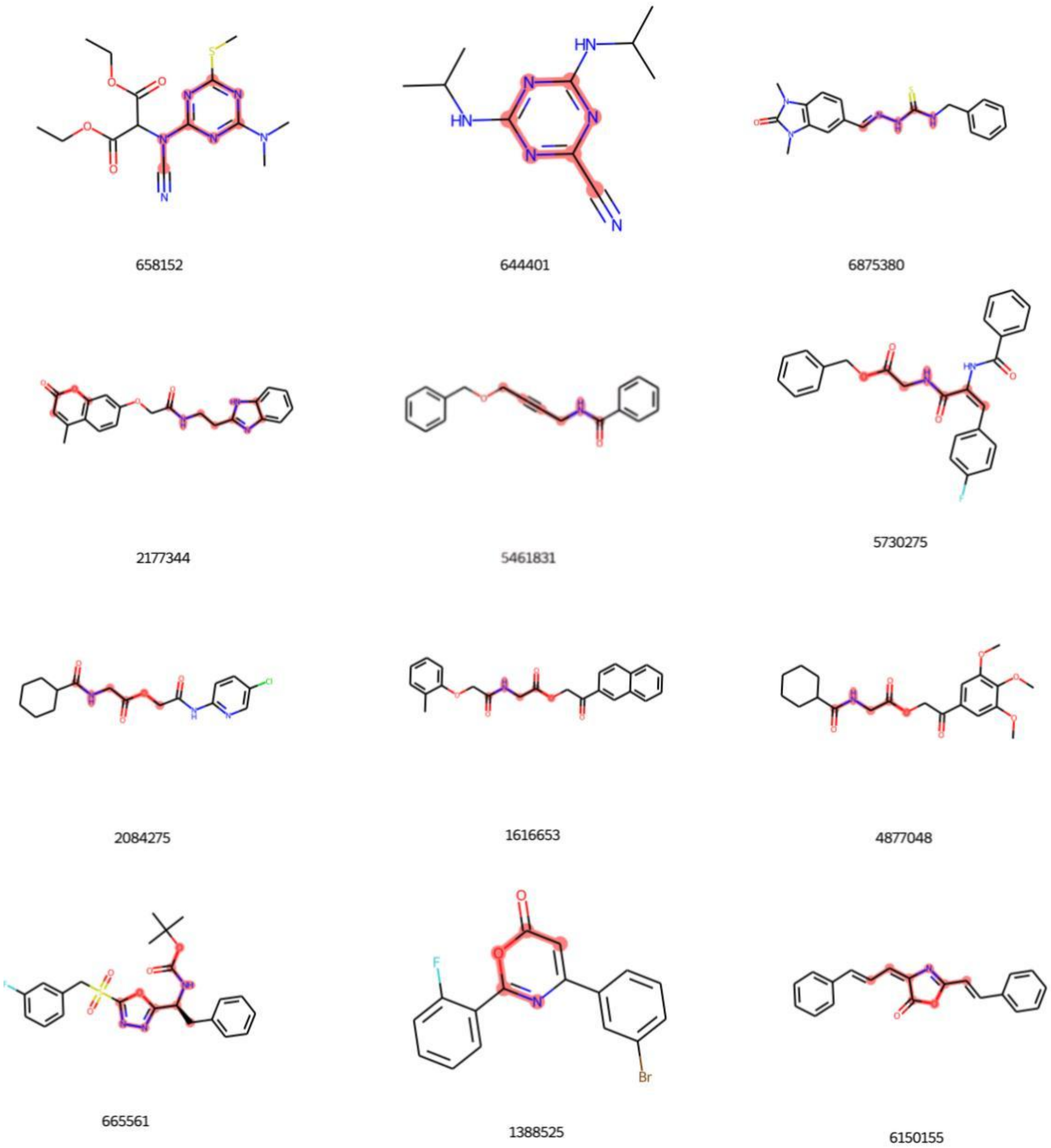


Figure 7. Topologies of representative active molecules in the validation set with their

corresponding PubChem CID numbers. The atoms highlighted in red over the 2D ligand topologies belong to the region enriched in active nodes (circled area in Fig 6).

The network tends to select functional groups containing the amide linkage –N(H)–C=O, or some bioisostere of it, like for instance: carboxylate –O–C=O from esters (i.e. CID 4877048), thioamide –N(H)–C=S and hydrazinylidene –N(H)–N=C from thiosemicarbazones (i.e. CID 6875380 in Fig 7), amidine –N(H)–C=N from benzimidazoles (i.e. CID 2177344) and from triazines (i.e. CID 658162) and –O–C=N from oxadiazoles (i.e. CID 665561).

The network also selects groups with electrophilic properties that can react covalently with the nucleophilic sulfur atom from the active-site cysteine residue, Cys 25, like carbonyl and thiocarbonyl groups as well as alkyne (i.e. CID 5461831) and nitrile (i.e. CID 644401) warheads.

Carbonyl and thiocarbonyl bonds usually form part of electrophilic groups that are already bioisosteric with the peptide bond, i.e. like in thiosemicarbazones and esters. Alkyne and nitrile bonds become bioisosteric with the peptide bond upon addition of the nucleophilic Cys 25 thiol group (see Fig 8 below). The fact that those warheads have to be bioisosteric with the peptide bond suggests that, beyond their inherent reactive properties, they are also important for molecular recognition by the enzyme.

Moreover, lactones like CID 1388525 and CID 6150155 (Fig 7) can be hydrolyzed to the open-chain form which has a peptide-like structure that better fits into the enzyme binding cleft (Fig 8). These types of prodrug-like molecules, similar to those that contain reactive warheads that become bioisosteric with the peptide bond after the nucleophilic addition (i.e. triazine nitriles, alkynes, among others) are often not handled properly by molecular docking because the enzyme-bound form of the molecule differs from that in the compounds database and molecular docking only considers noncovalent complementarity. On the other hand, the GCN+a+g is able to detect the key structural determinants for activity either in the prodrug form or in the final, covalently-bound form of the compounds. This could explain in part the better performance of the GCN+a+g with respect to docking.

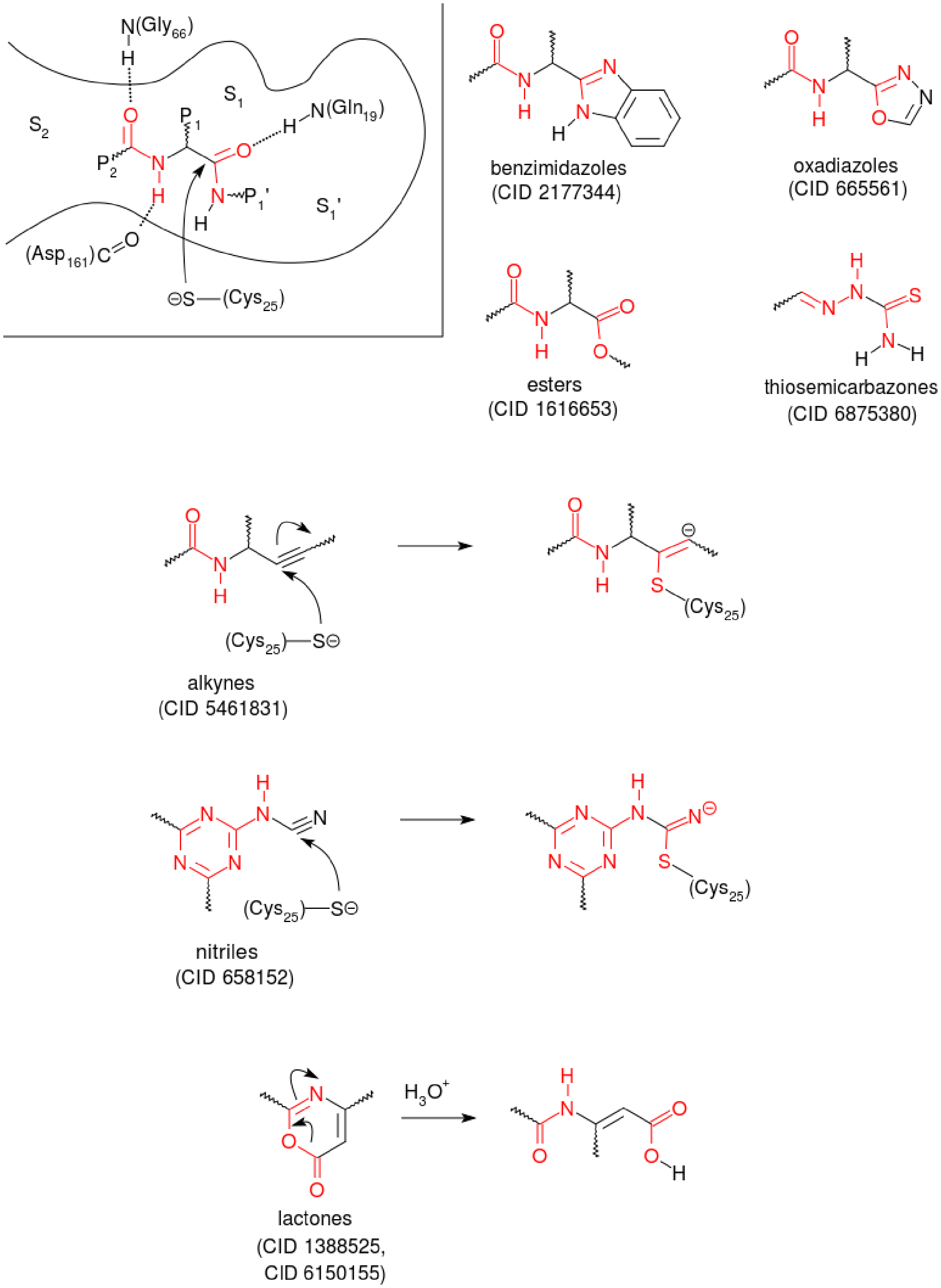


Figure 8. Peptide bonds and the bioisosteres of the peptide bond are highlighted as substructures, on representative chemotypes of Cruzain inhibitors. For each chemotype, an

example molecule from Fig 7 is indicated with the CID number. Top-left inset shows the protein residues within the enzyme binding cleft that interact with the substrate peptide linkages.

Fig 8 depicts the structure of common chemotypes in the validation set. Most of these chemotypes are well known inhibitors of Cruzain.[19] Peptide bonds and the bioisosteres of the peptide bond are highlighted in red. Usually two or more of those peptide-like groups are highlighted by the network within each structure (see Fig 7). These groups together are intended to mimic the amide linkages from the enzyme peptide substrate, as shown in Fig 8 inset (top-left). Therefore, the network emphasizes the importance of preserving the substrate-like structure of compounds for displaying activity against Cruzain.

Top-left inset in Fig 8 also shows the protein residues within the enzyme that interact with the substrate peptide linkages. The leftmost peptide bond fits into the narrowest part of the enzyme binding cleft, between the S1 and S2 sub-pockets, and provides strong anchoring through hydrogen bonds (H-bonds) with the backbone of Asp161 and Gly66. On the other hand, the rightmost peptide bond, namely the one that is cleaved by the enzyme, forms H-bonds with the oxyanion hole residue Gln19. That interaction is intended to stabilize the negative charge on the carbonyl oxygen of the substrate peptide bond, upon addition of the nucleophilic sulfur atom.[20]

Therefore, for the substrate-like chemotypes in fig 8, one should expect similar binding modes and similar interactions than those observed for the enzyme substrate peptide. However, visual inspection of docking poses from the previous virtual screening experiment reveals that in many cases the docking algorithm fails to predict the correct interactions for those chemotypes.

2.4 GCN-guided molecular docking

We can use the features learned by the GCN+a+g to tell the docking algorithm which interactions it should pay more attention to, in order to improve docking performance.

The pharmacophoric restraint functionality of rDock allows the user to introduce pharmacophoric points to reward ligand poses that match the desired characteristics or, otherwise, to penalize those that do not.

Three mandatory pharmacophoric restraints were included in the docking algorithm : a) two H-bond acceptors (Acc) placed at position coordinates to form H-bond with side-chain of Gln19 and backbone of Gly66 and b) one H-bond donor (Don) that H-bond with backbone of Asp161 (see Fig 9).

Fig 9 shows docking poses of an active compound from the AID 1478 dataset (CID 751269) before and after applying the pharmacophoric restraints. Upon applying the pharmacophoric restraints the compound atoms are rearranged to match the pharmacophoric points, depicted as spheres in Fig 9.

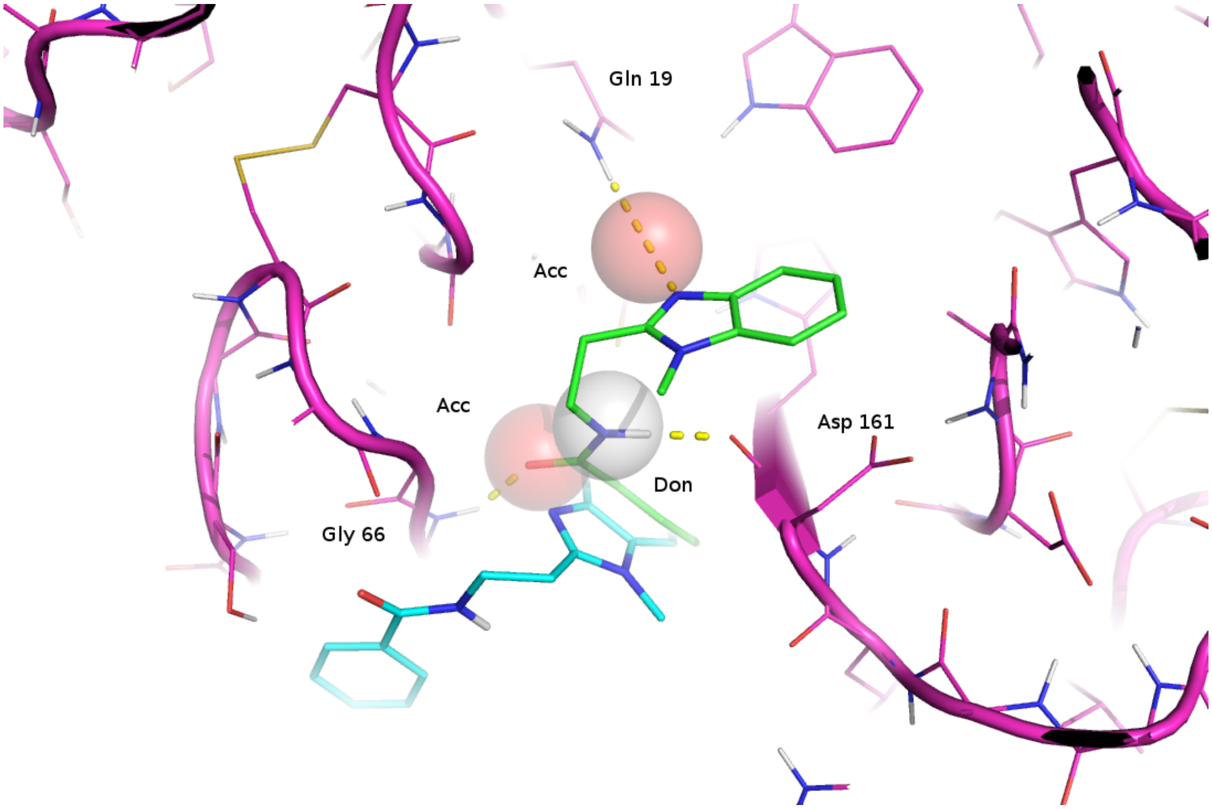


Figure 9. Binding pose of an active compound (CID 751269) from the similarity-filtered AID 1478 dataset obtained by free (cyan) and restrained (green) docking. The pharmacophoric restraints, depicted as translucent red (H-bond acceptor) and white (H-bond donor) spheres,

guide the compound for adopting a substrate-like conformation within the enzyme binding cleft.

Pharmacophoric restraints are incorporated into the docking algorithm in the form of distance penalties that add a positive (i.e. unstabilizing) term to the scoring function. The penalty for each restraint is based on the distance from the nearest matching ligand atom to the pharmacophore restraint center. Therefore, compounds that bear the structural characteristics highlighted by the GCN+a+g, i.e. that can form the key substrate-like interactions (depicted in the inset of fig 8), will be scored higher while ligands that lack those features will rank lower.

As can be seen in Fig 10, the performance of the guided docking has improved (AUC= 0.69) as compared to unbiased docking (AUC~ 0.63, Fig 2).

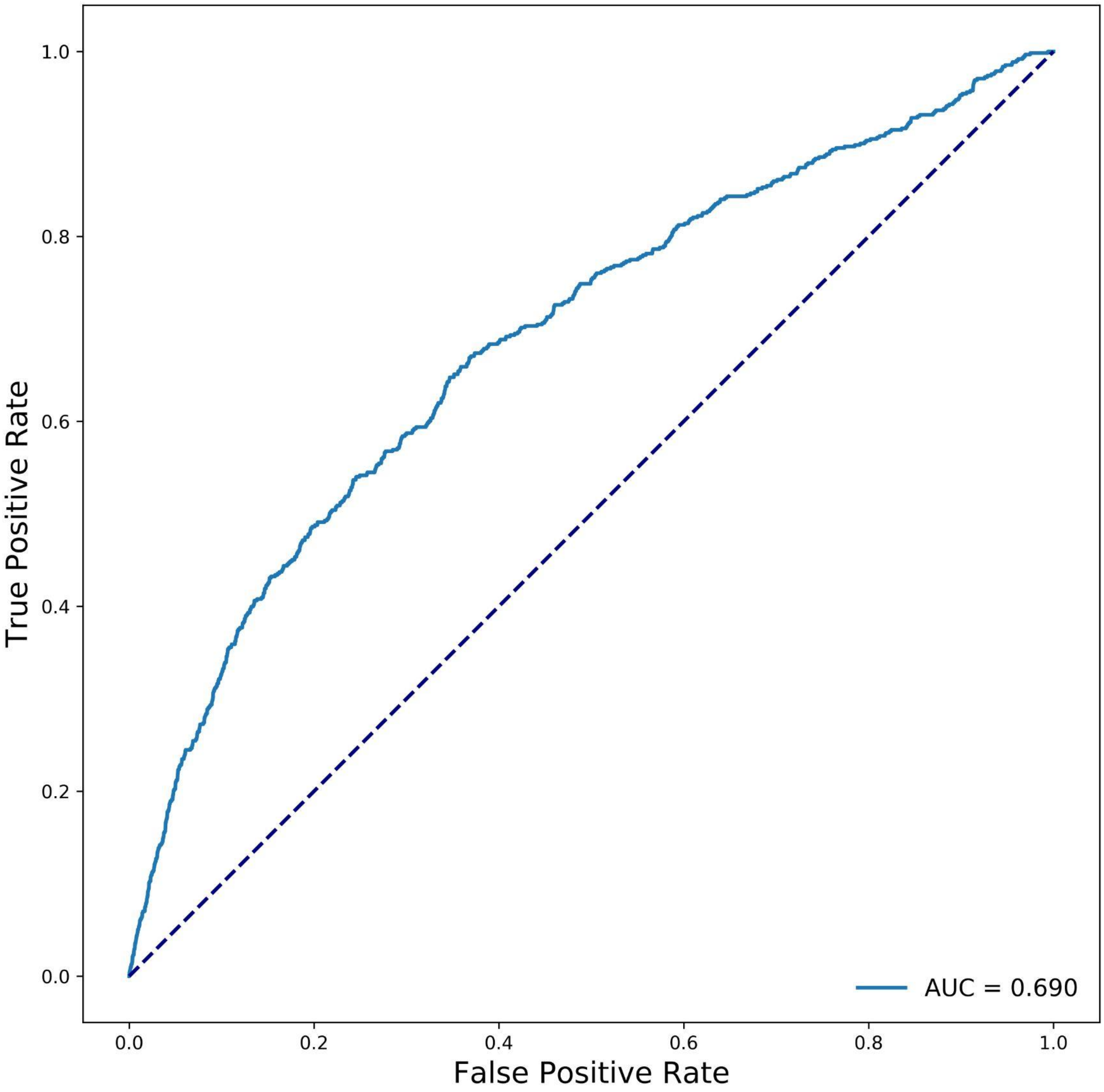


Figure 10. Docking of the similarity-filtered AID 1478 dataset guided by pharmacophore restraints that account for the relevant features learned by the GCN+a+g.

2.5 GCN as a pre-docking filter

Besides guiding the docking with the molecular features learned by the GCN, we could directly exploit the GCN predictive power to boost the docking performance.

In prospective virtual screening campaigns, the compounds from an unknown database are ranked according to the docking scores, and a certain percentage of the top-ranked compounds are prioritized for experimental testing.

The trained GCN, on the other hand, could not be used directly to rank compounds in the same way as docking, because it was not trained with real-valued activity data but with binary data (i.e. active or inactive). Nevertheless, due to its high performance for compound classification, the GCN could be used as a pre-docking filter, i.e. to filter out inactive molecules from the dataset. In this way, the docking algorithm would be fed with a dataset enriched in active compounds, which likely would raise the hit rate in prospective virtual screening campaigns.

The orange ROC curve in Fig 11 shows that the combination strategy of GCN followed by molecular docking, outperforms the standard docking of the similarity-filtered AID 1478 dataset. Compounds used to train the GCN were discarded in advance from the dataset to avoid biasing the performance assessment.

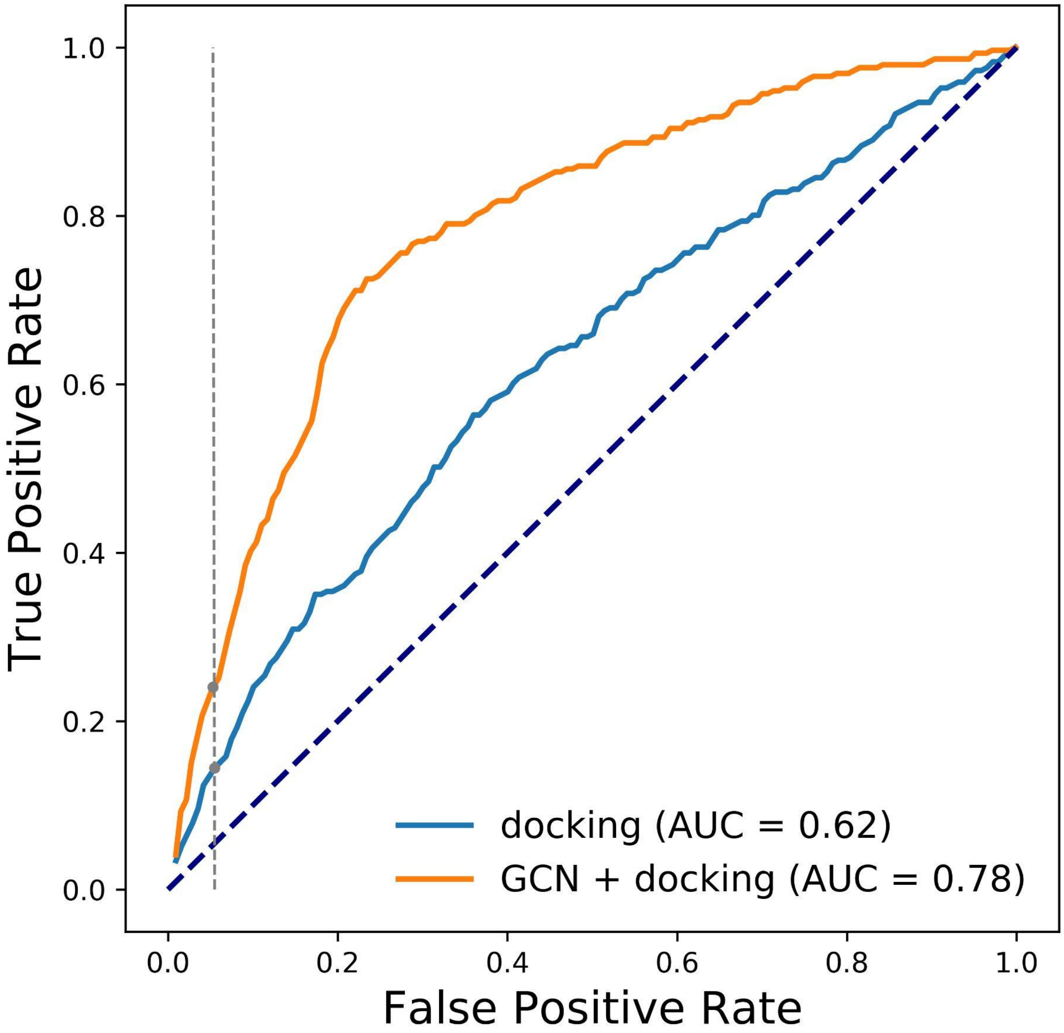


Figure 11. ROC Curve and Area Under the Curve (AUC) achieved by applying a combination strategy of GCN + docking, on the similarity-filtered AID 1478 dataset. Performance of the standard docking procedure is also depicted for comparison. Intersection of vertical dashed line with ROC curves represents the True Positive Rate (TPR) achieved by the top-ranked 5% of the database.

As evidenced by the vertical dashed line in Fig 11, if the same percentage of top-ranked compounds were prioritized for testing from both procedures, a quite higher hit rate will be achieved by the combined approach of GCN + docking, as compared to the standard docking procedure.

2.6 Covalent inhibitors

Despite the improvement in results obtained by applying GCN+a+g as a docking pre-filter, a large part of the active molecules are still poorly ranked by the docking algorithm. In the previous combination experiment, the docking calculations were run directly on the compounds that passed the GCN filter, without any intervention in the docking algorithm. Docking performance could be further improved by applying pharmacophoric restraints to the filtered compounds, but this may still be insufficient to prioritize covalently bonded and prodrug-like molecules.

Those chemotypes are inevitably missed by the docking algorithms that rank the compounds based only on shape and charge complementarity. On the other hand, compound prioritization based only on GCN outcome is unfeasible as already discussed above. A possible workaround might be to perform a similarity-based clustering of the compounds predicted by the GCN as actives and then select one or more representative scaffolds from each cluster for follow up, paying special attention to those chemotypes with reactive scaffoldings.

3. Computational details

Fig 12 depicts the overall procedure for virtual screening of the AID 1478 dataset with the GCN and molecular docking. For assessing the performance of the GCN, the dataset has to be partitioned in (at least) train and test sets, and only the last one should be used for model evaluation. As for molecular docking, in principle the entire dataset can be used for performance evaluation, since docking algorithms rank the compounds with a built-in (pre-trained) scoring function.

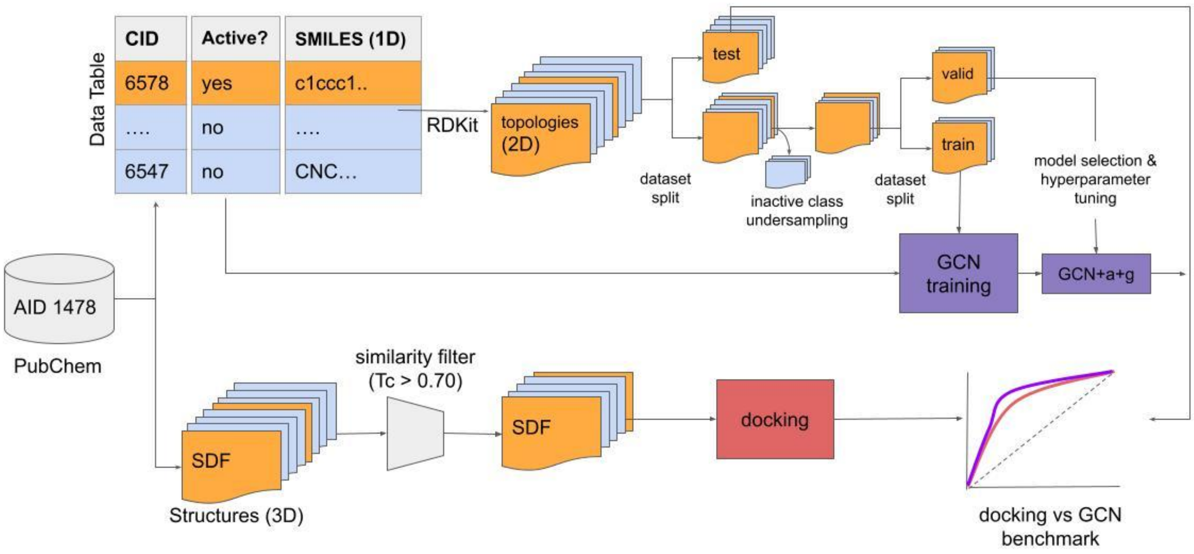


Figure 12. Workflow for the Virtual Screening of the AID 1478 dataset by the GCN and molecular docking

3.1 AID 1478 dataset

The PubChem AID 1478 dataset is composed of 197,846 substances that were assayed on a quantitative High Throughput Screening (qHTS) experiment against Cruzain (https://pubchem.ncbi.nlm.nih.gov/bioassay/1478). A data table containing the unique Compound IDentifier (CID), 1-Dimensional string representation (isomeric SMILES) and activity outcome for each compound in the dataset was retrieved from PubChem. Also the 3D structures in SDF format were retrieved from the database (see Fig 12).

After removing duplicates as well as compounds with inconclusive activity, a total of 193,973 compounds remained, out of which only 848 were actual inhibitors of Cruzain.

3.2 Molecular docking

Molecular docking of the compounds was performed at the active site of a Cruzipain structure retrieved from the Protein Data Bank (PDB code 2OZ2) with the docking software rDock.[21] The peptide-like inhibitor covalently bound to the enzyme in that PDB structure was used to define the active region for docking.

Besides the full, pre-processed AID 1478 dataset, a similarity-filtered dataset was constructed by removing inactive compounds very similar to the active ones that might not be correctly distinguished by the docking algorithm. The 1D fingerprints were calculated with openbabel[22] and those inactive compounds with a Tanimoto similarity coefficient Tc > 0.70 with respect to actives, were discarded. Only 77,719 inactive compounds passed the filtering process.

Docking poses were ranked by their scores as calculated with the standard rDock scoring function. The ability of the algorithm for prioritizing active ligands among top-ranked compounds was evaluated at different scoring thresholds by computing the ROC curve and measuring the Area Under the Curve (AUC).

3.3. Dataset preparation for GCN model building

The 2D molecular representations to be fed into the GCN were built from their corresponding 1D isomeric SMILES (Fig 12), with the help of the cheminformatics python toolkit, RDKit (https://www.rdkit.org).

An issue with the AID 1478 dataset is the severe class imbalance between active (~800) and inactive (~200K) examples. This might lead the GCN model to entirely ignore the minority class, on which predictions are most important to prioritize the active compounds. To overcome this problem we performed a random undersampling of the majority class to reduce the number of inactive examples until a class distribution of 1 active every 2 inactives (i.e. 1:2 ratio) was achieved. It is important to note that the change to the class distribution was only applied to the training dataset. The undersampling was not applied to the test set used to evaluate the performance of the model (Fig 12).

The dataset was split into two sets, for model training and testing (80/20 split), respectively. A stratified split was applied in order to preserve the same proportions of examples in each class as observed in the original dataset. After performing the undersampling on the dataset reserved for training, it was further split into train and validation sets (80/20 split), and the performance of the model was monitored during training on the validation set (Fig 12). All these data preparation steps were performed with the machine learning library for python, scikit-learn.[23]

3.4 Graph Convolutional Network

3.4.1 Graph representation of molecules

Atoms and bonds in a molecule can be represented by nodes and edges of a graph G = (A, X), where: (i) X is an N × F input feature matrix, N is the number of nodes and F is the number of input features for each node and (ii) A is an N × N matrix representation of the graph structure such as the adjacency matrix A of G, that contains the connectivity of the atoms in the molecule.[24]

Figure 13a shows the graph representation of an example molecule, n-propylamine. Adjacency matrix A as well as atom input features, including atom type, number of hydrogens attached, number of bonds (i.e. atom degree) and aromaticity were computed with the cheminformatic python module RDKit.

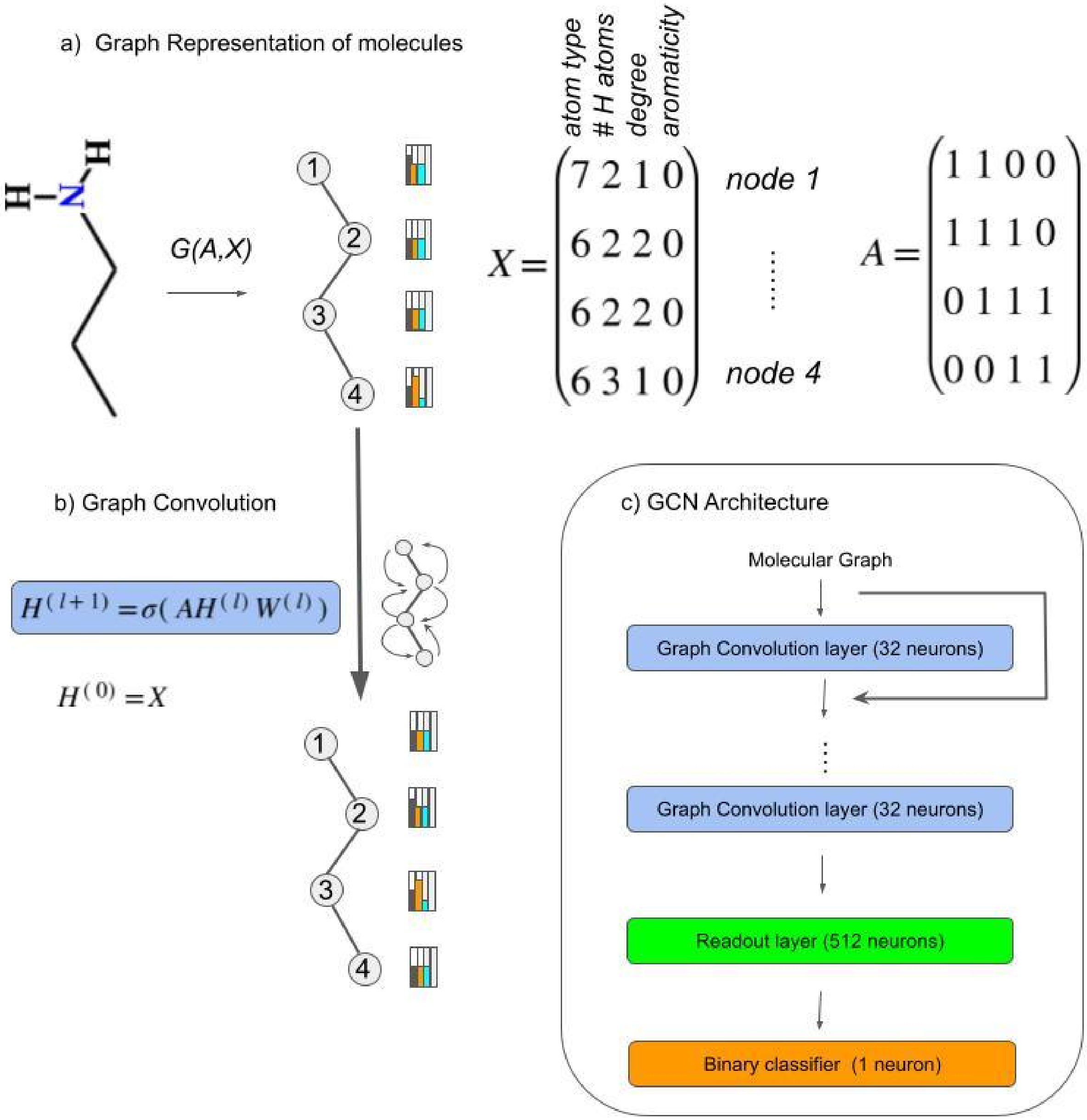

a) Graph Representation of molecules
G(A,X)
1
2
3
4
atom type
# H atoms
degree
aromaticity
X = (7 2 1 0; 6 2 2 0; 6 2 2 0; 6 3 1 0)
node 1
node 4
A = (1 1 0 0; 1 1 1 0; 0 1 1 1; 0 0 1 1)
b) Graph Convolution
$H^{(l+1)} = \sigma(AH^{(l)}W^{(l)})$
$H^{(0)} = X$
c) GCN Architecture
Molecular Graph
Graph Convolution layer (32 neurons)
Graph Convolution layer (32 neurons)
Readout layer (512 neurons)
Binary classifier (1 neuron)

Figure 13. (a) Graph representation of example molecule n-propylamine and (b) subsequent convolution to propagate the atom features through the neighbor nodes. The overall architecture of the vanilla GCN (c) consists of several convolutional layers with skip connections, followed by a readout function that sums up all the nodes for subsequent classification. Details about augmented GCN versions can be found at Ref 12.

3.4.2 Graph convolution

The graph convolution is performed by applying the propagation rule

$$H^{(l+1)} = \sigma\left(A \cdot H^{(l)} \cdot W^{(l)}\right) \quad (1)$$

where σ, $H^{(l)}$ and $W^{(l)}$ denotes the activation function, the feature matrix, and the weight matrix at layer l, respectively (with $H^{(0)}$= X, i.e. the input feature matrix). At each layer, the features $H^{(l)}$ are aggregated to form the next layer's features $H^{(l+1)}$ using the propagation rule σ (Fig 13b).

In this way, the graph convolutional layer represents each node as an aggregate of its neighborhood. In order to reflect atom features at long distances from a specific central atom, a multiple number of graph convolutions have to be applied.

3.4.3 GCN architecture

GCNs can be designed to perform tasks at the node level or full graph level. Graph level classification aims to predict the class label for an entire graph. The end-to-end learning for this task can be realized with a combination of graph convolutional layers, optionally graph pooling layers, and readout layers. Graph convolutional layers are responsible for extracting high-level node representations and the readout layer collapses node representations into a single graph representation. By applying a multi-layer perceptron combined with either a softmax or a linear activation function to the graph representation, we can perform classification or regression on the graph, respectively.[25]

In this work, we implemented a modified version of the vanilla GCN and augmented versions originally developed by Ryu et al.[12] Figure 13c shows the overall architecture of the vanilla GCN implemented in this work.

Regarding the augmented GCNs, they incorporate gate (GCN+g), attention (GCN+a) or both mechanisms simultaneously (GCN+a+g). Although the vanilla GCN incorporates a skip connection to avoid the vanishing gradient problem, it still has problems regulating the best update rate. Therefore, a gated skip connection mechanism (i.e. GCN+g) that finds the optimal update was also considered.[12] Moreover, an attention mechanism (i.e. GCN+a) allows the model to focus on relevant parts of the inputs and achieve a better and accurate prediction.[26]

Since real-valued activity data is only available for the ~ 800 active compounds in the AID 1478 dataset, we were not able to train the GCN for performing quantitative predictions, as the original GCN was intended for. Therefore, instead of a regression problem, the GCN was adapted to perform binary classification based on the compound's activity phenotype (i.e. active or inactive).

Also the hyperparameters (number of graph convolutional layers, batch size, number of epochs, etc) of the network were manually tuned to achieve the best performance possible (Table 1). The loss function monitored during the training epochs was the (binary) cross-entropy, which is a common metric to monitor during classification schemes.

Network adaptation and parameter tuning was performed on a copy of the original Ryu's GitHub repo at https://github.com/SeongokRyu/augmented-GCN. Except for the activation function of the last layer and the loss function, which were modified for a classification task, and the hyperparameter tuning, the overall architecture of the vanilla and augmented GCNs remained the same as in the original implementation of Ryu et al (2018).

Table 1. Hyperparameter setting of the deep learning model. Best values are highlighted in bold. GCN: graph convolutional network. GCN+a: graph convolutional network + attention, GCN+g: graph convolutional network + gate, GCN+a+g: graph convolutional network + attention + gate. In all settings, the optimizer was Adam.[27]

Hyperparameter	Values
Network architectures	GCN, GCN+a, GCN+g, **GCN+a+g**
Convolutional layers	3,4,**5**,6
Learning rate	0.01, **0.001**, 0.0001
Batch size	50, **100**, 150, 200

The pre-processed data set and Jupyter notebook demo with the best hyperparameters are available at https://github.com/lemyp-cadd/gcn-docking.git

4. Conclusions

This work was mostly inspired by the seminal paper by Ferreira et al. entitled “Complementarity Between a Docking and a High-Throughput Screen in Discovering New Cruzain Inhibitors” which dates back to 2010. They undertook a massive parallel docking and HTS screen of 197861 compounds against Cruzain and found that both techniques were complementary to each other and that docking’s weaknesses were orthogonal to those of HTS.

From the perspective of Computer Aided Drug Design, docking programs are not supposed to be used along with HTS, as in the referenced paper, but rather instead of the more expensive HTS. However, if docking alone were used to screen the AID 1478 library, some of the active chemotypes, recovered only by the HTS, would have been missed. Thus, the said “complementarity” between docking and HTS actually was evidencing that docking algorithms were not accurate enough at that time. Neither are they nowadays, even with the continuous advancement in computer software and hardware, the inherent complexity of biological systems challenges any of the currently available molecular modeling methods.

On the other hand, state of the art deep learning (DL) models like the GCNs, are able to capture the complex non-linear relationships between structural and biological data, but they lack the intuitivity of structure-based approaches.

In this work we proposed combination strategies to exploit the benefits of both, namely the ability of GCN models to capture complex relationships from the data and the interpretability of structure-based molecular docking, to virtually screen the AID 1478 library against Cruzain.

By plugging in the atomic embeddings learned by the GCN into the docking algorithm by means of pharmacophoric restraints, docking ability to retrieve the active ligands was enhanced.

Moreover, by applying the GCN as a pre-docking filter, the compound's library was enriched in active molecules and subsequent docking of the filtered library achieved significantly higher hit rates.

Combination strategies involving deep learning and classical molecular docking techniques offer a pragmatic way to circumvent the current technical limitations for modeling complex protein-ligand binding events by structure-based approaches.

ASSOCIATED CONTENT

Supporting Information: additional details of Graph neural networks are provided.

Data and software availability

The chemical structures and the biological activity annotations were obtained from public databases (PubChem: https://pubchem.ncbi.nlm.nih.gov/ and the Protein Data Bank: (www.rcsb.org/). Molecular descriptors and structure similarity measures were computed with the freely available cheminformatic tools RDKit (www.rdkit.org/) and OpenBabel (www.openbabel.org/), respectively. Docking calculations were carried out with open source software rDock (https://rdock.sourceforge.net/). Graph Neural Networks were modeled in Tensorflow (https://www.tensorflow.org/). Dataset preparation as well as model performance evaluation were performed with the machine learning library scikit-learn (https://scikit-learn.org/).

AUTHOR INFORMATION

Corresponding Author

*perunm2014@gmail.com

*emilioluisangelina@hotmail.com

Author Contributions

The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript.

Funding Sources

AML, JLGC, RNV and GAC are current or former fellowship holders from EVC-CIN or CONICET. ERP is a member of the Carrera de Personal de Apoyo (CPA). NMP and ELA are members of the Carrera del Investigador of CONICET (Argentina). This work was partially supported by grants from SECyT-UNNE and AGENCIA I+D+i. Barcelona Supercomputer Center computational resources were accessed through the granted project RES-BCV-2019-3-0015. The Geforce GTX Titan X GPU partially used for this research was donated by NVIDIA Corporation.

ACKNOWLEDGMENT

The authors declare that there are no conflicts of interest.

The authors acknowledge the computer resources at MareNostrum and the technical support provided by Barcelona Supercomputing Center (RES-BCV-2019-3-0015).

ABBREVIATIONS

GCN, Global Convolutional Network; Cz, Cruzain; HTS; High Throughput Screening; CADD, Computer-Aided Drug Design; ML; machine learning; SF, scoring functions; DL, Deep Learning; MLP, Multi-Layer Perceptron; RRN, Recurrent Neural Networks; CNN, Convolutional Neural Networks; qHTS, quantitative High Throughput Screen; PDB, Protein Data Bank; AUC, Area Under the Curve; FPR, False Positive Rate; TPR, True Positive Rate.

REFERENCES

(1) Baldi, A. Computational Approaches for Drug Design and Discovery: An Overview. *Syst. Rev. Pharm.* **2010**, *1 (1)*, 99–105. DOI: 10.4103/0975-8453.59519

(2) Liao, C.; Peach, M. L.; Yao, R.; Nicklaus, M. C. Molecular Docking and Structure-Based Virtual Screening. *In Silico Drug Discov. Des.* In *Future Science Book Series*, 2013; pp 6–20. DOI: 10.4155/EBO.13.181

(3) Deng, N.; Forli, S.; He, P.; Perryman, A.; Wickstrom, L.; Vijayan, R. S. K.; Tiefenbrunn, T.; Stout, D.; Gallicchio, E.; Olson, A. J.; Levy, R. M. Distinguishing Binders from False Positives by Free Energy Calculations: Fragment Screening against the Flap Site of HIV Protease. *J. Phys.*

Chem. B **2015**, *119 (3)*, 976–988. DOI: 10.1021/jp506376z

(4) Ferreira, R. S.; Simeonov, A.; Jadhav, A.; Eidam, O.; Mott, B. T.; Keiser, M. J.; McKerrow, J. H.; Maloney, D. J.; Irwin, J. J.; Shoichet, B. K. Complementarity between a Docking and a High-Throughput Screen in Discovering New Cruzain Inhibitors. *J. Med. Chem.* **2010**, *53 (13)*, 4891–4905. DOI: 10.1021/jm100488w

(5) Li, H.; Sze, K. H.; Lu, G.; Ballester, P. J. Machine-Learning Scoring Functions for Structure-Based Drug Lead Optimization. *WIREs Comput Mol Sci.* **2020**, *10 (5)*, e1465. DOI: 10.1002/wcms.1465

(6) Lim, J.; Ryu, S.; Park, K.; Choe, Y. J.; Ham, J.; Kim, W. Y. Predicting Drug-Target Interaction Using a Novel Graph Neural Network with 3D Structure-Embedded Graph Representation. *J. Chem. Inf. Model.* **2019**, *59 (9)*, 3981–3988. DOI: 10.1021/acs.jcim.9b00387

(7) Na, G. S.; Chang, H.; Kim, H. W. Machine-Guided Representation for Accurate Graph-Based Molecular Machine Learning. *Phys. Chem. Chem. Phys.* **2020**, *22 (33)*, 18526–18535. DOI: 10.1039/d0cp02709j

(8) Wieder, O.; Kohlbacher, S.; Kuenemann, M.; Garon, A.; Ducrot, P.; Seidel, T.; Langer, T. A Compact Review of Molecular Property Prediction with Graph Neural Networks. *Drug Discov. Today Technol.* **2020**, *37*, 1–12. DOI: 10.1016/j.ddtec.2020.11.009

(9) Korolev, V.; Mitrofanov, A.; Korotcov, A.; Tkachenko, V. Graph Convolutional Neural Networks as “General-Purpose” Property Predictors: The Universality and Limits of Applicability. *J. Chem. Inf. Model.* **2020**, *60 (1)*, 22–28. DOI: 10.1021/acs.jcim.9b00587

(10) Mercado, R.; Rastemo, T.; Lindelof, E.; Klambauer, G.; Engkvist, O.; Chen, H.; Bjerrum, E. J. Graph Networks for Molecular Design. *Mach. Learn. Sci. Technol.* **2021**, *2 (2)*. DOI: 10.1088/2632-2153/abcf91

(11) Coley, C. W.; Jin, W.; Rogers, L.; Jamison, T. F.; Jaakkola, T. S.; Green, W. H.; Barzilay, R.; Jensen, K. F. A Graph-Convolutional Neural Network Model for the Prediction of Chemical Reactivity. *Chem. Sci.* **2019**, *10 (2)*, 370–377. DOI: 10.1039/c8sc04228d

(12) Ryu, S.; Lim, J.; Hong, S. H.; Kim, W. Y. Deeply Learning Molecular Structure-Property Relationships Using Attention- and Gate-Augmented Graph Convolutional Network. arXiv:1805.10988 (2018). DOI: 10.48550/arxiv.1805.10988

(13) Sakai, M.; Nagayasu, K.; Shibui, N.; Andoh, C.; Takayama, K.; Shirakawa, H.; Kaneko, S. Prediction of Pharmacological Activities from Chemical Structures with Graph Convolutional Neural Networks. *Sci. Rep.* **2021**, *11 (1)*, 525. DOI: 10.1038/s41598-020-80113-7

(14) Torng, W.; Altman, R. B. Graph Convolutional Neural Networks for Predicting Drug-Target Interactions. *J. Chem. Inf. Model.* **2019**, *59 (10)*, 4131–4149. DOI: 10.1021/acs.jcim.9b00628

(15) Fout, A.; Byrd, J.; Shariat, B.; Ben-Hur, A. Protein Interface Prediction Using Graph Convolutional Networks. In *Advances in Neural Information Processing Systems, Proceedings of the 31st Conference on Neural Information Processing Systems 2017 (NIPS 2017)*, Long Beach, CA, Dec 4-9 2017; Guyon, I., Von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds, Curran Associates Inc. Publishers: Red Hook, NY, 2018; Vol 30, pp. 6531–6540.
https://proceedings.neurips.cc/paper/2017/file/f507783927f2ec2737ba40afbd17efb5-Paper.pdf

(16) Mysinger, M. M.; Carchia, M.; Irwin, J. J.; Shoichet, B. K. Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. *J. Med. Chem.* **2012**, *55 (14)*, 6582–6594. https://doi.org/10.1021/jm300687e.

(17) Goodfellow, I.; Bengio, Y.; Courville, A. Deep Learning. In *Adaptive Computation and Machine Learning series*, Diettrich, T. Ed; Massachusett Institute of Technology, 2016.

(18) van der Maaten, L.; Geoffrey E. H. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579-2605.

(19) da Silva, E.B.; do Nascimento Pereira, G.A.; Ferreira, R.S. Trypanosomal Cysteine Peptidases: Target Validation and Drug Design Strategies. In *Comprehensive Analysis of Parasite Biology: From Metabolism to Drug Discovery,* 2016, pp 121–145. DOI: 10.1002/9783527694082.ch5

(20) Turk, D.; Gunčar, G.; Podobnik, M.; Turk, B. Revised Definition of Substrate Binding Sites of Papain-like Cysteine Proteases. *Biol. Chem.* **1998**, *379 (2)*, 137–147. DOI: 10.1515/bchm.1998.379.2.137

(21) Ruiz-Carmona, S.; Alvarez-Garcia, D.; Foloppe, N.; Garmendia-Doval, A. B.; Juhos, S.; Schmidtke, P.; Barril, X.; Hubbard, R. E.; Morley, S. D. RDock: A Fast, Versatile and Open Source Program for Docking Ligands to Proteins and Nucleic Acids. *PLoS Comput. Biol.* **2014**, *10 (4)*, 1–8. DOI: 10.1371/journal.pcbi.1003571

(22) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An Open Chemical Toolbox. *J. Cheminform.* **2011**, *3,* 33. DOI: 10.1186/1758-2946-3-33

(23) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V. Scikit-learn: Machine learning in Python. *J Mach Learn Res*, **2011**, *12*, 2825-2830.

(24) Kipf, T. N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. arXiv:1609.02907, 2017. DOI:10.48550/arXiv.1609.02907

(25) Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P.S. A Comprehensive Survey on Graph

Neural Networks, arXiv:1901.00596, 2019. DOI: 10.48550/arXiv.1901.00596

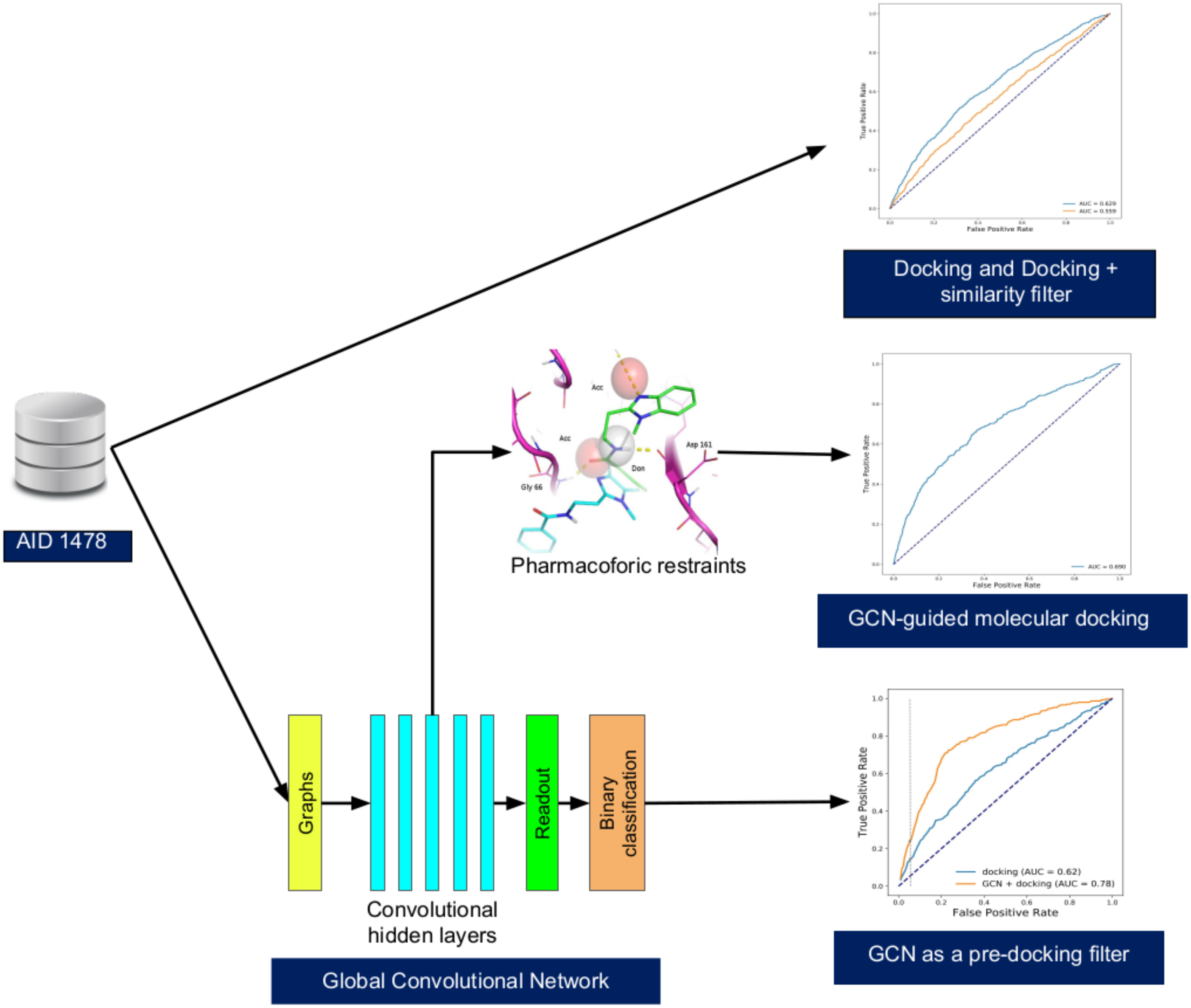
(26) Zhaoping X.; Dingyan W.; Xiaohong L.; Feisheng Z.; Xiaozhe W.; Xutong L.;, Zhaojun L.; Xiaomin L.; Kaixian C.; Hualiang J.; Mingyue Z. Pushing the Boundaries of Molecular Representation for Drug Discovery with the Graph Attention Mechanism. *J. Med. Chem.*, **2020**, *63* (16), 8749-8760. DOI: 10.1021/acs.jmedchem.9b00959

(27) Kingma, D. P.; Ba, J. L. Adam. A Method for Stochastic Optimization, arXiv:1412.6980, 2014. DOI: https://doi.org/10.48550/arXiv.1412.6980

For Table of Contents Use Only

Graph neural networks and molecular docking as two complementary approaches for virtual screening: a case study on Cruzain

Adriano M. Luchi, J. Leonardo Gomez Chavez, Roxana N. Villafañe, German A. Conti, E. Rafael Perez, Emilio L. Angelina*, Nelida M. Peruchena*


AID 1478
Docking and Docking + similarity filter
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
AUC = 0.629
AUC = 0.559
Acc
Acc
Don
Gly 66
Asp 161
Pharmacoforic restraints
GCN-guided molecular docking
AUC = 0.690
Graphs
Convolutional hidden layers
Readout
Binary classification
Global Convolutional Network
docking (AUC = 0.62)
GCN + docking (AUC = 0.78)
GCN as a pre-docking filter