Brendan - A Deep Convolutional Network for Representing Latent Features of Protein-Ligand Binding Poses

Thomas Lau, Ron Dror
Department of Computer Science
Stanford University
thomklau@stanford.edu

Abstract

Empirical molecular "fingerprints" are often used in computational drug discovery (specifically in QSAR methods [3]) to predict Protein-Ligand binding affinity. However, these fingerprints are based on rigid, geometrically-based chemical descriptors that must be hand-tailored to match quantum mechanical experiment data, making the development and choice of fingerprint features extremely difficult. In this paper, we introduce a deep convolutional network, Brendan, that allows us to learn the latent features of Protein-Ligand binding poses by learning from the validated crystallographic poses of PDBBind [14]. Although other approaches have attempted to use deep learning to predict Ki/Kd, toxcity, or potency of molecules (see [18] [1] [21] [10]), we are the first to explore the chemical intuition behind these models and present a chemically inspired deep learning framework that can accurately predict $-\log(K_i/K_d)$*. The main contributions of this paper are to: (1) explore the effect of using a Protein-Ligand centric language (through SPLIF voxels [5]) to represent our 3D crystallographic structures, (2) develop novel graph convolutional methods for crystallographic data, using the theoretic with of [6], and (3) show that the latent features learned (via the fully connected representation) can be used in other Protein-Ligand downstream regression/classification applications.*

1. Introduction

Nine of the top ten most prescribed medicines in the United States are small molecules [4]. Although "hot" new methods such as gene editing occupy the majority of articles about therapeutics, the reality is that small molecules have and will continue to make a significant impact on human health. Small molecule (ligand) based therapeutics work by binding to and changing the behavior of proteins of interest (proteins that are involved in disease-related pathways). Drug discovery is highly dependent on the prediction of protein-ligand binding affinity and function. In most cases, in both academia and industry, this prediction is done manually by a team of highly specialized medicinal chemists. However, the chemical space of synthesizable ligand-like small molecules intractable ($> 10^{60}$ compounds) [19]. QSAR (Quantitative structure-activity relationship models) methods, introduced in the early 2000s, comprised of the first efforts to automate the drug discovery process. These methods all create a fixed-length feature vector to describe the molecular properties of the ligand of interest. However, by "hashing" molecular features into a fixed-length feature vector, we lose all type of spatial relativity from our input structure. Although these methods have decent performance on ligand inputs, they perform

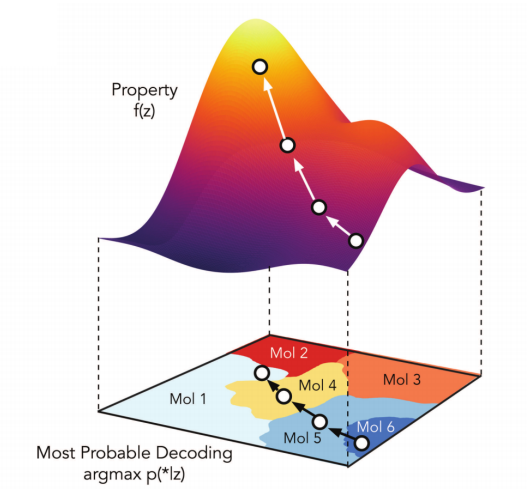


Figure 1. The chemical latent space learned by a deep convolutional network will allow us to use gradients to optimize functions that pertain to binding properties of a ligand-protein complex. Small perturbations in the binding latent space would provide molecules that have similar binding properties. Reproduced from [11].

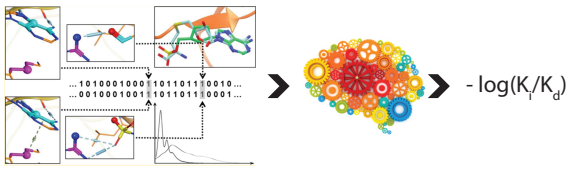


Figure 2. QSAR methods use traditional machine learning methods, such as random forests, to regress on fixed chemical feature vectors [3].

quite poorly on protein-ligand structures due to the importance of spatially dependent protein-ligand contacts. Depending on the contacts made, ligands can induce different downstream biological effects in protein pathways.

Physics based methods such as protein-ligand docking make highly inaccurate assumptions about both the interactions being formed and the conformation of the protein-ligand system but are still the most popular method for protein-ligand prediction[8]. Very recent methods ([21] [10]) have attempted to tackle the open problem of protein-ligand binding affinity via deep learning models. Although these new methods are promising, they significantly overfit - failing to generalize to protein classes not trained on. Predicting protein-ligand binding affinity is still an open problem, with the majority of models in this space being unexplored. In this paper, we explore 3 different types of novel deep learning models for protein-ligand binding poses: (1) SPLIF Voxels and 3D Convolutions, (2) Interaction Voxels and 3D Convolutions, (3) Interaction Graphs and Graph Convolutions. These different deep learning architectures are all used to regress on $-\log(K_i/K_d) \in \mathbb{R}$. We evaluate each of these methods' R^2 for predicting $-\log(K_i/K_d)$ for a held out test set of 1500 structures from PDBBind and explore the chemical intuition behind each model.

2. Related Work

2.1. QSAR Approaches

Popular QSAR fingerprints include ECFP and SPLIF [3] [5]. At every atom, these fingerprint methods create a circular radius, increasing the radius up to 5 Angstroms, hashing the set of atoms contained in this radius into a fixed length binary array. ECFP is the generalized version of this procedure, SPLIF is specific to motifs found between the protein and ligand within the binding site. The typical length of these bit vectors is $2^3 - 2^8$. Bit vectors are most commonly fed into a traditional machine learning algorithm, such as a random forest, in order to make classification or regression results. Similarity between molecular fingerprints is often computed using the Tanimoto coefficient, also known as the Jaccard index, between two bit vectors [2]. The intuition behind this similarity metric is that molecules with similar motifs should have the same chemical properties. Since QSAR fingerprints

More traditional QSAR approaches use fingerprint vectors that contain chemical descriptors of molecules, such as atomic weight, valence, partial charge, formal charge, hybridization, etc. These descriptors don't generalize well to large protein-ligand complexes due to their chemical complexity [3].

2.2. Physics Based Methods

Docking procedures provide physics based functions to approximate protein-ligand binding affinity. Often, docking is used to create ligand binding poses for a particular protein when the binding pose is not known. Docking programs such as Glide and Vina first perform a search of a predefined binding pocket in the protein. Then, potential poses are filtered such that no highly energetically unfavorable clashes between heavy atoms occur. Finally, the binding affinity of the remaining subset of poses are predicted using a physics based force field. Recent studies have shown that docking based methods often include the correct ligand pose in the final subset of results [17]. The weakness of docking methods lies in the crude, approximate force field that it uses to evaluate the final subset of poses. This is done in order to make the scoring set computationally tractable. More advanced polarizable or QM-based force fields are significantly more expensive [16]. With an accurate prediction of protein-ligand binding affinity (as provided by our method), the results of docking could be improved significantly. Recent work has attempted to do exactly this, but with suboptimal results [15].

2.3. Deep Learning Methods

2.3.1 Voxel Based Methods

Deep learning methods have only recently been applied to chemical problems. The first major paper that used convolutional networks was AtomNet, which uses voxlized 15x15x15 Å voxel volume input with 1x1x1 Å voxel size and classifies protein-ligand complexes as high or low binders [20]. However, AtomNet is proprietary software by AtomWise and it is extremely unclear what kind of voxel featurization AtomWise used. AtomNet is trained on the ChEMBL dataset, which contains 78k actives, 2M decoys, and 290 targets. However, ChEMBL does not contain crystallographic poses for active or decoy data. It has been speculated that AtomWise simply chose the highest ranking docking pose in order to obtain their voxel input volumes, but they refuse to answer any specific questions about their method. Although the results of AtomNet did not outperform previous methods, the paper represented a new interest in deep learning models for medicinal chemistry.

Wu et al. introduce a similar model, MoleculeNet. 3D crystallographic data from PDBBind used as training

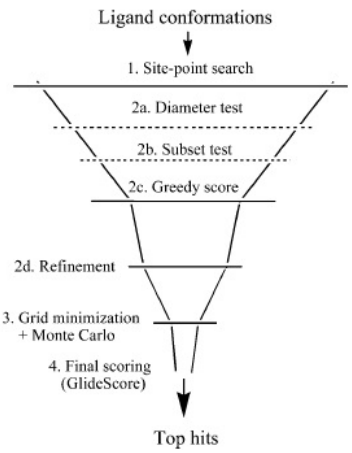


Figure 3. Docking methods first use geometrical-based filters to remove ligand poses with highly unfavorable heavy atom clashes. These geometrical filters are lightweight to compute and allow a large amount of the binding site to be explored by the ligand. Further refinement steps are taken, each being more computationally intensive and accurate. Finally, physics based force fields are applied to the refined set of ligand poses (≈ 300) to predict binding affinity. [8]

data. Although there is significantly less data in PDBBind (30000) compared to ChEMBL (78k actives, 2M decoys), PDBBind is crystallographic and can be seen as ground-truth. This aliviates the docking-specific bias that results from using ChEMBL data. MoleculeNet uses SPLIF, salt bridge, and H bond terms within each voxel. Therefore, MoleculeNet simply contains a language for chemical fragments within each voxel, without a distinction between protein and ligand. Pi-Pi Cation and Pi-Pi Stacking terms were originally included in each voxel but were removed since they reduced performance. Since we are trying to learn binding features, it makes sense to instead use a language that is centered around ligand-protein interaction. From Zhang et al., we can see intuitively that we have to choose a language that is represented by our bit vector that will let our network visually learn from examples (see Figure 10).

MoleculeNet expanded on AtomNet - adding many more features to each voxel, including partial charge, atomic mass, and ECPF [21]. However, MoleculeNet is still outperformed by methods such as Random Forest on the PDBBind dataset. MoleculeNet also uses a 15x15x15 Angstrom input with 1x1x1 Angstrom voxels to featurize the binding site of a protein. This is problematic because for one input, there are $2^8 \times 15 \times 15 \times 15 \approx 1M$ parameters. This highly encourages the model to overfit to training data. Additionally, input volumes are not rotated such that the network will maintain rotation invariance.

2.3.2 Graph Convolution Methods

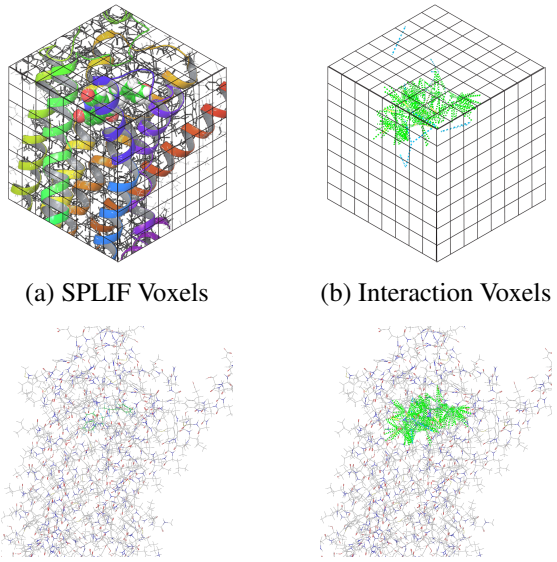
The most notable graph convolution methods relating to molecules are Duvenaud et al. and Kearnes et al. [7] [13]. Both methods introduce similar fully-differentiable functions to represent small molecules as finite-sized feature vectors. The main innovation behind these papers is that these "neural fingerprint" methods are connected to a downstream regression loss, allowing the neural fingerprint to be optimized for the regression task of interest.

There has been a notable exception of graph convolutional methods for protein-ligand centric tasks. MoleculeNet implements the method in Duvenaud et al. but achieves subpar results ($R^2 = 0.1894$) [13] [21]. A naive implementation of Duvenaud et al. will not work for protein-ligand prediction because it does not capture the most important factors that drive binding - non-bonded interactions. Since the ligand and protein are two different, non-connected graphs, the method does not know where the ligand is placed in the protein. For all the method knows, the protein and ligand do not bind at all. If we are restricting ourselves to using crystallographic data, it is important for us to take advantage of the high resolution of data granted to us.

Gomes et al. introduces a Atomic Convolution method that is similar to graph convolutional methods. The Atomic Convolution can be seen as a graph convolution where all k neighboring atoms within a neighbourhood of d Angstroms are connected by an undirected edge. Atomic Convolutions achieve state of the art performance on the PDBBind dataset, however, a large amount of information is thrown out by removing bond and interaction information between atoms. Additionally, significant spatial resolution is lost in this method. Atomic Convolutions perform significantly better than graph convolutions though because it still captures some amount of spatial information.

2.3.3 Autoenconder Methods

Iit is very hard to navigate the large chemical space of small molecules since neighboring graph operations can render molecules invalid or unsynthesizable. Bombarelli et al. created an autoencoder, using SMILES strings as input/output to their network [11]. The latent vector that results can be used to optimize any given function, $f(z)$, which is very useful from optimizing chemical functions. However, small perturbations in the latent chemical space are not guaranteed to give new molecules with similar binding properties. Often, although SMILE strings may be very similar, chemically, they can have significantlly diferent properties. The latent feature vector captured by a protein-ligand binding regressor could instead be used, since small perturbation would translate to small perturbations in the binding space. Additionally, the autoencoder in Bombarelli et al.



(a) SPLIF Voxels (b) Interaction Voxels

(c) Duvenaud et al. (d) Interaction Convolutions

Figure 4. A variety of novel methods for predicting protein-ligand binding affinity are explored in this paper.

only works on small molecules that can be represented via SMILE strings (approximately 2D structures). For pratical purposes, a protein-ligand autoencoder would be the most helpful

3. Methods

13,000 crystallographic binding poses from PDBBind are used for all methods in this paper. PDBBind is a set of refined crystallographic structures from the Protein Data Bank (PDB) that contain highly accurate protein-ligand binding affinities [14]. Structures were split into 10000 for training, 1500 for validation, and 1500 for test sets. Structures were split randomly into each set. The same random splits were used for each method.

3.1. SPLIF Voxels

A 21x21x21 Angstrom input cube is drawn around the center of each protein binding site. Protein-Ligand features are inputted into 1x1x1 Angstrom voxels. In each voxel, a 2^8 sized input vector contained SPLIF (Structural Protein-Ligand Interaction Fingerprints) features of protein-ligand motifs [5]. Input grids are transposed across all 3 axises to help the network maintain rotation invariance.

To generate these fingerprint vectors, for the atoms within a given 1x1x1 Angstrom voxel, all protein-ligand motifs within 5 Angstroms of those atoms are uniquely hashed using MD5 into the voxel's bit vector. Note that this means that inputs are very sparse for voxels close to the input edge. SPLIF bit vectors were chosen such that the network could learn the strength of each protein-ligand interaction. Unlike ECFP fingerprints, since SPLIF is protein-ligand based, each bit maintains the spatial interaction between a protein and ligand [5]. With ECFP fingerprints, the network does not know what is ligand and what is protein - which can be problematic with low 1 Angstrom spatial resolution. Spatial resolution could be increased, however, with 1x1x1 Angstrom voxels and 2^8 bit vectors, maximum batch sizes of 10 could be fit onto a GTX 1080 before running out of memory. Smaller voxels would also require many more network parameters and encourage overfitting. The average value of each index in a voxel's bit vector is 1e-4 with the maximum number of hash collisions being 5 - indicating that there is enough information for the network to learn about specific protein-ligand motifs.

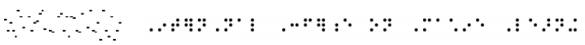
After data preparation, inputs are fed into a ResNet-style network containing 3D Convolutional layers to predict $-\log(K_i/K_d)$ (see Figure 5). A variety of network sizes with different regularization strengths were trained - 50 layers, no dropout, no regularization; 25 layers, 20 percent dropout, L2 regularization of 1e-2 on fully connected layers, 15 layers, 30 percent dropout, L2 regularization of 3e-2 on fully connected layers. All networks were trained using the Adam optimizer with L2 loss and a learning rate of 1e-3. L2 is an appropriate loss function for this regression because changes in $-log(K_i/K_d)$ linearly affect binding affinity. Despite this linear relationship, L2 is chosen over L1 due to its favorable dynamics during training. A batch size of 10 was used, model was trained for 100 epochs.

3.2. Interaction Voxels

The interaction voxel method also places a 21x21x21 Angstrom input cube around the protein's binding site and divides space into 1x1x1 Angstrom voxels. Schrodinger's Maestro API is used to identify protein-ligand interactions that are known to play a role in binding. The OPLSv3 force field is then used to get the kJ/Mol interaction energy associated with a specific protein-ligand interaction [12]. Protein-Ligand interaction energies between atom sets is quantified by the sum of electrostatic and Lennard Jones potentials. OPLSv3 has been quantitatively shown to be one of the best force fields for calculating energies between general protein-ligand systems [12]. Since we are not performing any temporal simulations, force field bias is not considered.

Once protein-ligand interactions are identified, the midpoint of the interaction is determined and the interaction energy is added to the voxel's feature vector. Each feature vector for a voxel contains energies for (1) Hydrogen Bonds, (2) Pi Stacking, (3) Hydrophobic contacts, and (4) Salt Bridges.

Interaction Voxels were trained using a 15 layer ResNet, 30 percent dropout, L2 regularization of 3e-2 on fully connected layers. A small network was used due to the condensed information of interaction energies. In a sense, we



(a) Binary (b) Braille

Figure 5. Zhang et al. speculate that their convolutional network was able to understand sentiment from text due to the bit vector's similarity to braille. For Brendan to learn about protein-ligand binding features, we need to choose a chemical language that centered around the task of protein-ligand interactions.

used physics-based force fields to perform lower-level feature detection. Therefore, fewer layers should be needed. The network was trained using the Adam optimizer with L2 loss and a learning rate of 1e-3

3.3. Graph Convolutions

For this method, protein-ligand binding poses are encoded in a graph structure. In all graph convolutional methods explored (and in past literature), atoms are represented as nodes in the graph and edges represent a molecular interaction (both bonded and non-bonded). Each node in the graph contains a graph signal $s \in \mathbb{R}^h$ where h is the dimension of the signal.

The graph signals at each node are a concatenation of:

- One hot vector of atom element (Length 44)
- One hot vector of atom degree (Length 11)
- One hot vector of number of hydrogens attached (implicit hydrogen model) (Length 5)
- Implicit atom valence
- One hot vector of atom Hybridization Type (Length 5)
- Formal atom charge
- Number of radical electrons
- Boolean isAromatic
- Atomic Mass

The goal of a Graph Convolution Network (of which there are many varieties) is to learn a function of graph signals on a graph $G = (V, E)$ with feature signal matrix $F \in \mathbb{R}^{N \times D}$ and adjacency matrix (weighted or unweighted) $A \in \mathbb{R}^{N \times N}$.

The main failing of previous graph convolutional methods on protein-ligand data has been the failing to include important non-bonded interactions that are relavant to binding. Interactions between protein-ligand and protein-protein atoms are search for 20 Angstroms around the center of the protein binding site. These unbonded interactions are included in our graph structure.

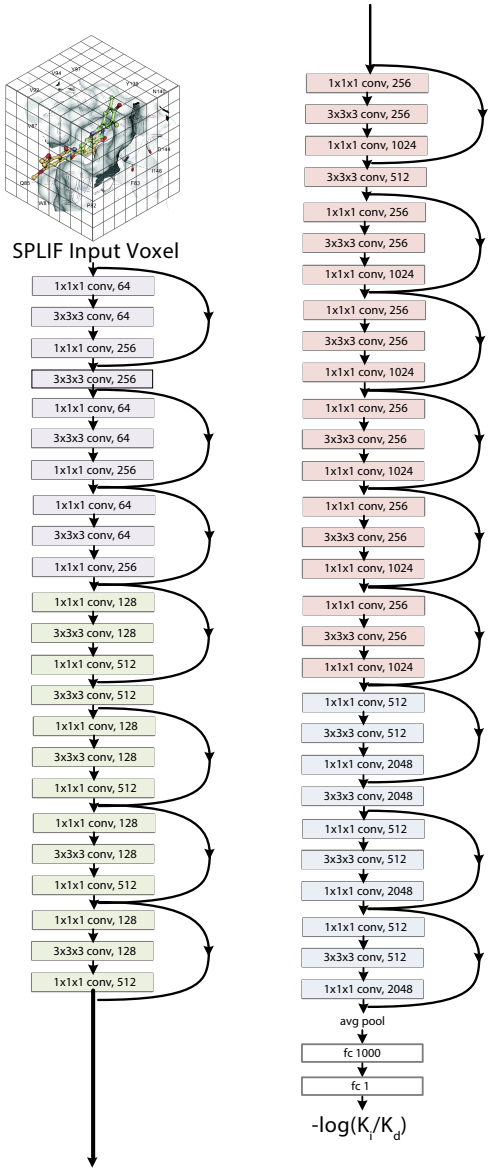
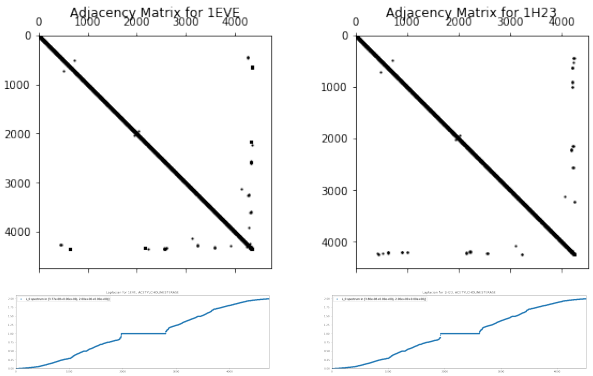


Figure 6. Network architecture for Brendan. Radial arrows signify a residual connection between layers. Each convolutional layer is a RxRxR 3D convolution in which each filter has 2^8 weights (the size of the SPLIF bit vector). Brendan is trained using Atom with L2 loss with a batch size of 10.

The results of Duvenaud et al., "Neural Graph Fingerprints", are explored via the DeepChem library [21]. Neural fingerprints are created by summing R hidden graph layers. Each graph layer involves a pooling operation where the signals at each node are convolved with neighboring nodes. This convolution is formalized as $\sigma(\tilde{A}H^{(l)}W^{(l)})$ where σ is a non-linear activation function such as ReLU, $H^(l)$ is the signal at time l for a given node (with $H^{(0)} = X$), and $W^{(l)}$ is the learnable weight matrix for the l^{th} hidden graph



(a) Structure of 1EVE (b) Structure of 1H23

Figure 7. Adjacency matrix and graph of eigenvalues of the Laplacian matrix for structures 1EVE and 1H23, both acetylcholinesterase receptors.

convolutional layer. Notice that at each time step, previous node signals are stored and updated via the convolutional update rule $\sigma(\tilde{A}H^{(l)}W^{(l)})$ in which new signals are computed and used for future updates. $\tilde{A}$ is actually the symmetric normalization of the adjacency matrix. The full update rule is $\sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)})$. A fully connected layer is then used to learn a desired property, given the learned fingerprint vector. We explore how adding unbonded edges affects the naive version of this procedure.

Defferrard et al. is also tested on this graph representation [7]. This method uses Chebyshev polynomial approximations to learn convolutional filters for the k nearest neighbors for a node in the graph. This method also uses a graph pooling layer in which efficient graph coarsening is applied to approximate the structure of the Laplacian. When nodes signals are pooled, they are simply added into the coarsened node. The coarsened graph is represented as a binary tree for efficient traversal with orphaned nodes containing two "ghost" node children to maintaing the binary tree assumption. After futher graph pooling and convolutions, the resulting graph signals are fed into a fully connected layer to regress on a specific output, in this case binding affinity. Due to the high computational cost of this method, only atoms around the binding site are included in our graph. Additionally, since this method relies on a fixed number of graph nodes and graph structure, we always select 100 atoms around the binding site, starting from a counter-clockwise atom numbering. Since the Laplacian for each graph is different, we expand on this method by recomputing it for each graph. As a sanity check, we can see that the Laplacian for proteins in the same family share extremely similar structures (see Figure 7).

PDBBind: Regression on -log(K_i/K_d); R^2 Performance			
Methodology	Train	Valid	Test
ECFP	0.373	0.361	0.337
ECFP Grid	0.960	0.488	0.471
SPLIF Grid	0.971	0.501	0.497
Interaction Grid	0.915	0.402	0.348
Naive Graph Convolution	0.193	0.196	0.189
Atom Convolution [21]	0.962	-	0.562
Brendan Graph Convolution	0.916	0.567	0.503

Figure 8. Performance metrics for different models that were regressed on PDBBind

4. Dataset

All methods are trained on the PDBBind general set. The general set contains 13,000 entries and are split into 10,000 for training, 1,500 for validation, and 1,500 for testing. The PDBBind general set was chosen over the refined (4000 entries) and core (290 entries) sets because of the larger amount of data available. Additionally, structures are minimized using OPLSv3 with heavy atom constraints, so the resolution of the crystallographic structures is not of major concern. To further mitigate the problem of resolution, missing hydrogens are added via PDBFixer.

PLIP and Maestro are used to detect non-bonded terms in the binding site. OPLSv3 is used to verify the energies of these non-bonded interactions for interaction voxels. RDKit and DeepChem are used to calculate graph signals for each atom.

5. Results and Discussion

5.1. SPLIF Voxels

As expected, using SPLIF Voxels with a large 50 layer ResNet and no regularization highly overfit to the training data R^2:(Train: 0.989, Valid: 0.273, Test: 0.235). Using a deep learning model actually significantly hurt performance compared to using a traditional random forest approach. This is because the last layers of the ResNet contain a significant number of layers (512 and 2048) in certain modules, allowing the network to simply memorize the training data that it saw. In fact, the number of layers actually exceeds the amount of training data fed to the network. Although it looks like the network is learning (see Figure 8), on closer inspection, training loss in final epochs is ¡ 1 while validation loss never goes below 5.

The medium sized network of 25 ResNet layers removes a significant portion of the later, high filter count convolutions. The hypothesis behind this was that with fewer filters, the network wouldn't be able to memorize as much training data and would be forced to generalize. 20 percent dropout and L2 regularization of 1e-2 on fully connected

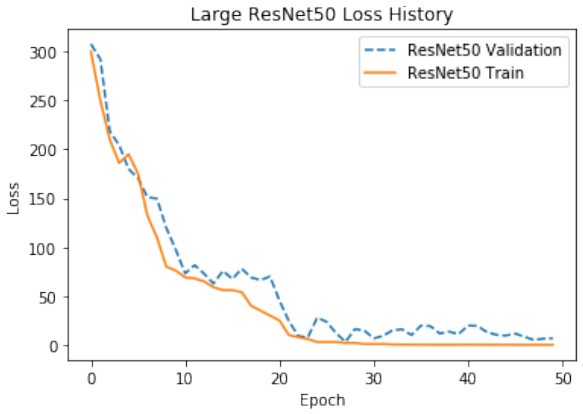


Figure 9. Training and Validation loss for the large ResNet50 model with 3D Convolutions. As shown in the graph, there is a large gap between training and validation loss, which signals us to overfitting.

layers were also used to help generalization. This significantly helped performance and resulted in the best performing model R^2:(Train: 0.971, Valid: 0.501, Test: 0.497). The gap in training and test loss is also seen in this case - training loss is < 1 while validation loss is < 3. Although the average loss for validation and test sets is enough to be chemically useful (poor, medium, strong binder), there is still a large room for improvement.

A small network of 15 layers, 30 percent dropout, L2 regularization of 3e-2 on fully connected layers was trained to see how many layers are needed to keep this generalization. Once again, layers were removed from the end (highest layer number) of the ResNet. A noticeable drop in performance was noticed R^2:(Train: 0.913, Valid: 0.409, Test: 0.431). This seems to indicate that more than 15 layers are needed to fully capture the binding trends that are occurring.

It is extremely unlikely that a voxel based method would perform significantly better without an astronomical increase in data. The network needs to learn the what each of 2^8 bits represent in terms of protein-ligand motifs and learn how the spatial orientation of bits within and close to voxels affect binding energy.

5.2. Interaction Voxels

Interaction energies are input into voxels, as explained in the methods section of this paper. The best performing ResNet25 structure from SPLIF Voxels was used for this regression problem. After 50 epochs, R^2 of (Train: 0.915, Valid: 0.402, Test: 0.348) was received.

Although this method outperforms SPLIF Voxels, it is quite surprising that it still highly overfits on training data since each voxel in the input volume only contains about 5 floats, compared to the 2^8 bits found in SPLIF voxels. This would imply that for Interaction Voxels (also potentially for SPLIF Voxels) that the structure of the input volume does not contain enough information to generalize to other protein-ligand complexes. This would make sense since 1x1x1 Angstrom input vectors are actually quite coarse in terms of spatial resolution. Additionally, interaction energies are added to the midpoint between two interacting atoms. This is a problem because in reality, interactions form edges that connect atoms in a graph. We can clearly see here that voxel-based inputs are not appropriate for protein-ligand data and should be retired from use.

5.3. Graph Convolutions

Duvenaud et al. with addded unbonded edges was performed on PDBBind with R^2:(Train: 0.916, Valid: 0.567, Test: 0.503). Defferrard et al. was also implemented with unbonded edges with R^2:(Train: 0.618, Valid: 0.223, Test: 0.211).

In retrospect, it is not surprising that Defferrard et al. does not perform well on protein-ligand data. Since learned filters are connected to the Laplacian of the input graph, this method assumes that each nodes performs the same function in the graph. This is obviously not the case in terms of a protein-ligand binding site. Our results for Duvenaud et al. are extremely promising and come close to matching the performance of Atomic Convolutions [21]. Very coarsely, connecting each node with its neighbors within a radius of 5A should give similar results to Atomic Convolutions. This leads us to believe that there may be other nonbonded interactions that are not captured by our graph that are accounted for by Atomic Convolutions. More graph edges would lead to faster communication of graph signals between nodes, allowing for more expressive functions based on signals to be learned. Message Passing algorithms for graphs are able to account for different edge types and could be an easy way to connect seeming non interacting atoms such that information can flow easier through the graph [9].

This is the first time that graph convolutional methods have been successfully applied to protein-ligand crystallographic data and represents a new path forward for this class of problems.

5.4. Downstream Machine Learning Applications

To show the potential of protein-ligand specific methods for other machine learning applications, we take the latent features learned by the ResNet25 network and use them to measure binding similarity between poses. As shown by (Lau et al., Unupublished), ligands that are known to bind to certain families of proteins often bind in the same pose. If the latent features of protein-ligand binding are captured, we can use these latent features to measure similarity between poses for different ligands.

Glide docking is first run on 12 ligands that are known to bind to B2AR family of receptors. Markov Chain Monte

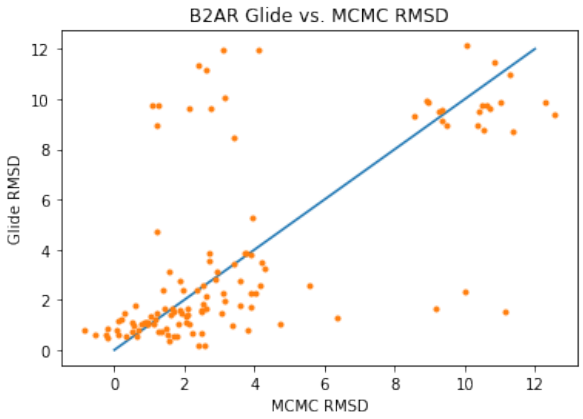


Figure 10. The latent features learned by the ResNet25 model are a good representation of binding features.

Carlo is run for T=100 steps to find the maximum set of ligands that overlap. Overlap between two ligand poses is measured by the L2 distance between latent vectors from the FC layer. This procedure is repeated for all 12 receptors from the B2AR family.

Since we know the correct answers, we can select positive and negative examples from our generated Glide docking list. These positive and negative examples are first fed through the ResNet25 network and the FC vector is given to a random forest. The random forest learns which latent vectors correspond to positive and decoy examples. As shown in figure 10, the FC feature vector captures enough information about the binding pose of the ligand that it can predict binding energy more accurately than Glide.

6. Acknowledgements

Many thanks are given to Scott Hollingsworth who helped refine datasets and introduce the idea of similar ligand binding poses, which motivated the downstream machine learning application. Joe Paggi performed initial experiments about similar ligand binding poses using binary fingerprint vectors, contributing foundational code regarding nonbonded interaction detection that was very useful to make interaction grids. The Pande Lab and DeepChem were amazing resources and much of the featurization code borrows from ideas in their open source library. Graph Convolutional methods used were modified from existing implementations in cnn_graph and DeepChem. Brendan Kelly provided many chemical intuitions about what models may be learning. Ron Dror provided much guidance during the research project.

Disclaimer: The downstream machine learning application that regards similar ligand binding is a seperate project that I have been working on. I had already written much of the code to identify positive/decoy pose examples from Glide and Joe Paggi wrote the random forest code that was used to predict strength of binding. Part of the Brendan project will be presented at the CS191W software fair - however, the focus will be on the similar ligand binding pose problem. For more information about this project, take a look at my notebooks under notebooks_thomas in the supplementary material. The supplementary code attached is a mix of code for the ComBind project (my side project) and the Brendan project (this paper).

References

[1] H. Altae-Tran, B. Ramsundar, A. S. Pappu, and V. Pande. Low data drug discovery with one-shot learning. *ACS Central Science*, 3(4):283–293, 2017.

[2] D. Bajusz, A. Rácz, and K. Héberger. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of Cheminformatics*, 7(1):20, 2015.

[3] A. Cherkasov, E. N. Muratov, D. Fourches, A. Varnek, I. I. Baskin, M. Cronin, J. Dearden, P. Gramatica, Y. C. Martin, R. Todeschini, V. Consonni, V. E. Kuzmin, R. Cramer, R. Benigni, C. Yang, J. Rathman, L. Terfloth, J. Gasteiger, A. Richard, and A. Tropsha. Qsar modeling: Where have you been? where are you going to? *Journal of Medicinal Chemistry*, 57(12):4977–5010, 2014. PMID: 24351051.

[4] Y. Cohen. Small molecules: The silent majority of pharmaceutical pipelines. *xconomy: exome*, 2015.

[5] C. Da and D. Kireev. Structural proteinligand interaction fingerprints (splif) for structure-based virtual screening: Method and benchmark study. *Journal of Chemical Information and Modeling*, 54(9):2555–2561, 2014. PMID: 25116840.

[6] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *CoRR*, abs/1606.09375, 2016.

[7] D. K. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. Convolutional networks on graphs for learning molecular fingerprints. *CoRR*, abs/1509.09292, 2015.

[8] R. A. Friesner, J. L. Banks, R. B. Murphy, T. A. Halgren, J. J. Klicic, D. T. Mainz, M. P. Repasky, E. H. Knoll, M. Shelley, J. K. Perry, D. E. Shaw, P. Francis, and P. S. Shenkin. Glide: a new approach for rapid, accurate docking and scoring. 1. method and assessment of docking accuracy. *Journal of Medicinal Chemistry*, 47(7):1739–1749, 2004. PMID: 15027865.

[9] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. *CoRR*, abs/1704.01212, 2017.

[10] J. Gomes, B. Ramsundar, E. N. Feinberg, and V. S. Pande. Atomic convolutional networks for predicting protein-ligand binding affinity. *CoRR*, abs/1703.10603, 2017.

[11] R. Gómez-Bombarelli, D. K. Duvenaud, J. M. Hernández-Lobato, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *CoRR*, abs/1610.02415, 2016.

[12] E. Harder, W. Damm, J. Maple, C. Wu, M. Reboul, J. Y. Xiang, L. Wang, D. Lupyan, M. K. Dahlgren, J. L. Knight, J. W. Kaus, D. S. Cerutti, G. Krilov, W. L. Jorgensen, R. Abel, and R. A. Friesner. Opls3: A force field providing broad coverage of drug-like small molecules and proteins. *Journal of Chemical Theory and Computation*, 12(1):281–296, 2016. PMID: 26584231.

[13] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, and P. Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of Computer-Aided Molecular Design*, 30:595–608, Aug. 2016.

[14] Z. Liu, M. Su, L. Han, J. Liu, Q. Yang, Y. Li, and R. Wang. Forging the basis for developing proteinligand interaction scoring functions. *Accounts of Chemical Research*, 50(2):302–309, 2017. PMID: 28182403.

[15] J. C. Pereira, E. R. Caffarena, and C. N. dos Santos. Boosting docking-based virtual screening with deep learning. *Journal of Chemical Information and Modeling*, 56(12):2495–2506, 2016. PMID: 28024405.

[16] J. W. Ponder, C. Wu, P. Ren, V. S. Pande, J. D. Chodera, M. J. Schnieders, I. Haque, D. L. Mobley, D. S. Lambrecht, R. A. DiStasio, M. Head-Gordon, G. N. I. Clark, M. E. Johnson, and T. Head-Gordon. Current status of the amoeba polarizable force field. *The Journal of Physical Chemistry B*, 114(8):2549–2564, 2010. PMID: 20136072.

[17] D. Ramírez and J. Caballero. Is it reliable to use common molecular docking methods for comparing the binding affinities of enantiomer pairs for their protein target? *International journal of molecular sciences*, 17(4):525, 2016.

[18] G. Subramanian, B. Ramsundar, V. Pande, and R. A. Denny. Computational modeling of -secretase 1 (bace-1) inhibitors using ligand based approaches. *Journal of Chemical Information and Modeling*, 56(10):1936–1949, 2016. PMID: 27689393.

[19] A. M. Virshup, J. Contreras-Garca, P. Wipf, W. Yang, and D. N. Beratan. Stochastic voyages into uncharted chemical space produce a representative library of all possible drug-like compounds. *Journal of the American Chemical Society*, 135(19):7296–7303, 2013. PMID: 23548177.

[20] I. Wallach, M. Dzamba, and A. Heifets. Atomnet: A deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *CoRR*, abs/1510.02855, 2015.

[21] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. S. Pande. Moleculenet: A benchmark for molecular machine learning. *CoRR*, abs/1703.00564, 2017.