Convolutional Neural Networks for Protein-Ligand Scoring

Matt Ragoza[1,2], Elisa Idrobo[3,6], Joshua Hochuli[2,4], Jocelyn Sunseri[5] and David Koes[5]

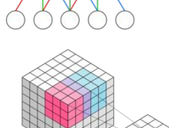
[1]Department of Neuroscience, [2]Department of Computer Science, [3]TECBio REU @ Pitt, [4]Department of Biological Sciences, [5]Dept. of Computational and Systems Biology, University of Pittsburgh, Pittsburgh, PA 15260,
[6]The College of New Jersey, Ewing, NJ 08618

Abstract

Computational approaches to drug discovery reduce the time and cost associated with experimental assays and enable the screening of novel chemotypes. Structure-based drug design methods rely on scoring functions to rank and predict binding affinities and poses. The ever expanding amount of protein-ligand binding and structural data enables deep machine learning techniques for protein-ligand scoring.

We describe a convolutional neural network (CNN) scoring function that takes as input a comprehensive 3D representation of a protein-ligand interaction. A CNN scoring function automatically learns the key features of protein-ligand interactions that determine binding. We train and optimize our CNN scoring functions to discriminate between correct and incorrect binding poses and known binders and nonbinders. We find that our CNN scoring function outperforms the AutoDock Vina scoring function when ranking poses both for pose prediction and virtual screening.

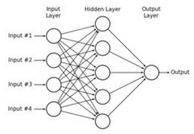
Background



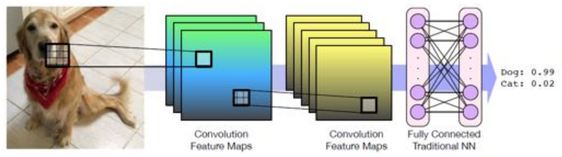
Protein-ligand scoring provides a metric of binding strength between small molecules and target proteins. This has a wide array of uses such as **virtual screening**, which filters large databases of candidate molecules for potential hits, and **docking**, which predicts the binding pose of a ligand.

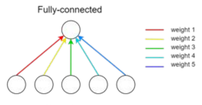
Machine learning strategies have been used to treat scoring as a classification problem between "good" and "bad" binding states, though this often requires manually selecting properties that the model uses for discrimination, for example pairwise interactions and global counts of typical chemical interactions. However, other machine learning models can learn the most important features directly from data.

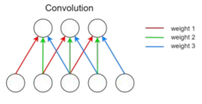
Neural networks are a supervised machine learning algorithm inspired by the nervous system. A basic network consists of an input layer, one or more hidden layers, and an output layer of interconnected nodes. Each hidden node computes a **feature** that is a function of the weighted input it receives from the nodes of the previous layer.



Input data are fed forward through the network, and a prediction is output by the last layer. A neural network is trained by iteratively updating its **weights** by minimization of an **objective function**, for example, the mean squared deviation between predictions and their ground truth labels.







Within the last decade **convolutional neural networks** have become the state-of-the-art in image classification. Convolutional layers only have connection weights to small spatial subsets of the previous layer, and apply these **weight kernels** across the entire input to produce **feature maps**.

The fact that convolutional layers learn local features and apply them across the entire input space leads to faster training and improved accuracy on data with a spatial structure.

The rise of **GPU computing** in combination with other advances has made training networks with many more layers feasible, leading to the surge of research in **deep learning**. Each successive layer in a deep neural network learns features at a higher level of abstraction.

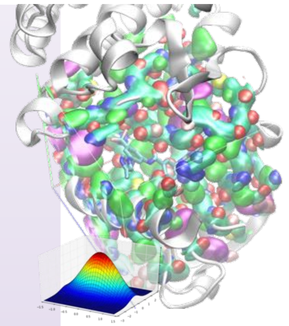
Protein-ligand scoring is a natural generalization of image recognition where the full 3D "images" of protein-ligand complexes are used for training. Convolutional neural nets trained on protein-ligand interactions have the potential to provide substantially more accurate scoring functions for improved docking and virtual screening.

Methods

Datasets

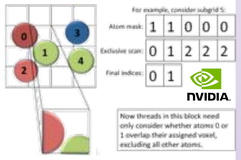
All poses generated with smina/AutoDock Vina
Community Structure-Activity Resource (CSAR)
- CSAR version 2010 with 2011 update
- 337 targets, eliminate weak binders
- Crystal structures → Reference poses
- Poses < 2Å RMSD → actives
- Poses > 4Å RMSD → decoys

Database of Useful Decoys-Enhanced (DUD-E)
- 101 targets
- Active and decoy ligands
- Unknown correct poses



GPU Accelerated Molecular Gridding

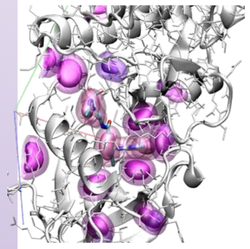
- Parallelize over *atoms* to obtain a mask of atoms that overlap each grid region
- Use exclusive scan to obtain a list of atom indices from the mask
- Parallelize over *grid points*, using reduced atom list to avoid $O(N_{atoms})$ check



Input Format

Voxel grid centered at active site calculated from molecular data
34 atom type channels
- 16 receptor atom types
- 18 ligand atom types

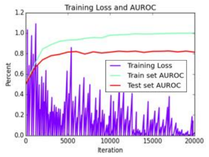
23.5Å^3 Gaussian atom grid
- 0.5Å resolution
- 48^3 points



Training

Caffe Deep Learning Framework
Train to 10,000 iterations
Layer-wise model definition
- N-dimensional input layer
- Convolutional layers
- Non-linear layers (rectified linear units)
- Fully-connected layers
- Softmax (convert to probabilities)
- Multinomial logistic loss (2-class)

Performance
- Mini-batch parallelism (batch size=10)
- Multi-GPU support



Model Evaluation

Receiver operator characteristic (ROC)
- False positive vs. true positive rate
- Area under ROC curve (AUC)

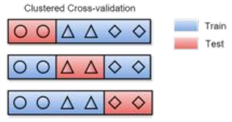
Clustered 3-fold cross-validation
- Split targets into 3 balanced folds
- DUD-E targets with 80% similarity were grouped into same fold
- Train 3 models, leaving one fold out
- Combine performance on test sets
- Avoids testing on targets/ligands similar to training set

Bootstrapped AUC

Optimization

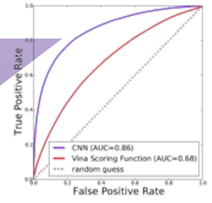
CSAR set
Change single parameter relative to a reference model
- Width of network
- Depth of network
- Resolution of grid
- Rotating, translating, balancing and shuffling during training
- Types of pooling layers
- Numerical vs binary occupancies
- Values of radius multiplier
- Fully connected layer at the end

Evaluate by accuracy and training time
Combine best changes into new reference model and repeat



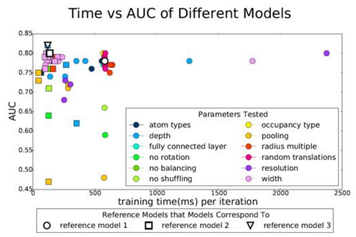
DUD-E Evaluation

Cross-validation was done with the best model
Different ratios of DUD-E to CSAR data were used to train
Multiple poses of ligands vs single poses were tested
- The maximum score of a ligand's poses was taken as the ligand's score

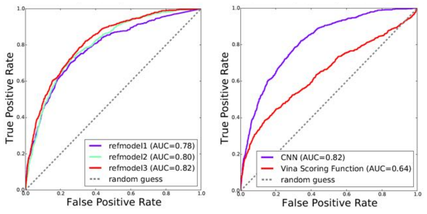


Results

Optimization

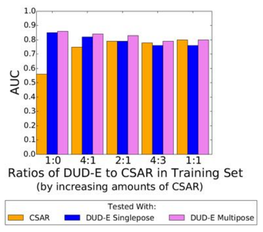


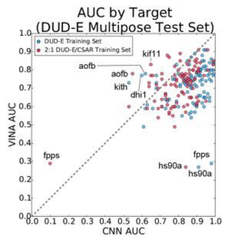
- Each round of optimization increased accuracy and decreased training time
- Rotations, small translations, balancing between actives and decoys and shuffling order during training reduce overfitting to data
- Reducing dimensions lowers training time
- Higher resolution increases accuracy but also significantly increases training time



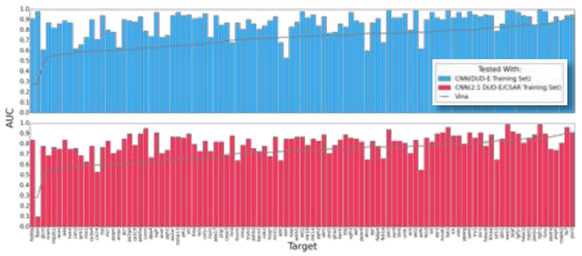
- Optimization increased the AUC of the best model from 0.78 to 0.82
- Best model has a better AUC than Vina's scoring function

Ligand binding prediction with DUD-E



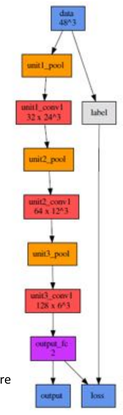


- Training with CSAR and testing on DUD-E or vice-versa is not very effective
- Scoring multiple poses, instead of the pose top-ranked by Vina, and taking the maximum score increases accuracy for virtual screening, suggesting the CNN model can select better poses
  - Using both CSAR and DUD-E to train increases this gain



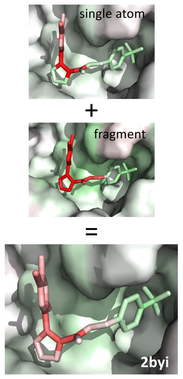
CNN scoring trained with DUD-E has better performance than Vina for 90% of targets

Training with a DUD-E/CSAR 2:1 mix is better for 81% of targets



Visualization

Method



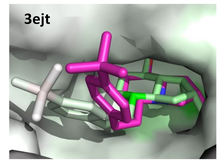
Ligand
- Single atom decomposition
  - Remove atoms one by one and score
  - Compute score difference
  - Set atom with score difference
- Fragment decomposition
  - Fragment ligand with RDKit
  - For each fragment
    - Score ligand without fragment
    - Accumulate score difference on fragment atoms
- Average single atom and fragment scores

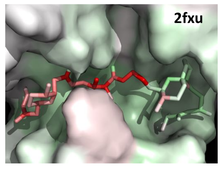
Protein
- Remove whole residues at a time
- Compute score difference
- Set all residue atoms to score difference

Store difference in b-factor field of .pdb file
Visualize with PyMOL

Insights

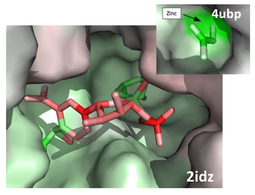
Spatially accurate: When compared with crystal poses, portions of the test poses that are in the same location score well. Atoms far removed from their crystal location often score poorly.

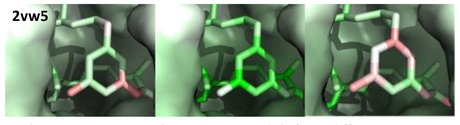




Ligand and receptor agreement: Effects are often shown on both molecules at interacting locations.

Ligand more significant than receptor: Ligands often show more significant negative and positive values than receptors, except in the presence of metal ions, which often dominate the score.





Little consensus on carbon atoms: Models from different training sets disagree on which carbons significantly contribute to overall score.

Future Work

- Explore alternative network topologies, such as residual neural networks
- Evaluate the use of noise models when training
- Investigate the use of CNN scoring for affinity prediction
- More informative visualizations from backpropagated gradients
- Extract positional gradients from neural network to support energy minimization
- Use CNN energy minimization to implement CNN-based *pose generation*
- Use reinforcement learning to iteratively refine CNN models for pose generation
- Deploy an open-source comprehensive CNN-based molecular docking and energy minimization software package (http://github.com/gnina)

Conclusion

We have shown that a convolutional neural network with a well-optimized model and appropriate training dataset has great potential in aiding with drug discovery. Creating variety in the poses through rotation, translation and shuffling during training are important in training the model. Other parameters such as network depth and width can also reduce overfitting. Visualizations highlight the pose sensitivity of the CNN model and can emphasize regions of interest in the protein-ligand complex.

Our best model performs better than Autodock Vina at pose selection when evaluated for pose predication performance (CSAR) and virtual screening performance (DUD-E), although the nature of the training data greatly influences the result.

This work is supported by R01GM108340 from the National Institute of General Medical Sciences. TECBio REU @ Pitt is supported by the National Science Foundation under Grant DBI-1263020 and is co-funded by the Department of Defense in partnership with the NSF REU program.