Convolutional Neural Networks for Protein-Ligand Scoring

Matt Ragoza1,2, Elisa Idrobo3,6, Joshua Hochuli2,4, Jocelyn Sunseri5 and David Koes5

1Department of Neuroscience, 2Department of Computer Science, 3TECbio REV @ Pitt, 4Department of Biological Sciences, 5Dept. of Computational and Systems Biology, University of Pittsburgh, Pittsburgh, PA 15260, 6The 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 function to discriminate between correct and incorrect binding poses and known binders and decoys. 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 maximization of an objective function, for example, the mean squared deviation between predictions and their ground truth labels.

Methods

Datasets

All datasets used in the experiments were processed differently in order to normalize the data and overall count the data.

Training

CNNs are trained using backpropagation of errors through time (BPET). The learning process is broken into epochs, where each epoch is a batch of the training data. The parameters are continuously updated by the optimizer during training.

Optimization

Each round of optimization increased accuracy and decreased training time.

Visualization

Spatially accurate: Compared with the crystal, overlapping portions of the docked pose score well. Non-overlapping score poorly.

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 reduced overfitting to data.
- Higher resolution increases accuracy but also significantly increases training time.
- Model C facilitates substantial amount of more type interaction from atomic (static) score.
- Optimization increased the AUC of the best model from 0.74 to 0.82.

Visualization

Spatially accurate: Compared with the crystal, overlapping portions of the docked pose score well. Non-overlapping score poorly.

Future Work

- Explore alternative network topologies, such as residual neural networks
- Evaluate the use of noise models when training
- Further 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/crns)

Conclusion

We have shown that a convolutional neural network with a well-optimized model and 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 basic model performs better than AutoDock Vina at pose selection when evaluated for pose prediction performance (CSAR) and virtual screening performance (DUD-E), although the nature of the training data greatly influences the result.