GNINA: Deep Learning for Molecular Docking

Paul Francoeur, Matthew Ragoza, Rachel Rosenzweig, Jocelyn Sunseri, David Ryan Koes

Department of Computational and Systems Biology, University of Pittsburgh

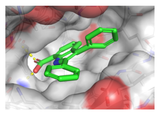
http://github.com/gnina/

Abstract

Molecular docking is an important tool for computational drug discovery that aims to predict the binding pose of a ligand (drug) to a target protein. Identifying a correctly oriented pose requires a scoring function that has a global optimum close to the experimentally observed pose. Additionally, it should be differentiable with respect to atomic positions so that it can be used for gradient-based pose optimization.

We describe a differentiable grid-based convolutional neural network scoring function and explore its application in an end-to-end GPU-optimized molecular docking workflow. We show that convolutional neural networks trained on experimental data can successfully identify correct binding modes and meaningfully rank and score compounds. We describe visualization approaches that map the CNN score back to the atomic inputs to help guide medicinal chemistry optimization and provide insight into the functioning of the neural network. Source code is available under an open-source BSD/GPL license as part of the gnina package.

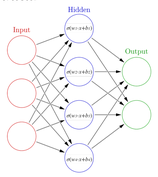
Background



Protein-ligand scoring provides a metric of binding strength between small molecules and target proteins and is a critical subroutine of molecular docking and structure-based drug design. An ideal scoring function would correctly identify accurate ligand poses and predict the binding affinity of the ligand for the protein.

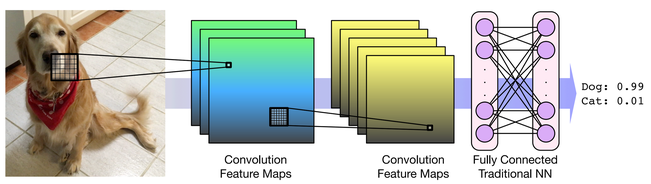
One approach for scoring is to use **machine learning**. Traditionally, this has required manually selecting molecular features, such as pairwise interactions and counts of typical chemical interaction patterns, that are used to train a model. However, other, **non-parametric**, machine learning models can learn the most important features directly from low-level representations of the data.

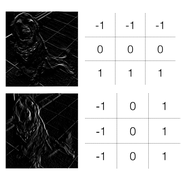
Neural networks are a supervised machine learning algorithm inspired by the nervous system. A basic network consists of an input layer, one or more hidden layers, and an output layer of interconnected nodes. Each hidden node computes a feature that is a non-linear function of the weighted input it receives from the nodes of the previous layer. A neural network with a finite number of nodes can approximate any continuous function to within a given error over a bounded input domain.

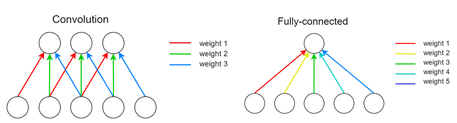


Input data are fed forward through the network, and a prediction is output by the last layer. A neural network is trained by iteratively updating its weights using back-propagation and stochastic gradient descent. Gradients are calculated with respect to a **loss function** such as the mean squared error between predictions and the ground truth labels.

Convolutional neural networks are the state-of-the-art in image recognition. Convolutional layers apply a small non-linear kernel function iteratively across the input to produce a feature map. A function recognizes a local spatial feature of the input. As convolutions are applied to feature maps, higher order features in the input are recognized.

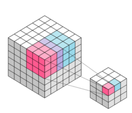






(Left) 2D convolutional kernels for recognizing horizontal and vertical edges. (Above) Convolution layers apply the same weights to subregions of the input. The shared weights reduce the number of parameters in the model.

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.



Datasets

SMINA docked and minimized poses are used for training.

Redocked

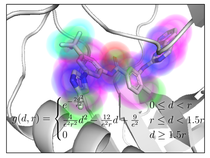
PDBbind refined set
4053 complexes
52,166 ligand poses
affinity data for all ligands

Cross-Docked

Structures from Pocketome
2923 distinct pockets
27,142 receptor structures
4,138,117 non-redundant ligand poses

Data Representation

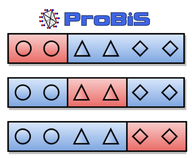
24x24x24Å grid at 0.5Å resolution
14 ligand and 14 receptor atom types
Continuous Gaussian density
CUDA optimized grid generation

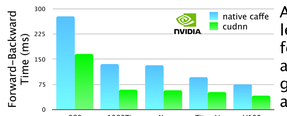


$$g(d,r) = \begin{cases} e^{-2d^2} & 0 \le d < r \\ e^{-2r^2}\left(\frac{16}{e}d^2 - \frac{32}{e}rd + \frac{16}{e}r^2\right) & r \le d < 1.5r \\ 0 & d \ge 1.5r \end{cases}$$

Training

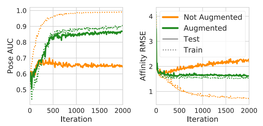
Clustered cross-validation splits datasets to avoid including highly similar examples in both the training set and test set. *Redock* set is clustered using protein sequence and ligand fingerprint similarity. *Cross-dock* set is clustered using ProBis pocket similarity.

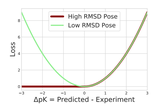




A custom fork of the **Caffe** deep learning framework with support for **cuDNN** accelerated 3D layers and on-the-fly molecular gridding can train 100X faster on a GPU than on a single CPU.

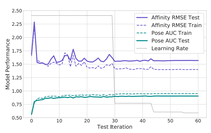
Data augmentation is performed by applying random rotations and translations (±6Å) to protein-ligand complex structures. This reduces overfitting and compensates for the coordinate-frame dependency of a 3D grid representation.



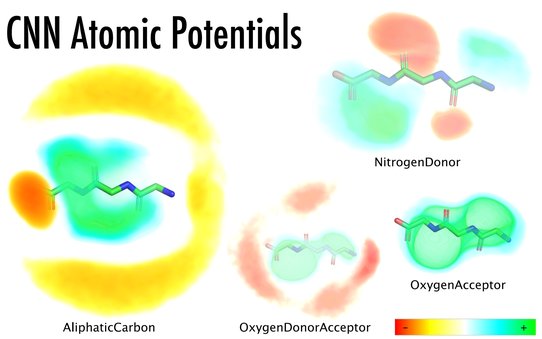


Models are trained to optimize both a **logistic loss** for distinguishing between low RMSD (<2Å) and high RMSD (>4Å) poses (classification) and a mean squared error **L2 loss** for affinity prediction (regression). Low RMSD poses are only penalized for *over* predicting the affinity.

A fixed **learning rate** is dynamically stepped when training performance stops improving. This enables early termination and quick convergence. A test iteration consists of 1000 training iterations with batch size 50.

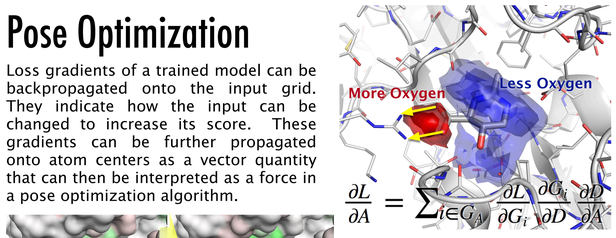


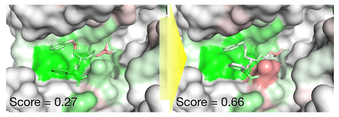
CNN Atomic Potentials



Pose Optimization

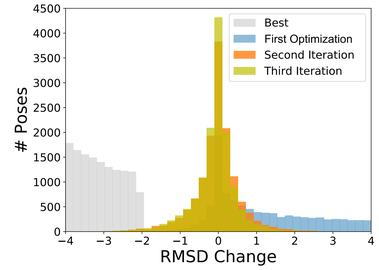
Loss gradients of a trained model can be backpropagated onto the input grid. They indicate how the input can be changed to increase its score. These gradients can be further propagated onto atom centers as a vector quantity that can then be interpreted as a force in a pose optimization algorithm.

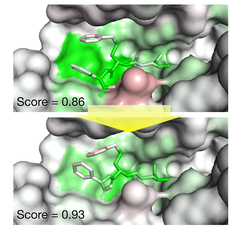




A model trained on docked poses does not learn to properly steric clashes, as such interactions are not present in the training set. This becomes evident when the model is used to optimize a pose.

Iterative Refinement





Adding CNN optimized structures to the training set corrects cases where the model was confidently wrong and improves sensitivity to steric clashes.

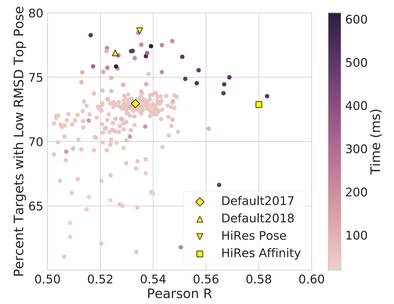
Hyperparameter Optimization

50 parameters for training and the network topology were explored both automatically and rationally

Systematic Each parameter was individually varied

Bayesian Spearmint was used to construct a Gaussian process model of the parameter space and suggest the most informative parameters to evaluate

RF+GA A random forest model of the parameters is trained to predict model performance and a genetic algorithm used to search for novel parameters with the best predicted performance

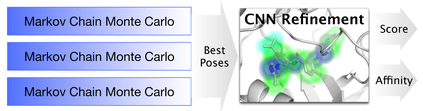


Docking

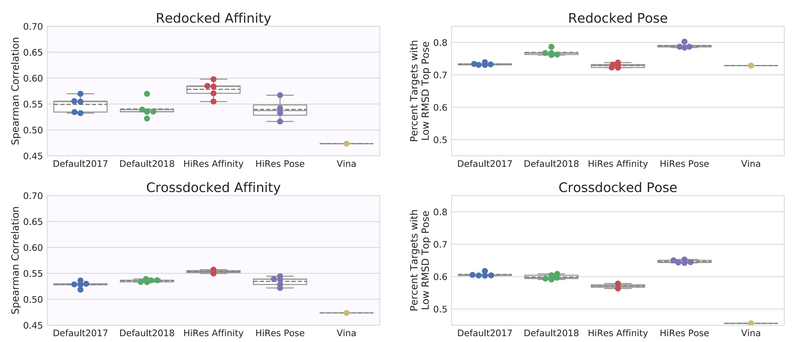
Monte Carlo sampling is performed with grid-accelerated Vina scoring. Improved poses identified during sampling are optimized and ranked using CNN scoring.

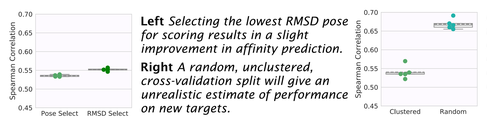
D3R Grand Challenge 3

	cnn docked affinity	cnn rescore affinity	cnn docked scoring	cnn rescore scoring	vina
cat	0.0701	0.154	-0.0351	0.178	**0.179**
p38a	-0.0784	-0.116	-0.329	-0.305	**-0.0631**
vegfr2	0.366	**0.484**	0.434	0.448	0.414
jak2	**0.428**	0.338	0.39	0.27	0.106
jak2 sub3	**0.68**	0.369	-0.372	0.159	-0.633
tie2	0.648	**0.835**	0.136	-0.078	0.561
abl1	0.634	**0.745**	0.005	0.182	0.713



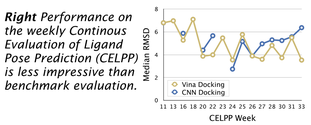
Cross Validation Performance



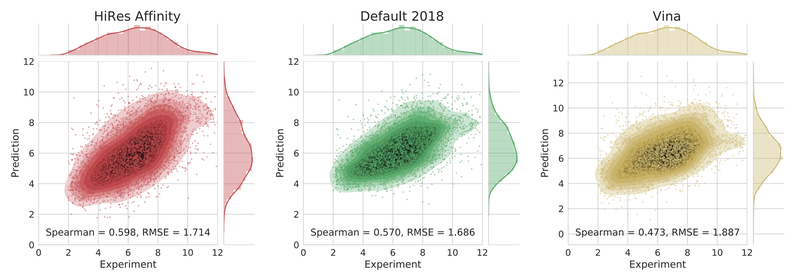


Left *Selecting the lowest RMSD pose for scoring results in a slight improvement in affinity prediction.*

Right *A random, unclustered, cross-validation split will give an unrealistic estimate of performance on new targets.*

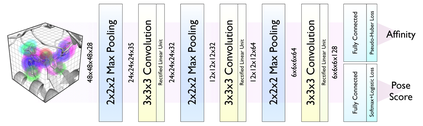


***Right** Performance on the weekly Continous Evaluation of Ligand Pose Prediction (CELPP) is less impressive than benchmark evaluation.*

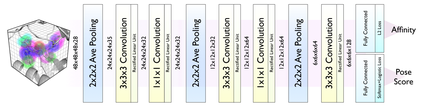


Models

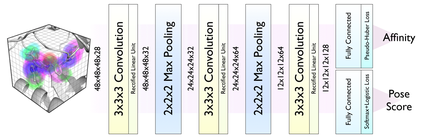
Default2017 Baseline model for optimization



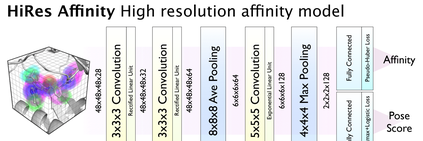
Default2018 Optimized model with network-in-network architecture and fast evaluation



HiRes Pose High resolution pose selection model



HiRes Affinity High resolution affinity model



Acknowledgements

This research was supported by R01GM108340 from the National Institute of General Medical Sciences and contributions from aigrant.org, Google Cloud, NVIDIA Corporation, and the University of Pittsburgh Center for Simulation and Modeling.