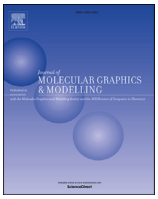
Journal of Molecular Graphics and Modelling 84 (2018) 96–108

Contents lists available at ScienceDirect

Journal of Molecular Graphics and Modelling

journal homepage: www.elsevier.com/locate/JMGM



Visualizing convolutional neural network protein-ligand scoring

Joshua Hochuli, Alec Helbling, Tamar Skaist, Matthew Ragoza, David Ryan Koes*

Department of Computational and Systems Biology, University of Pittsburgh, 3501 Fifth Ave, Pittsburgh, PA, 15260, United States

ARTICLE INFO

Article history:
Received 7 March 2018
Received in revised form
18 May 2018
Accepted 7 June 2018
Available online 18 June 2018

Keywords:
Protein-ligand scoring
Molecular visualization
Deep learning

ABSTRACT

Protein-ligand scoring is an important step in a structure-based drug design pipeline. Selecting a correct binding pose and predicting the binding affinity of a protein-ligand complex enables effective virtual screening. Machine learning techniques can make use of the increasing amounts of structural data that are becoming publicly available. Convolutional neural network (CNN) scoring functions in particular have shown promise in pose selection and affinity prediction for protein-ligand complexes.

Neural networks are known for being difficult to interpret. Understanding the decisions of a particular network can help tune parameters and training data to maximize performance. Visualization of neural networks helps decompose complex scoring functions into pictures that are more easily parsed by humans. Here we present three methods for visualizing how individual protein-ligand complexes are interpreted by 3D convolutional neural networks. We also present a visualization of the convolutional filters and their weights. We describe how the intuition provided by these visualizations aids in network design.

1. Introduction

Protein-ligand scoring is an important computational method in a drug design pipeline [1–6]. In structure-based drug design methods, such as molecular docking, scoring is an essential subroutine that distinguishes between correct and incorrect binding modes and ranks the probability that a candidate molecule is active. Improved scoring methods will result in more effective virtual screens that more accurately identify enriched subsets of drug candidates, providing more opportunities for success in subsequent stages of the drug discovery pipeline.

The wealth of protein-ligand structural and affinity data enables the development of scoring functions based on machine learning [7–14]. Of particular interest are methods that use convolutional neural networks (CNNs) [15–21] to recognize potent protein-ligand interactions, as CNNs have been remarkably successful at the analogous image recognition problem [22–24]. Unlike force field or empirical scoring functions, whose functional form is designed to represent known physical interactions such as hydrogen bonding or steric interactions, machine learning methods can derive both their model structure and parameters directly from the data. However, this increase in model expressiveness comes at the cost of reduced model interpretability.

The lack of interpretability of a CNN model presents challenges both when developing a scoring function and in understanding its application. Choosing input representations, managing training and test data, and determining optimal parameters all depend on understanding how the CNN behaves. Simple “black box” treatment of the model is not sufficient to guide such decisions. Additionally, visualizations can provide human-interpretable insights to help guide medicinal chemistry optimization.

In the image classification domain, there are a number of methods that provide insight into the inner workings of a trained CNN by projecting network decisions back on the readily visualized input image. These methods reveal what parts of an input image are important [25,26] and how the input is represented at different layers in the network [27]. Loss gradients have also been used to visualize what aspects of its input a model has learned to favor for different predicted classes [28]. Here we investigate grid-based CNN scoring of protein-ligand complexes and show how network decisions can be projected back to an atomistic granularity.

We visualize the convolutional filters of the first layer of the network to gain insight in the initial featurization learned. In order to gain atomistic insight into specific network decisions (e.g., why a ligand is scored as having a high/low affinity), we introduce and compare three methods for projecting the network's decision onto the molecular input: masking, gradient, and conserved layer-wise relevance propagation (CLRP). CLRP is a novel refinement of

* Corresponding author.
E-mail address: dkoes@pitt.edu (D.R. Koes).

https://doi.org/10.1016/j.jmgm.2018.06.005
1093-3263/© 2018 Elsevier Inc. All rights reserved.

layer-wise relevance propagation (LRP) [29,30] that better compensates for zero-weight activations. This is important since such activations emerge naturally from "empty" space in the input where there are no protein or ligand atoms (e.g. implicit solvent). This enables visualizations that account for the contributions of solvent to the final prediction of the network.

We apply each method to a network that was trained for both pose selection (distinguish low-RMSD from high-RMSD poses) and affinity prediction. Convolutional filter visualization provides insight into the low-level features identified by the network. We compare and contrast the three atomistic visualizations and show how they provide different insights and have different properties. Our visualization implementations and CNN models are available under an open-source license as part of gnina, our framework for structure-based deep learning based off of AutoDock Vina [31] and Caffe [32], at https://github.com/gnina.

2. Methods

After describing the design and training of a CNN model for pose scoring and affinity prediction, we describe an approach for analyzing the learned weights of the first layer of a grid-based CNN model and three distinct methods for mapping a CNN prediction back onto the atomic input.

2.1. Training

For our CNN model, we extend our previously described architecture [15] as shown in Fig. 1. The atoms of the input complex are represented using truncated Gaussians and 35 distinct atom types, shown in Table 1. This continuous representation is discretized onto a cubic grid that is 23.5 Å on each side and has a resolution of 0.5 Å. The input is fed through three units of max pooling and convolution with rectified linear unit (ReLU) activation functions. Each convolutional layer applies a $3 \times 3 \times 3$ convolutional filter across its input with a stride of one to generate an output feature map with the same dimension as the input. The result of the convolutional layers is mapped to the network outputs with two separate fully connected layers, with no hidden layers. One fully connected layer is trained to score poses by generating a probability distribution over the two pose classes, low (<2Å) RMSD and high (>4Å) RMSD poses, using a softmax layer (which scales predictions to be between zero and one and sum to one) and a logistic loss function:

$$\sigma(\widehat{y})_i = \frac{e^{\widehat{y}_i}}{\sum_{j=1}^{K} e^{\widehat{y}_j}} \tag{1}$$

Table 1
The 35 atom types used in gnina. Carbon atoms are distinguished by aromaticity and adjacency to polar atoms ("NonHydrophobe"). Polar atoms are distinguished by hydrogen bonding propensity.

Receptor Atom Types	Ligand Atom Types
AliphaticCarbonXSHydrophobe	AliphaticCarbonXSHydrophobe
AliphaticCarbonXSNonHydrophobe	AliphaticCarbonXSNonHydrophobe
AromaticCarbonXSHydrophobe	AromaticCarbonXSHydrophobe
AromaticCarbonXSNonHydrophobe	AromaticCarbonXSNonHydrophobe
Calcium	Bromine
Iron	Chlorine
Magnesium	Fluorine
Nitrogen	Nitrogen
NitrogenXSAcceptor	NitrogenXSAcceptor
NitrogenXSDonor	NitrogenXSDonor
NitrogenXSDonorAcceptor	NitrogenXSDonorAcceptor
OxygenXSAcceptor	Oxygen
OxygenXSDonorAcceptor	OxygenXSAcceptor
Phosphorus	OxygenXSDonorAcceptor
Sulfur	Phosphorus
Zinc	Sulfur
	SulfurAcceptor
	Iodine
	Boron

$$L_{pose}(y, \widehat{y}) = -\sum_{i=1}^{K} 1(y = i)\log(\sigma(\widehat{y})_i) \tag{2}$$

The other fully connected layer is trained to predict the binding affinity in log units using a pseudo-Huber loss function. This loss interpolates between an L2 and L1 loss according to a parameter δ to reduce outlier bias:

$$L_{pseudo-Huber}(y, \widehat{y}) = \delta^2 \sqrt{1 + \left(\frac{y - \widehat{y}}{\delta}\right)^2} - \delta^2 \tag{3}$$

As the training set includes incorrect (>4 Å RMSD) poses, for which the correct binding affinity is not well-defined, a hinge loss is used so that the affinity prediction loss is only incurred on high RMSD poses if the affinity is predicted to be too high. The complete model used for training is available at https://github.com/gnina/models.

For training data we use a set of poses generated by redocking the ligands of the 2016 PDBbind refined set [33]. Poses were generated using the AutoDock Vina scoring function [31] as implemented in smina [13]. The binding site for docking was defined using the pocket residues specified in the PDBbind. The input ligand conformation was generated from 2D SMILES using RDKit [34]. To increase the number of low RMSD poses in the training set, the docked poses were supplemented by energy

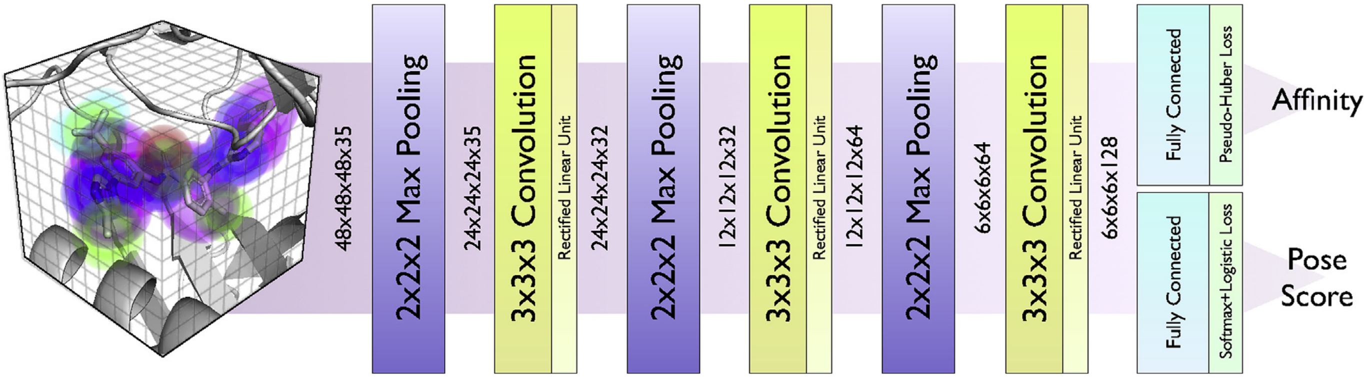


Fig. 1. Architecture of the network used to evaluate visualization methods. The input is a voxelized grid of Gaussian atom type densities.