Visualization of Convolutional Neural Network Scoring of Protein-Ligand Binding

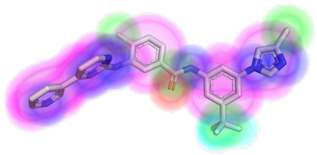
Joshua Hochuli[1,2], Matthew Ragoza[3], and David Ryan Koes[3]

[1]Department of Computer Science, [2]Department of Biological Sciences, [3]Department of Computational and Systems Biology, University of Pittsburgh, Pittsburgh, PA

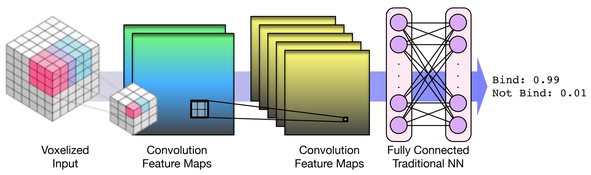
Abstract

Convolutional neural networks provide a promising approach to scoring protein-ligand interactions. Neural networks are inherently difficult to analyze and understand, often being called 'black boxes'. Running a given protein-ligand complex through a network simply produces a value from 0 to 1 to represent the probability that the pose is correct, without providing any insight as to how that number was generated. Visualizations of the neural network's decision-making process allow for the analysis of its understanding of chemical interactions. We describe multiple visualization workflows and provide examples of how the resulting visualizations can aid in biological and chemical understanding of the interaction while also discussing current limitations of the method.

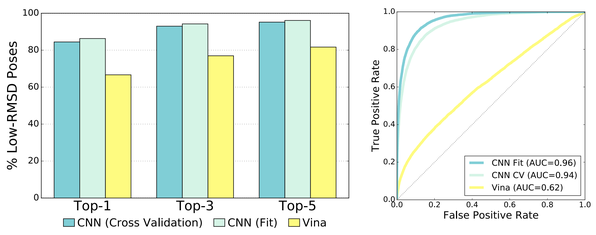
Background



The ligand and receptor atoms of a complex are represented as Gaussian densities of atom types.



Atom densities are voxelized to a grid that is the input to a 3D convolutional neural network. The network is trained to distinguish between low (<2Å) RMSD and high (>4Å) RMSD docked poses generated using AutoDock Vina and the 4056 structures of the refined set of PDBbind.



The trained network substantially outperforms AutoDock Vina at selecting and ranking poses. Visualizations of the network provide insights into what the network has learned and how it processes specific features.

References

Ragoza, M., Hochuli, J., Idrobo, E., Sunseri, J. and Koes, D.R., 2016. *Protein-Ligand Scoring with Convolutional Neural Networks*. arXiv preprint arXiv:1612.02751

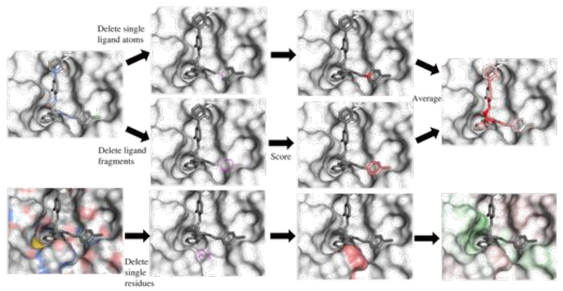
Samek, W., Binder, A., Montavon, G., Lapuschkin, S. and Müller, K.R., 2016. *Evaluating the visualization of what a deep neural network has learned.* IEEE Transactions on Neural Networks and Learning Systems.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R. and Samek, W., 2015. *On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation*. PloS one, 10(7), p.e0130140.

Masking

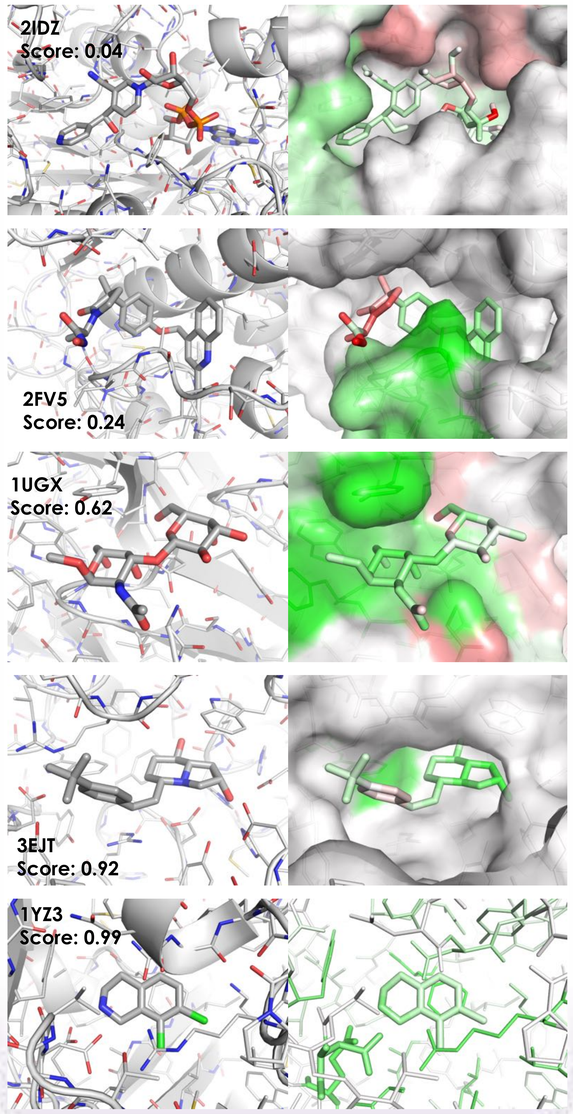
How the CNN scores changes to the structure

Atom importance is determined by the change in score upon removal of the atom, either individually or as part of molecular fragments or protein residues.



Pros:
- Easy to implement – no need to modify CNN
- Can evaluate large modifications (fragments)

Cons:
- Evaluates non-physical structures
- Inefficient – requires many CNN evaluations
- No volumetric output



Gradients

How the CNN changes the structure to score better

Backpropagation propagates the gradient of the loss throughout the network and can ultimately calculate the gradient with respect to the input values.

$$\delta^N = \nabla_a L \odot \sigma'(z^N)$$ ***Compute output error***

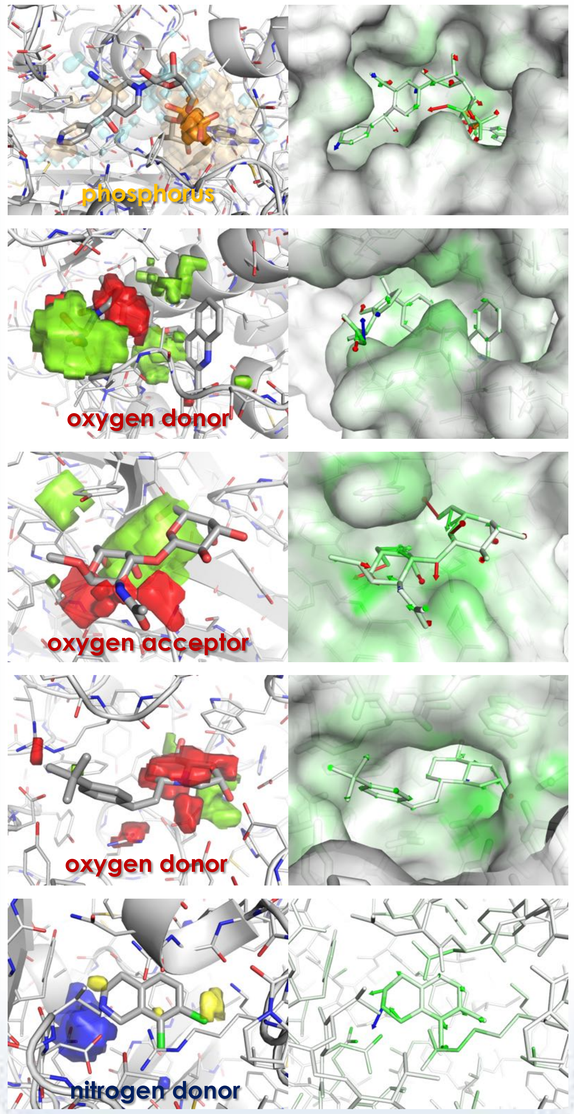
$$\delta^l = \left((w^{l+1})^T \delta^{l+1}\right) \odot \sigma'(z^l)$$ ***Backpropagate***

$$\frac{\partial L}{\partial x_j} = \delta_j^0$$ ***Calculate gradient with respect to input***

$$\frac{\partial L}{\partial A_{\{x,y,z\}}} = \sum_j \frac{\partial L}{\partial x_j} \frac{\partial x_j}{\partial dist_{A,x_j}} \frac{\partial dist_{A,x_j}}{\partial A_{\{x,y,z\}}}$$

Pros:
- Implemented in a single backwards test
- Indicates how to improve structure
- Volumetric visualization
- Provides directionality

Cons:
- Intermixes positive and negative effects
- Does not explain current score



Relevance

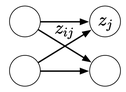
How the CNN scores the unchanged structure

Relevance propagation propagates the classifier output throughout the network as a relevance quantity. At each network level the total relevance is conserved.

$$f(x) = \ldots = \sum_{d \in l+1} R_d^{(l+1)} = \sum_{d \in l} R_d^{(l)} = \ldots = \sum_d R_d^{(1)}$$

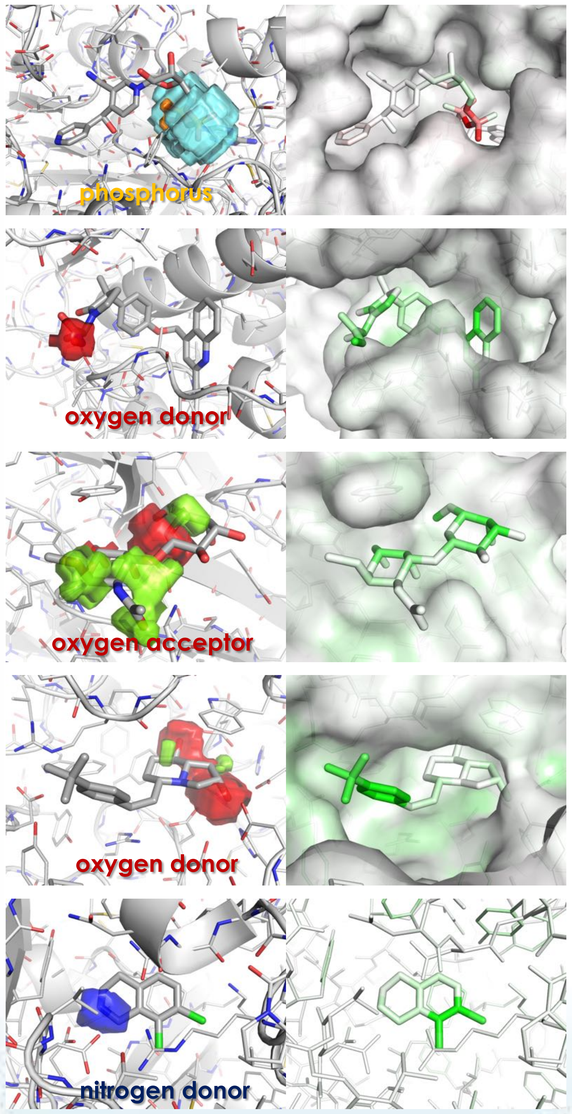
The negative and positive components of the relevance are treated separately.

$$R_{i \leftarrow j}^{(l,l+1)} = R_j^{(l+1)} \cdot \left(\alpha \cdot \frac{z_{ij}^+}{z_j^+} + \beta \cdot \frac{z_{ij}^-}{z_j^-}\right)$$



Pros:
- Implemented in a single backwards test
- May better delineate classifier decision boundary
- Separates positive and negative effects
- Volumetric visualization
- Explains current score

Cons:
- Custom implementation for each network layer
- Does not extrapolate beyond input
- No directionality



This work is supported by R01GM108340 from the National Institute of General Medical Sciences