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

2019 Spring @ https://qdata.github.io/deep2Read/

Presentation by Eli Draizen

4/10/19
Background

- CNNs have been great at predicting protein-ligand interactions poses and binding affinities
- However, CNNs are difficult to interpret
- A new method is needed to:
  - Reveal which parts of the atoms are important
  - Understand how the atoms are represented at different layers
  - Understand how what aspects of the atoms the model learns to favor different classes
- Created 4 new visualization methods:
  - First layer filter heatmaps
  - Masking
  - Gradient
  - Conserved Layer-wise Relevance Propagation
Method

- 23.5 Å x 23.5 Å x 23.5 Å grids @ 0.5 width voxels
- 3 x 3 x 3 filter, stride 1
- Atom coordinates discretized into 4D grid (3D space + 1D atom type features) based on the Van der Waals radius and distance of atom to grid point
  - Will explain in detail during the Atomic Gradient
Loss Functions

**Affinity**
- Log units using pseudo-Huber
- Interpolated b/w L2 and L1 loss according parameter $\delta$

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

- If low resolution (>4Å RMSD), use hinge loss instead

**Pose Score**
- Score poses by generating probability distribution over high res (<2Å) and low res (>4Å), scaled to [0,1] with softmax
- Logistic loss:

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

$$\sigma(\hat{y})_i = \frac{e^{\hat{y}_i}}{\sum_{j=1}^{K} e^{\hat{y}_j}}$$
## Input Data

### Data Sources

1. Known poses, binding sites, and binding affinities from PDBBind2016 (15,814 protein-ligand complexes)
2. Alternate conformers of ligand generated with RDKit and redocked using Vina
3. Predicted poses from model after 3 rounds of iteratively training

**Total: 255,035 protein-ligand complexes**

### Features

<table>
<thead>
<tr>
<th>Receptor Atom Types</th>
<th>Ligand Atom Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>AliphaticCarbonXSHydrophobe</td>
<td>AliphaticCarbonXSHydrophobe</td>
</tr>
<tr>
<td>AliphaticCarbonXSNonHydrophobe</td>
<td>AliphaticCarbonXSNonHydrophobe</td>
</tr>
<tr>
<td>AromaticCarbonXSHydrophobe</td>
<td>AromaticCarbonXSHydrophobe</td>
</tr>
<tr>
<td>AromaticCarbonXSNonHydrophobe</td>
<td>AromaticCarbonXSNonHydrophobe</td>
</tr>
<tr>
<td>Calcium</td>
<td>Bromine</td>
</tr>
<tr>
<td>Iron</td>
<td>Chlorine</td>
</tr>
<tr>
<td>Magnesium</td>
<td>Fluorine</td>
</tr>
<tr>
<td>Nitrogen</td>
<td>Nitrogen</td>
</tr>
<tr>
<td>NitrogenXSAcceptor</td>
<td>NitrogenXSAcceptor</td>
</tr>
<tr>
<td>NitrogenXSDonor</td>
<td>NitrogenXSDonor</td>
</tr>
<tr>
<td>NitrogenXSDonorAcceptor</td>
<td>NitrogenXSDonorAcceptor</td>
</tr>
<tr>
<td>Oxygen</td>
<td>Oxygen</td>
</tr>
<tr>
<td>OxygenXSAcceptor</td>
<td>OxygenXSAcceptor</td>
</tr>
<tr>
<td>OxygenXSDonorAcceptor</td>
<td>OxygenXSDonorAcceptor</td>
</tr>
<tr>
<td>Phosphorus</td>
<td>Phosphorus</td>
</tr>
<tr>
<td>Sulfur</td>
<td>Sulfur</td>
</tr>
<tr>
<td>Zinc</td>
<td>SulfurAcceptor</td>
</tr>
<tr>
<td></td>
<td>Iodine</td>
</tr>
<tr>
<td></td>
<td>Boron</td>
</tr>
</tbody>
</table>
Results

- Trained in 150,000 iterations with batch size 50
- Each batch balanced number of low- and high-res poses
- Every pose is randomly rotated and translated relative to ligand center
- Tested against other CSAR dataset not in training data:

![Graph](attachment:image.png)

(a) ROC curve showing TPR vs FPR for CNN (AUC=0.89) and Vina (AUC=0.61)

(b) Scatter plot comparing prediction vs experiment results for CNN (R=0.74, RMSE=1.44) and Vina (R=0.55, RMSE=1.86)
Results (Docked Poses)

(a) 1o0h: 2.698/0.255  
(b) 1w4o: 4.933/0.983  
(c) 4dv: 5.951/0.894
Convolutional Filter Visualization (Averaged)

- First layer shows how network maps atoms types
- Averaged over all dimensions (3x3x3)
- Some types have low avg wts cross all filters (metals)
  - Network isn't overfitting rare metals
- Some types have all neg vals
  - Network learned to turn off those filters. Removing them may make simpler model
Convolutional Filter Visualization (Full)
Convolutional Filter Visualization (Full)
Masking

- Repeat for all ligand atoms, color atoms by masking score
- Repeat for all protein residues in binding site, color residue by masking score
- Computationally demanding since the NN is run many times
Atomic Gradient

- All atomic coordinates (not discretized) are used directly as input
- Discretization is differentiable and is what is fed into NN

\[ g(d, r) = \begin{cases} 
    e^{-\frac{2d^2}{r^2}} & 0 \leq d < r \\
    \frac{4}{e^2r^2} d^2 - \frac{12}{e^2} d + \frac{9}{e^2} & r \leq d < 1.5r \\
    0 & d \geq 1.5r
\end{cases} \]

\[ \frac{\partial g}{\partial d} = \begin{cases} 
    \frac{4d}{r^2} e^{-\frac{2d^2}{r^2}} & 0 \leq d \leq r \\
    \frac{8}{e^2r^2} d - \frac{12}{e^2} & r < d < 1.5r \\
    0 & d \geq 1.5r
\end{cases} \]

Discritation function with VDW and distance of atom to voxel

Differentiable with respect to distance

\[ \frac{\partial f}{\partial a} = \sum_{g \in G_a} \frac{\partial f}{\partial g} \frac{\partial g}{\partial d} \frac{\partial d}{\partial a} \]

Grad of scoring func w/ respect coordinates -- chain rule + sum over grid points with same atom type that overlap the atom, Ga

- Give insight into how input should be changed to produce a better output
- Calculated forward pass first, then the backward pass computes loss gradient
- Negative vector is how atom should be moved in 3D space
Conserved Layer-wise Relevance Propagation (CLRP)

- Calculates a Relevance score that is propagated back through NN
- “Performed proportionally to the input activations of each layer, such that the relevance of node \( i \) in layer \( l \) is the sum of the relevances of its successor nodes, \( j \), weighted by the activation value generated along the edge \( z_{ij} \) during the forward pass”
  - Input activation: \( z_{ij} = x_i w_{ij} \), where node \( i \) is in layer \( l \) with successor node \( j \)
    \[
    R_i^{(l)} = \sum_j \frac{z_{ij}}{\sum_i z_{ij}} R_j^{(l+1)}
    \]
    \[
    f(x) = \ldots = \sum_{d \in l+1} R_d^{(l+1)} = \sum_{d \in l} R_d^{(l)} = \ldots = \sum_d R_d^{(1)}
    \]
    Invariant across layers
- Redistribute relevance directed at dead nodes to remaining nodes in layer

\[
S_l = \sum_j \begin{cases} 
0 & \text{if } z_j \neq 0 \\
R_j & \text{if } z_j = 0
\end{cases} \\
R_j = \begin{cases} 
0 & \text{if } z_j = 0 \\
R_j + \frac{z_j}{S_l} & \text{if } z_j \neq 0
\end{cases}
\]
Results (low scoring complex)

Affinity Prediction Score = 2.698

- **Gradients**
  - Move aromatic away from his => not learned to value aromatics

- **CLRP**
  - Focuses on central ribose => highlight decision boundaries?

- **Masking**
  - Aromatic isn't favored?
Results (low affinity score, high pose score)

Affinity Prediction Score = 4.933

- CLRPR
  - Phosphate and uracil groups more relevant

- Masking
  - T45 is more favorable, which interacts with uracil
Results (middling affinity score, good pose score)

Affinity Prediction Score = 5.951

- Gradient
  - Arrows in ring point ot center => smaller func group?
  - Shift?
- Masking
  - Disfavors aromatics
- CLRP
  - C and O of carbonyl counter-balance => artifact of decomposing score to atoms?
Additive Analysis

- Can individual atom masking scores sum to the total score?
  - Linear relationship: score can be decomposed
- Pose scoring
  - Squashed to [0, 1], changes not that meaningful
- Affinity prediction is more correlated
Atomic Score Correlations From Different Methods

- Some agreement b/w Gradient and CLRP
- However, there is a general lack of correlation, which shows each score will provide a different insights
Analyzing Empty Space

- 99% of dead nodes in 1st layer
  - Implicit solvent?
- **Green**: favorable relevance scores. If the protein or ligand filled this space it would have a higher score
- **Red**: unfavorable relevance scores. If the protein or ligand filled this space it would have a lower score
Conclusion

- All visualizations methods relay different information
- **Gradient:** Can show what the NN “wants” to produce a higher scoring output in a single forward and backward pass
- **CLRP:** Preserves the relevance of each atom in a single forward and backward pass
- **Masking:** Manipulate the input to understand the changes in values. Very costly since it runs NN thousands of times
DeepMind
Input

Predict pairwise interactions using a MaxEnt model on CATH cluster reps (~6k):

\[ P(\sigma|h,J) = \frac{1}{Z(h,J)} \exp\left( \sum_{i=1}^{N} h_i(\sigma_i) + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} J_{ij}(\sigma_i,\sigma_j) \right) \]

From EVFold Output for Ras, From John Ingram, Debbie Marks Lab
Deep Dilated Convolutional Residual network

1 residual block
Modifies a 64x64x128 representation from the previous block

Repeat 220 times, cycling through dilations 1, 2, 4, 8

21 million parameters

Dilated convolutions
Efficient long-range interaction

Batch norm
Elu
Project down
64 dim
Batch norm
Elu
3x3 dilated
Batch norm
Elu
Project up
128 dim

N x N Input features

Residual network blocks

N x N Distance predictions
Auxiliary losses

- We know the contact map encodes secondary structure
  - A distance network should be good at predicting it
- Auxiliary loss of secondary structure from 1D reductions for both (i, i+63) and (j, j+63)
  - Ensembled across all 2D crops
- Q3 Accuracy on CASP11 ~84%
- Predicting secondary structure improves contact prediction

Two N x 8 secondary structure predictions

N x N Input features

N x N x 40 Distance predictions
Accuracy vs computational cost

Repeated gradient descent

Using simple vdW instead of score2

Highly parallelizable

(Averaged over a subset of targets)