BOLTZMANN LABS PRIVATE LIMITED

PREDICTION OF PROTEIN-LIGAND BINDING AFFINITY VIA 3D-CNN

Objective:

The Objective of this project is Protein-Ligand Binding Affinity Prediction via 3D-Convolutional Neural Networks. The reference review paper for this work is K-deep which proposes an end-to-end framework based on the 3D Convolutional Neural Networks for predicting protein-Iigand binding affinity.

What is Binding Affinity?

Binding Affinity is the strength of the binding-interaction between a single biomolecule (e.g. Protein or DNA) to its ligand/binding partner (e.g. drug or inhibitor).

Binding affinity is typically measured and reported by the equilibrium dissociation constant (Kd), inhibition constant (Ki) and half-concentration (IC-50). These properties are used to evaluate the ranking order and strengths of bio-molecular interactions.

The smaller the Kd value, the greater the binding affinity of the ligand for its target. The larger the Kd value, the more weakly the target molecule and ligand are attracted to and bind to one another.

Prediction of protein-ligand binding affinity is one of the most important properties in computational chemistry and drug discovery. We do use here a fast machine learning approach for prediction of binding affinity using 3D Convolutional Neural Network. Here, we focus on the accurately predicting protein−ligand binding affinity using structural information from both proteins and ligands.

DATASET DESCRIPTION - For the work described here, we mainly use the PDB-bind (v.2016) database, containing 13,308 protein−ligand complexes and their corresponding experimentally determined binding affinities collected from literature and the Protein Data Bank (PDB), in terms of dissosiation (Kd), inhibition (Ki) or half-concentration (IC50) constant.

A smaller refined subset (nr = 4057) is extracted from general set following quality protocols addressing structural resolution and experimental precision of the binding measurement. These controls exclude complexes with a resolution higher than 2.5 Å, or an R-factor higher than 0.25, ligands bound through covalent bonds, ternary complexes or steric clashes, and affinity not reported either in Kd or Ki or falling out of a desired range (Kd < 1pM) among other criteria. A core set (nc = 290) is selected from the larger refined set as a representative non-redundant subset for benchmarking purposes.

We basically use this refined set of datasets for our project for eliminating the redundancy and high-resolution complexes.

WORKFLOW OF ALGORITHM -

Both protein and ligand are featurized via a voxelized 24 Å representation of the binding site considering eight pharmacophoric-like properties.

These descriptors are used by a three-dimensional convolutional neural network model, which in turn learns the binding affinity of the complex given enough training examples. Once trained, the network is used to predict unseen examples.

METHODOLOGY - We adapted the set of descriptors detailed in for both protein and ligand. In particular, we use a 3D voxel representation of both proteins and ligand using a Van Der Waals radius for each atom type, which in turns gets assigned to a particular property channel (hydrophobic, hydrogen-bond donor or acceptor, aromatic, positive or negative ionizable, metallic and total excluded volume).

We duplicate the number of properties to account for both protein and ligand, by using the same ones in each, up to a total of 16 different channels. These descriptors are computed on a fixed 24 Å3 sub-grid centered on the geometric center of the ligand, in practice capturing a neighborhood of the binding site. The voxelization routines are available within the HTMD Python framework for molecular discovery.

VOXELIZATION DETAILING - Atom typing for the protein requires the protein to be protonated and to include the atom bond information. From the data available, the Molecule does not contain all bond information. Prepare Protein for Atom typing perform most necessary operations such as removing non-protein atoms, adding hydrogens, guessing bonds and guessing protein chains.

The protonation will a) move atoms to optimize hydrogen networks b) add missing sidechains and c) the bond guessing can go wrong if atoms are very close to each other which can happen when adding side-chains.

We calculate the voxel information for the protein. By default, getVoxelDescriptors will calculate the bounding box of the molecule and grid it into voxels. As we don’t use point properties but smooth them out over space, we will add 1 A buffer space around the protein for the voxelization grid so that the properties don’t cut off at the edge of the grid. This will visualize each voxel channel as a separate VMD molecule so that you can inspect each individually. You can play around with the IsoValue in the Iso-Surface representation of each channel to inspect it in more detail.

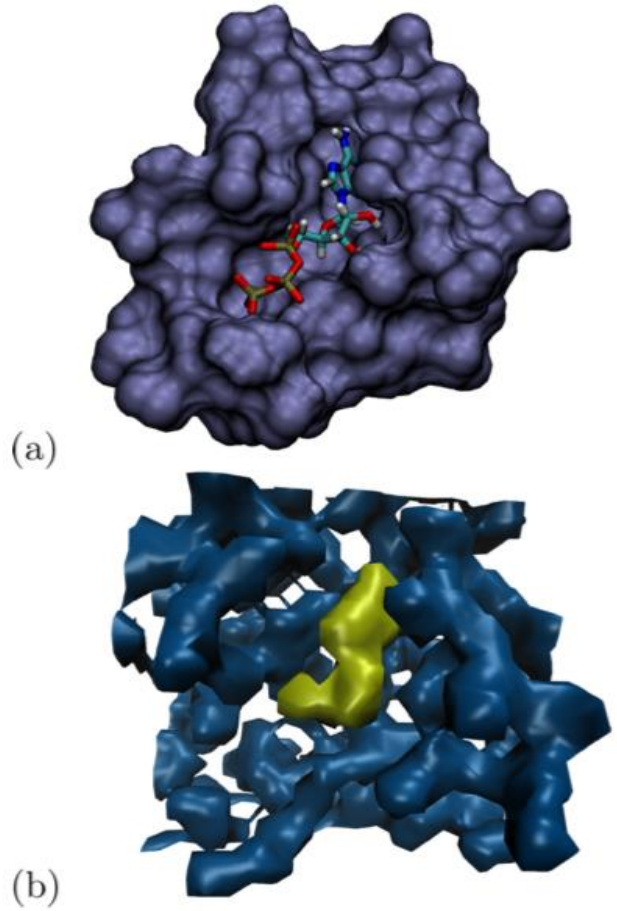

Figure 2. (a) PDB Id 2HMU pocket and bound ligand ATP. (b) Voxel representation of the hydrophobic channel for both protein (blue) and ligand (yellow).

To perform 3D Convolutions, we need to reshape the data to (nsamples, nchannels, d, h, w) with the last three dimensions corresponding to the 3D elements. Our data is in (d*h*w, nchannels) format so we first transpose and then reshape it.

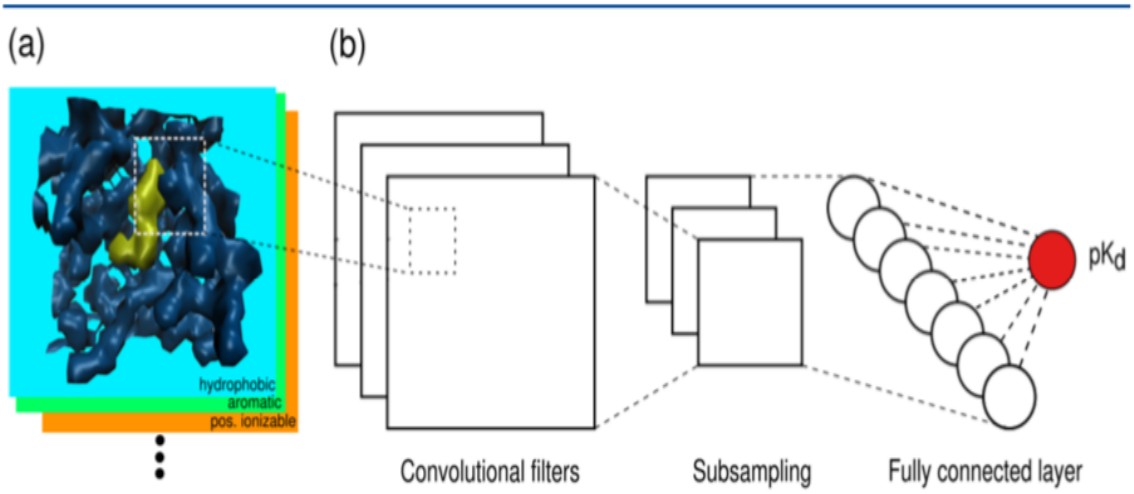

Figure 3. Workflow schema of the proposed model. (a) Both protein and ligand are featurized via a voxelized 24 Å representation of the binding site considering eight pharmacophoric-like properties. (b) These descriptors are used by a three-dimensional convolutional neural network model, which in turn learns the binding affinity of the complex given enough training examples. Once trained, the network is used to predict unseen examples.

Workflow of Algorithm explanation with the help of figure

Architectural Design Strategies-

Strategy 1. Replace 3×3 filters with 1×1 filters

- Given a budget of a certain number of convolution filters, we can choose to make the majority of these filters 1×1, **since a 1×1 filter has 9× fewer parameters than a 3×3 filter**.

Strategy 2. Decrease the number of input channels to 3×3 filters

- Consider a convolution layer that is comprised entirely of 3×3 filters. The total quantity of parameters in this layer is: (number of input channels) × (number of filters) × (3×3)
- We can **decrease the number of input channels to 3×3 filters using squeeze layers**, mentioned in the next section.

Strategy 3. Down sample late in the network so that convolution layers have large activation maps

- The intuition is that large activation maps (due to delayed downsampling) can lead to higher classification accuracy.

Summary

- Strategies 1 and 2 are about judiciously decreasing the quantity of parameters in a CNN while attempting to preserve accuracy.
- Strategy 3 is about maximizing accuracy on a limited budget of parameters.

References:

https://pubs.acs.org/doi/full/10.1021/acs.jcim.7b00650

https://pubs.acs.org/doi/suppl/10.1021/acs.jcim.7b00650/suppl_file/ci7b00650_si_001.pdf

https://software.acellera.com/docs/latest/moleculekit/tutorials/voxelization_tutorial.html

https://pytorch-lightning.readthedocs.io/en/latest/

https://pytorch-lightning.readthedocs.io/en/latest/trainer.html