Protein Interface Prediction using Graph Convolutional Networks

Alex Fout†, Jonathon Byrd†, Basir Shariat†, Asa Ben-Hur

Department of Computer Science, Colorado State University 2017

Problem and Motivation

Proteins are large, complex molecules which perform many important cellular functions. These functions typically involve proteins binding together to form complexes. Prediction of the interface between two bound proteins is an important research problem with applications in genetic diseases and pharmacological research.

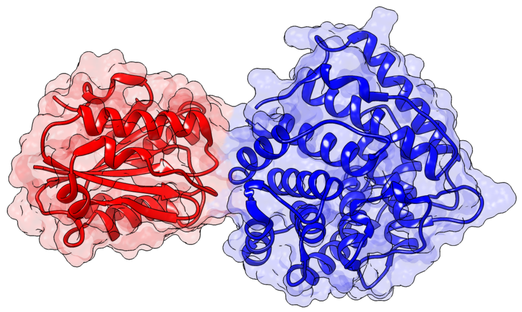

Figure: Two proteins in their bound formation (PDB ID 4M76).

We treat interface prediction as a binary classification problem that considers **whether or not a pair of amino acid residues from two proteins are part of the interface** [3].

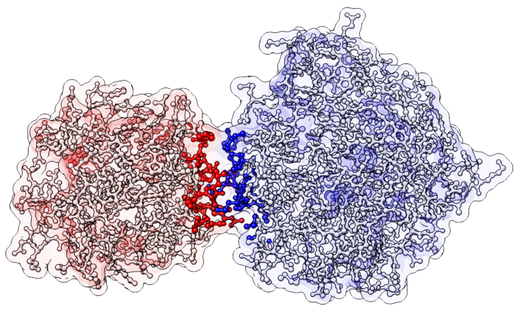

Figure: Protein binding, dark red and blue residues are part of the interface (positive class).

Proteins as Graphs

We model proteins as graphs, where nodes represent residues and edges represent relationships between residues. Features on the nodes and edges encode protein sequence and structural information.

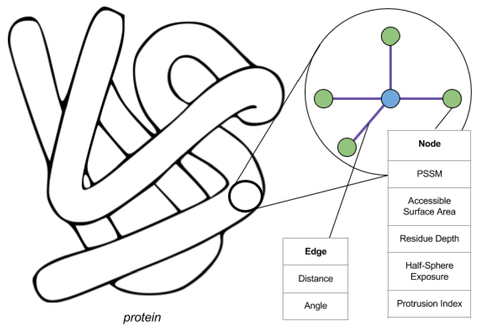

Figure: Graph representation of a local neighborhood of amino acid residues within a protein.

Learning Local Features from Graphs

Traditional Convolutional neural networks apply filters to receptive fields and generate new representations for each pixel. For graphs, node representations can be generated by applying convolutional filters to all nodes in a local receptive field.

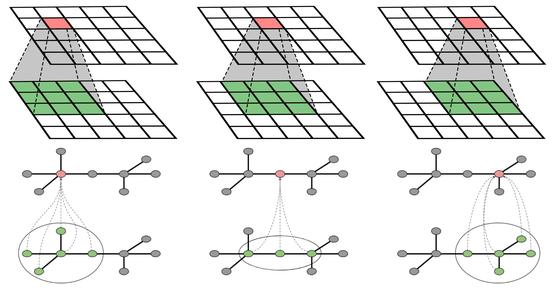

Figure: Convolution on a grid vs a graph.

However, whereas pixels in two grid-based receptive fields can be brought into correspondence, graph-based receptive fields have no obvious correspondence.

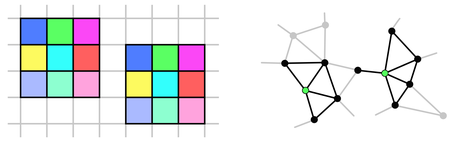

Figure: Receptive fields in a grid and graph context.

We can either impose an ordering on neighbors to give unique weights to each, or avoid ordering neighbors and assign the same weights to each:

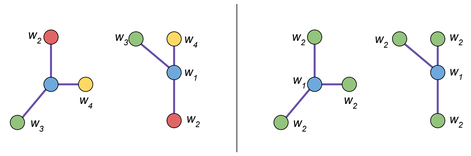

Figure: Assigning weights to graph receptive fields by either ordering (left) or by treating all neighbors identically.

We examine both unordered and ordered approaches, and also apply filters to edges.

$$z_i = \sigma\left(W^{\mathrm{C}} x_i + \frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i} W^{\mathrm{N}} x_j + b\right), \tag{1}$$

$$z_i = \sigma\left(W^{\mathrm{C}} x_i + \frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i} W^{\mathrm{N}} x_j + \frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i} W^{\mathrm{E}} A_{ij} + b\right), \tag{2}$$

$$z_i = \sigma\left(W^{\mathrm{C}} x_i + \frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i} W^{\mathrm{N}}_j x_j + \frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i} W^{\mathrm{E}}_j A_{ij} + b\right). \tag{3}$$

Data

Methods were evaluated using complexes from the Docking Benchmark Dataset[5].

Data Partition	Complexes	Positive examples	Negative examples
Train	140	12,866 (9.1%)	128,660 (90.9%)
Validation	35	3,138 (0.2%)	1,874,322 (99.8%)
Test	55	4,871 (0.1%)	4,953,446 (99.9%)

Table: Training, validation, and testing data.

Pairwise Classification and Network Architecture

Examples are composed of pairs of residues from two proteins, i.e. we classify pairs of nodes from two separate graphs.

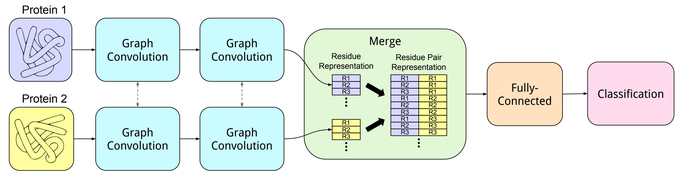

Figure: Pairwise network architecture.

Our graph convolutional layers can be stacked to learn hierarchical representations. Features are then combined to form the representation of pairs of residues, which is passed through a standard dense layer before classification.

Results

Method	Convolutional Layers 1	2	3	4
No Convolution	**0.812 (0.007)**	0.810 (0.006)	0.808 (0.006)	0.796 (0.006)
Diffusion (DCNN) (2 hops) [1]	**0.790 (0.014)**	–	–	–
Diffusion (DCNN) (5 hops) [1])	**0.828 (0.018)**	–	–	–
Single Weight Matrix (MFN [2])	0.865 (0.007)	0.871 (0.013)	**0.873 (0.017)**	0.869 (0.017)
Node Average (Eq (1))	0.864 (0.007)	0.882 (0.007)	**0.891 (0.005)**	0.889 (0.005)
Node and Edge Average (Eq (2))	0.876 (0.005)	**0.898 (0.005)**	0.895 (0.006)	0.889 (0.007)
DTNN [4]	0.867 (0.007)	0.880 (0.007)	**0.882 (0.008)**	0.873 (0.012)
Order Dependent (Eq (3))	0.854 (0.004)	0.873 (0.005)	**0.891 (0.004)**	0.889 (0.008)

Table: Median AUC across all complexes in the test set. Results shown are the average and standard deviation over ten runs. The previous state-of-the-art SVM method, PAIRPred [3], achieves a median AUC of 0.863.

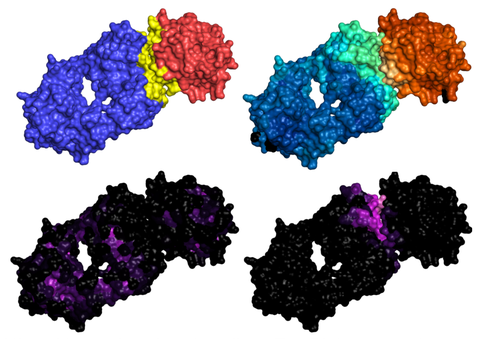

Figure: Top Left: 3HI6, with true interface in yellow. Top Right: network output for each residue, maxed over all possible neighbors, with brighter colors indicating higher scores. Bottom: Activations for two convolution filters.

References

[1] James Atwood and Don Towsley. "Diffusion-convolutional neural networks". In: *Advances in Neural Information Processing Systems*. 2016, pp. 1993–2001.

[2] David K Duvenaud et al. "Convolutional networks on graphs for learning molecular fingerprints". In: *Advances in Neural Information Processing Systems*. 2015, pp. 2224–2232.

[3] Afsar Minhas et al. "PAIRpred: Partner-specific prediction of interacting residues from sequence and structure". In: *Proteins: Structure, Function, and Bioinformatics* 82.7 (2014), pp. 1142–1155.

[4] Kristof T Schütt et al. "Quantum-chemical insights from deep tensor neural networks". In: *Nature communications* 8 (2017), p. 13890.

[5] Thom Vreven et al. "Updates to the integrated protein–protein interaction benchmarks: docking benchmark version 5 and affinity benchmark version 2". In: *Journal of molecular biology* 427.19 (2015), pp. 3031–3041.

This work was supported by the National Science Foundation under grant no DBI-1564840.

† denotes equal contribution