DREU Final Report: Exploring Diffusion State Distance and Using it in Protein Function Prediction

Indrani Ray
University of California, Berkeley
ray-indrani@berkeley.edu

Mentor: Dr. Lenore Cowen
Tufts University
lenore.cowen@tufts.edu

August 4, 2017

Abstract

Diffusion State Distance (DSD), defined in [1], is a novel way to calculate distances in graphs. For a connected graph, DSD outputs a distance matrix of distances between all pairs of nodes. We worked on two projects: 1) We tried to improve upon methods for protein function prediction in protein-protein interaction (PPI) networks. 2) We explored the landscape of possible DSD matrices for different popular families of unweighted graphs.

1 Introduction

1.1 Protein Function Prediction

An important problem in biological networks is protein function prediction. The idea behind protein function prediction is to use the known functionali-

ties of some proteins to predict the unknown functionalities of other proteins based on associations and connections between these two sets in a protein-protein interaction (PPI) network. Cao et al. [1]. developed a metric, Diffusion State Distance (DSD), which takes into account graph diffusion when calculating proximity between nodes. More specifically, given a node, its closest t neighbors in DSD distance vote on which functional labels to assign to it. Together with Taylor Yeracaris from Carleton College and Joshua Tso from Tufts University, I looked at trying a multistep process as described below.

1.2 DSD Matrices for Popular Families of Graphs

The second project was to use the DSD matrix to explore graph non-isomorphisms. The idea is that every connected graph has a corresponding DSD matrix. The elements in those matrices can be lexicographically sorted by rows and columns (Note: this will get rid of the labeling of DSD values to associated node pairs.). The claim is that the resulting sorted matrices are invariant under graph isomorphisms. Thus, if two graphs give different sorted matrices, then they are not isomorphic. To this extent, we began an exploration of DSD matrices of different families of graphs (complete, straight paths, trees, etc.) with different numbers of nodes. A few interesting claims were made under these observations, but the topic is still being studied for future work.

1.3 Structure of this Paper

The following sections are organized as follows. The following three sections are with regards to the protein function prediction work: Section 2 gives some background information including how DSD works and a description of where our data comes from, Section 3 describes our methodology, and Section 4 explains our results. Section 5 discusses the ongoing work corresponding to DSD matrices and graphs.

2 Background

2.1 How DSD Works

From Cao et al. [1] we get the following (which has been rephrased in this paper). Let $V = \{v_1, v_2, ..., v_n\}$ be the set of all vertices in a connected, undirected graph. Define $He^{\{k\}}(A, B)$ as the expected number of times a random walk of k steps starting from node A will reach node B in an unweighted graph. For a weighted graph, we bias the walk proportionally toward the more confident edges. Assuming k is fixed, we can write this as $He(A, B)$. Then, $\forall v_i \in V$ we can define an n-dimensional vector $He(v_i) = (He(v_i, v_1), He(v_i, v_2), ..., He(v_i, v_n))$. The Diffusion State Distance between two vertices, $u, v \in V$ is $DSD(u, v) = ||He(u) - He(v)||_1$ which is the L_1 norm of the He vectors of u and v.

2.2 The PPI Network

Our work uses the MIPS (Munich Information Center for Protein Sequences) FunCat (functional catalogue) mapping between proteins and functional categories [2]. The deeper levels in MIPS are more informative and specific than the higher levels, so this work uses MIPS Third Level Annotations. The PPI network we use contains protein pairs and confidence scores associated with the likelihood that there is an edge between them. These come from all known physical interactions in the *S. cerevisiae* PPI network which comes from version 3.2.102 of the BioGRID database [3]. We run this through a package known as cDSD [4] which uses the largest connected component which contains 4,990 nodes and 74,310 edges.

3 Our Methods

The overall procedure for protein function prediction is similar for both classes of methods and is as follows. We first take the list of of proteins and functionalities which comes from the MIPS Third Level Annotation [2]. Entries in this list include the protein and corresponding functionalities (e.g. YGL122C 11.04.03 16.03.03 20.01.21). We then randomly split the list into two groups – known and unknown. For the unknown group, we erase the

functionalities. This set becomes the test set. The other is the training set. We use what we know about the PPI network from entries in the BioGrid database [3]. which show two proteins and a corresponding confidence score about whether or not they are connected (e.g. YGL122C YJR138W 0.25), to predict the functionalities of the unknown group based on the functionalities of proteins "around" them. To make "around" more definite, we use cDSD [4]. This determines a distance between every node pair in the network. For each unknown node, we rank its neighbors based on DSD, and use a set of the top "closest" neighbors to make predictions. We finally cross-validate these results by seeing if the guessed functionality for each unknown protein was part of one of the original functionalities that had been initially erased. We tally up how many predictions were true and develop an accuracy percentage to study the success rate of our methods.

3.1 Multiple Runs

For this method, we look at the unknown proteins and their neighbors within a certain radius. The radius is determined for each unknown node by the maximum DSD of the top ten closest known neighbors. Using this set of neighbors, we predict each unknown node's functionality by looking at the functionalities of its known neighbors and weighing these values using the reciprocal of the DSD between the two nodes. Therefore, a known node that is "closer" to the unknown has a higher weight assigned to its votes. The functionality with the maximum vote wins. Ties are broken by random selection from the most popular votes, and the unlabeled nodes are labeled with their top predictions. . The process is run a second time with these predictions, so that if a previously unknown node is "closer" to an unknown node than an actually known one, it can also use its recently predicted functionality to vote for a new prediction for this unknown node. However, the votes cast from these prior predictions are only weighted half as much (i.e. $\frac{0.5}{DSD}$). Once again, ties are broken by random selection from the most popular votes. The final predictions are then cross-validated with the values that were initially erased.

There are two supplementary methods based on this one that incorporate a notion we will call "threshold". This is used during the first round. For each unknown gene, instead of having one functionality with the maximum vote be the initial prediction, we have a set of functionalities (whose votes are greater than or equal to the threshold value times the maximum vote). This

set of functionalities is then used to make predictions in the second round. So far, we have tried threshold values of 0.7 and 0.9.

3.2 Cascade Version

For this method, we take the list of unknown proteins and rank them based on the proportion of the top 10 "closest" neighbors that are known, weighted by $\frac{1}{DSD}$ (in other words, each known neighbor contributes a vote which is the reciprocal of the distance to the unknown node and these values are summed up for each unknown node). We go through the ranked list of unknown nodes and assign unknown proteins functionalities based on the most popular votes from their neighbors. However, the vote for an unknown node may depend on both functionalities predicted for other unknowns and actual functionalities from knowns. So, the top unknown node will get a prediction based on its known neighbors' functionalities, but as we progress down the ranked list, some unknowns chosen functionality may depend on the predictions found for other prior unknowns. Again, the functionality is weighted $\frac{1}{DSD}$ if it comes from a known node and $\frac{0.5}{DSD}$ if it comes from an unknown node, and ties are broken by random selection.
There are two additional methods based on this one. The first one updates the ranking of the unknown list in real time (so every time an unknown's unknown neighbor gets predicted, it's number of "known" neighbors increases). This in turn changes the order in which the nodes are predicted. The second method is similar to the original, but uses the radius method from the Multiple Runs (i.e. instead of looking at the top 10 neighbors, it looks at the neighbors within the radius which is equal to the maximum DSD of the 10th "closest" known neighbor).

4 Results and Conclusion

The results from these experiments are summarized in the following table. As most accuracies values fell near each other, their average is used as a statistical measure. MR corresponds to the Multiple Runs method and CV corresponds to the Cascade Version method. We have yet to collect more results, and so far, each method has been tried a different number of times. We make a result correct if we successfully label it with one of its known

functional labels. There are 181 different different functionalities on the 3rd level of MIPS.

Method Used	Number of Trials	Average Accuracy
MR original	8	49.17%
MR with 0.7 threshold	14	48.86%
MR with 0.9 threshold	9	49.13%
CV original	11	49.44%
CV with realtime ranking	6	49.12%
CV with radius method	11	48.61%

As most of our results are near 50%, we are doing slightly better than previous methods which have around a 45% accuracy rate [1]. One ongoing idea we have to improve this is to keep the set of all predictions for each unknown node after the first round when we use a prediction to make a new one for the Cascade Version; but, it is yet to be implemented. For future work, we hope to develop better methods from looking at the biological interactions between proteins and combining cDSD with other function prediction methods.

5 DSD Matrices and Graphs

The idea behind this work is to explore DSD matrices for different graph classes. In order to do so, various graphs of 3, 4, 5, 6, and more nodes were studied – in particular, complete graphs, trees, and straight paths. The DSD matrices for the graphs were computed using an online server [1]. The following claims and conjectures were found, but have yet to be proven.

5.1 DSD Matrices of Complete Graphs

A pattern was found from looking at the DSD matrices of the complete graphs with 3, 4, 5, and 6 nodes. This pattern was used to make predictions about the DSD matrices of complete graphs of 7, 8, 9, and 10 nodes; the predictions were then validated by using the online server. The pattern is as follows: The converged DSD matrices for complete graphs have 0's along the diagonal and the same value, let us call it k, everywhere else. Claim: For an $n+1$-noded complete graph, k is the sum of the reciprocals of the triangular numbers up to the triangular number corresponding to that with base n. In other words, $k = 1 + \frac{1}{3} + \frac{1}{6} + \frac{1}{10} + ... + \frac{1}{T_n}$.
In order to prove this, the method of computing DSD was studied by hand. A 5-node complete graph was observed and DSD was computed by hand for various lengths of random walks. The following was found:

Length of Random Walk	k value	k value
0	2	2
1	$\frac{3}{2}$	$2 - \frac{1}{2}$
2	$\frac{13}{8}$	$2 - \frac{1}{2} + \frac{1}{8}$
3	$\frac{51}{32}$	$2 - \frac{1}{2} + \frac{1}{8} - \frac{1}{32}$

The DSD values of random walks on a complete graph of 5 nodes are the sums of terms in a geometric series with initial term 2 and common ratio $\frac{-1}{4}$. So for a random walk of length m, $k = \frac{2(\frac{-1}{4}^n - 1)}{\frac{-5}{4}}$. This claim was generalized to say: The k value for a complete graph of $n+1$ nodes with random walk of length m is the sum of the first $m+1$ terms of a geometric series with first term 2 and common ratio $\frac{-1}{n}$. From here, it follows that for a converged DSD matrix, the k value is what this series converges to. These claims

are yet to be proven. However, it was found that the sums of reciprocals of triangular numbers is the converged sum of this geometric series. I.e. $1 + \frac{1}{3} + \frac{1}{6} + \frac{1}{10} + ... + \frac{1}{T_n} = \frac{2}{1-\frac{-1}{n}}$.

Proof (By Induction on the Number of Nodes)

Let a be the number of nodes.

Base case: $a = 2$. Then, $1 + \frac{1}{3} = \frac{4}{3} = \frac{2}{1-\frac{-1}{2}}$. We are done.

Inductive hypothesis: Assume the equation holds true for $a = x$. That is, $1 + \frac{1}{3} + \frac{1}{6} + \frac{1}{10} + ... + \frac{1}{T_x} = \frac{2}{1-\frac{-1}{x}}$.

Show it holds true for $a = x + 1$.

$$\begin{aligned}
1 + \frac{1}{3} + \frac{1}{6} + \frac{1}{10} + ... + \frac{1}{T_x} + \frac{1}{T_{x+1}} &= \frac{2}{1-\frac{-1}{x}} + \frac{1}{T_{x+1}} \\
&= \frac{2}{1-\frac{-1}{x}} + \frac{1}{\binom{x+2}{2}} \\
&= \frac{2}{1-\frac{-1}{x}} + \frac{1}{\frac{(x+2)(x+1)}{2}} \\
&= \frac{2}{1-\frac{-1}{x}} + \frac{2}{(x+2)(x+1)} \\
&= \frac{2x}{(x+1)} + \frac{2}{(x+2)(x+1)} \\
&= \frac{2x(x+2)+2}{(x+1)(x+2)} \\
&= \frac{2x^2+4x+2}{(x+1)(x+2)} \\
&= \frac{(2x+2)(x+1)}{(x+1)(x+2)} \\
&= \frac{2x+2}{x+2} \\
&= \frac{2}{1-\frac{-1}{x+1}}
\end{aligned}$$

Therefore, $1 + \frac{1}{3} + \frac{1}{6} + \frac{1}{10} + ... + \frac{1}{T_n} = \frac{2}{1-\frac{-1}{n}}$.

5.2 Other Findings and Future Work

1. In order to prove the above relationship of converged DSD matrices being equal to the sum of the terms in a geometric series, we again computed DSD by hand for a graph with 5-nodes with lengths of random walks of 0, 1, 2, and 3. Given that an initial node had weight 1, we found that at each step of a random walk, the weight of the non-initial nodes were equal. For a complete graph of 5 nodes, these weights were $0, \frac{1}{4}, \frac{3}{16}, \frac{13}{64}$ for random walks of length 0, 1, 2, and 3 respectively. We came up with a formula to determine the weight at the non-initial nodes for random walks of length m for a graph with 5 nodes to be $2 - 2(\frac{4^m - (-1)^m}{4^m * 5})$ and generalized this to a graph with $n + 1$ nodes to $2 - 2(\frac{n^m - (-1)^m}{n^m * (n+1)})$. This has yet to be proven.

2. Another observation was made with regards to DSD values from 3, 4, 5, and 6-noded linear graphs. It is conjectured that the minimum non-zero DSD value for such a graph with $n + 1$ nodes is the same as the k value corresponding to the complete graph of $2n$ nodes. However, this has yet to be proven.

6 Acknowledgements

The author would like to thank her mentor Dr. Lenore Cowen from Tufts University for inspiring her to to pursue this project as well as for support and mentorship. She would also like to thank her fellow students Taylor Yeracaris and Joshua Tso for their contribution to this project as well as fellow students Yuelin Liu, Samuel Slate, Faith Ocitti, Rebecca Newman, Daniel Meyer, and Shari Sun for their support. She would also like to thank the CRA Committee on the Status of Women in Computing Research (CRA-W) for establishing this Distributed Research Experiences for Undergraduates (DREU) program which has given her the opportunity and support to participate in this research experience.

References

[1] Cao M, Zhang H, Park J, Daniels NM, Crovella ME, Cowen LJ, et al. (2013) Going the Distance for Protein Function Prediction: A New Dis-

tance Metric for Protein Interaction Networks. PLoS ONE 8(10): e76339. https://doi.org/10.1371/journal.pone.0076339

[2] Ruepp A, Zollner A, Maier D, Albermann K, Hani J, et al. (2004)The funcat, a functional annotation scheme for systematic classification of proteins from whole genomes. Nucleic acids research 32: 5539–5545.

[3] Stark C, Breitkreutz B, Reguly T, Boucher L, Breitkreutz A, et al. (2006) Biogrid: a general repository for interaction datasets. Nucleic Acids Res 34: D535–D539.

[4] Cao M, Pietras CM, Feng X, Doroschak KJ, Schaffner T, Park J, Zhang H, Cowen LJ, and Hescott B. (2014) New directions for diffusion-based network prediction of protein function: incorporating pathways with confidence. Bioinformatics 30: i219–i227.