The Interaction-Interaction Model for Disease Protein Discovery

Ken Cheng*

December 10, 2017

Abstract

Network medicine, the field of using biological networks to develop insight into disease and medicine, is a budding area of research where current efforts have been rewarded with only a middling level of success. Various network models have been proposed to help advise and direct medical research; this paper explores the possible utility of the "interaction-interaction" (II) network model by adapting some current protein-protein-interaction (PPI) algorithms and comparing the performance amongst algorithms over the II and PPI networks.

Introduction

Proteins are an essential building block for life and the complex interactions among proteins are important, when not crucial, to maintaining homeostasis. If a protein exhibits abnormal behavior, it may cause other proteins to also exhibit abnormal behavior, which may cascade into a "snowball" effect, resulting in undesired symptoms in an organism. Thus, when we, as humans, want to find a cure for a particular disease, it is beneficial to discern and study which proteins, if any, may be the causing the disease. However, the human genome contains over 20,000 protein-encoding genes (where gene can produce one or more variants of a protein), and scientists are just currently unable to study all the complex interactions of all the proteins for a particular disease. We therefore turn to computational models to help select good protein candidates to focus on.

Prior Work

In Barabasi, Gulbahce, and Loscalzo [1], the authors give a summary of the state of "network medicine," the field of using networks to help study diseases, and demonstrate the difficulty of the problem by citing various examples and approaches that were met with, at best, mild success. At the time of their writing, Barabasi et al. state that algorithms to discover possible affected proteins given a set of currently affected ("*seed*") proteins can be generally classified into three types:

1. *linkage-based*, where direct neighbors of affected proteins are marked as suspects,
2. *modularity, or pathway-based*, where proteins that are active in the same biological pathway are marked as suspects, and

*Email: kencheng@cs.stanford.edu, SID: 05607007

3. *diffusion-based*, where proteins are marked as suspects if they happen to be well-connected to other known affected proteins (usually done using random walks on a protein-protein interaction network).

The authors point out that linkage-based algorithms tend to perform the worst because they can only consider local information (i.e. a node's susceptibility is dependent only on its neighbors), whereas diffusion-based algorithms inherently use the entire network topology to identify possible affected proteins.

Furthermore, Barabasi et al. also list out various network models that have been used to further research in various diseases:

1. *protein-protein interaction (PPI) networks*, where nodes are proteins and edges represent a direct interaction between two proteins,

2. *metabolic disease networks*, where nodes are diseases and edges represent a connection of the diseases through shared enzymes in a certain metabolic pathway

3. *phenotypic disease networks*, where nodes are diseases and edges represent a sufficient level of comorbidity (when two diseases occur at the same time).

Current research in this field seem to focus upon developing algorithms over PPI networks. Ghiassian, Menche, and Barabasi [2] developed an algorithm that calculated scored a protein's likelihood to participate in a disease by using the significance of the connections that protein had to other (diseased and non-diseased) proteins in the PPI network. Agrawal, Zitnik, and Leskovec [3] noted that current algorithms, which only considered a protein's direct neighbors in the PII, seemed to be fundamentally missing some information, and instead, proposed to look at algorithms that used the neighbors' neighbors.

Definition of the Problem

Let P be the set of all proteins involved in a particular biological system (this is typically the human interactome). A disease module for a particular disease D is a subset $M_D \subset P$ that contains the set of proteins that is known (or suspected) to be related to the disease. We would then like to develop an algorithm that computes the following:

Given a disease δ with disease module M_δ, which proteins $p \in P$ are also involved in D, if any?

Scientists can then use the result of the algorithm to focus on those proteins and expand our understanding of the disease in question.

Intuition of II Networks

Though Barabasi et al. pointed out various network models, network medicine research has been more or less focused upon algorithms over the PPI network. This is likely because, as Ghiassian et al. writes, "there is increasing evidence that proteins associated with a particular disease have distinct interactions" in the PPI network [2]. However, the PPI network may be too high of an abstraction, as we will illustrate in the following example.

Consider the PPI network in Figure 1a, where the shaded node A means that A is in the disease module of a hypothetical disease δ. By looking at graph, it may be tempting to point out B as the next best candidate to research for δ, since A directly interacts with B. However, depending on the

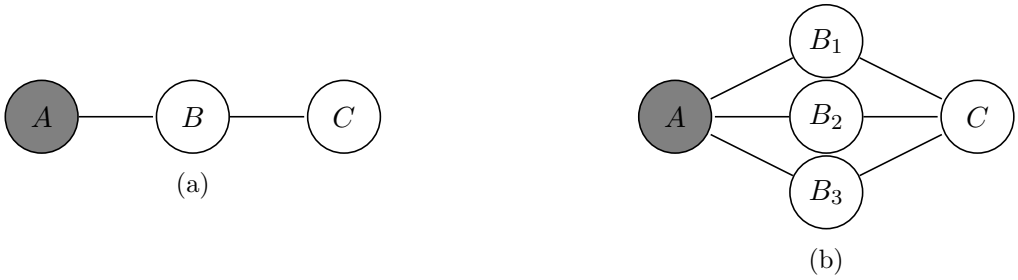


Figure 1: Hypothetical PPI networks that help illustrate the limitations of algorithms over PPI networks. Shaded nodes represent the disease module. (1a) Note that depending on the physical interactions, A can just as much of an impact on C as it can on B. (1b) C may be a good candidate to research, as opposed to any one of the B nodes.

interactions between A, B, and C, C may very well be the best candidate. One such hypothetical case could be that in normal operation, A and B form a protein complex AB until it is in the presence of C, at which point, B releases A and binds to C instead. Figure 2 illustrates this process.

Now, consider a mutation that causes A to bind tightly with B, making it difficult for C to bind with B and eject A. This may cause an abundance of C in the cell, which may in turn cause other symptoms. It may therefore be interesting to study protein C. C may become more interesting than other protein if there are many intermediary proteins in between it and a diseased protein, as shown in Figure 1b. In this case, we may expect that C is affected by A via one of the B proteins, but we don't know which of the B nodes may be affected.

Instead of the PPI network, we could use a network based on the relationships between the individual protein-protein interactions. Ideally, we would have a network whose nodes represent the interaction between a pair of proteins, and whose edges represent a direct cause-and-effect relationship. So for example, the PPI network in Figure 1a could be represented with either II networks in Figure 3a or Figure 3b, *depending on how the interactions affect each other.* In the first case, we can infer that interaction AB directly affects interaction BC, as in the previous example, and so, protein C may be interesting to study. In the second case, we can infer that the interaction AB does not directly affect BC, so protein C becomes less interesting to study. (Note that in the latter case, C may still be part of the disease, if the *disease* causes B to misbehave.)

The II network can therefore be more expressive than the PPI network, which may fill in the gap of information as suggested by Agrawal et al in [3].

(a) Protein complex AB forms naturally, until in the presence of C

(b) Binding of C to B causes A to unbind.

Figure 2: Hypothetical instance of the network in Figure 1a.

(a) (b)

Figure 3: Example II networks. Shaded nodes mean that the interaction is compromised (one of the proteins in the interaction is in the disease module). (3a) The interaction between A and B affects the interaction between B and C or vice versa (e.g. promotion or inhibition). (3b) The interaction between A and B does not affect the interaction between B and C, and vice versa (e.g. different binding sites).

Data and Limitations

Since we, as human beings, do not completely understand the human interactome, we do not have the true II or PPI network graph, and we can only build them incrementally through experiments. So even though the II network may be more expressive, it is also significantly harder to construct than the PPI network, for the following reasons:

- Assays to detect protein-protein interactions are easier to design and perform compared to assays to detect the cause-and-effect of two events.
- The number of possible interaction-interaction effects is much greater than the number of possible protein-protein interactions ($\Theta(|P|^4)$ vs. $\Theta(|P|^2)$).

Thus, each data point in the II network takes many more resources than the PPI network.

As a result, we do not have II network data to perform experiments on. Instead, we can generate a crude II network $G_{II} = (V_{II}, E_{II})$ from the current PPI network $G_{PPI} = (V_{PPI}, E_{PPI})$ via the following method:

$$V_{II} = \{\{i, j\} \mid \forall (i, j) \in E_{PPI}\}$$
$$E_{II} = \{(\{i, j\}, \{j, k\}) \mid \forall i, j, k \in V_{PPI} \text{ s.t. } (i, j) \in E_{PPI} \wedge (j, k) \in E_{PPI}\}$$

That is, we turn every edge in the PPI network into a node in the II network, and we put an edge between two nodes in the II network if the corresponding edges (interactions) in the PPI network share a node (protein) in common. Using the data at `http://snap.stanford.edu/pathways/bio-pathways-network.tar.gz`, we transform the PPI network of 20k nodes and 340k edges into an II network of 340k nodes and 61M edges. A summary of the networks can be seen in Table 1.

It is worth noting that since the II network is constructed from the PPI network, there is no additional information in this II network than the source PPI network. However, "natural" algorithms over the structure of the II network may prove to be more useful or insightful than natural algorithms over the structure of the PPI network.

Network	$\|V\|$	$\|E\|$	$E[k]$	$\|\{v \in V \mid k_v = 0\}\|$	k_{max}	Largest SCC ratio
PPI network	22552	342353	30.36	995	2131	0.9543
II network	342353	61034796	356.56	0	3828	0.9999

Table 1: Comparison of some properties of the PPI network and the II network. k here refers to node degree.

Algorithm 1 The linkage-based scoring algorithm. Every node gets a score depending on whether or not its neighbors are affected (according to the disease module), and the degree of the neighbor. A PPI node is affected if it appears in the disease module. An II node is affected if one of the proteins in its node is in the disease module.

```
function LinkageBasedScoreNode(Graph G, Node node, DiseaseModule dm)
    score ← 0
    for (node, nbr) ∈ G do
        if dm.IsAffected(nbr) then
            score ← score + (1 / Degree(nbr))
        end if
    end for
    return score
end function
```

Algorithms

To test the efficacy of restructuring the data as an II network, we ran similar algorithms over both the PPI network and the II network. In general, for every node, we computed a score, using the network and the disease module. We then use the scores to select a suspected protein. That protein is then added to the algorithm's model of the current disease module, so that it can be used to help determine other protein suspects.

For PPI networks, we just suggest the node (and therefore, protein) with the largest score.

For II networks, we aggregate the data. Recall that every node in the II network is a pair of proteins. To generate a score for a protein, we sum over the scores of all the nodes that the protein is a part of. Then, we select the protein with the largest score.

For scoring nodes, we used two different types of algorithms: a simple linkage-based algorithm that only checks if its neighbors are affected (see Algorithm 1), and a random-walk-with-teleport algorithm (see Algorithm 2) with 10,000 steps and a teleportation parameter of 0.5.

Algorithm 2 The random walk with teleport algorithm. The teleport set is the set of nodes that are affected (according to the disease module). Note that we compute the score of all the nodes at once. We used NUM-ITERATIONS = 10000 and TELEPORTATION-PARAMETER = 0.5.

```
function RandomWalkScores(Graph G, DiseaseModule dm)
    scores ← map()
    current-node ← randomly-select-affected-node(G, dm)
    for NUM-ITERATIONS steps do
        scores[current-node] ← scores[current-node] + 1
        if with-random-probability-of(TELEPORTATION-PARAMETER) then
            current-node ← randomly-select-affected-node(G, dm)
        else
            current-node ← randomly-select-one-neighbor(current-node)
        end if
    end for
    return scores
end function
```

Scoring Methodology

It is important to note that since the disease modules are determined by biological experiments, it is likely that the disease module data is incomplete. It is therefore important that our scoring mechanism must be resilient to our gaps in knowledge. For example, if an algorithm were able to determine every protein that participated in a disease, it may also propose proteins that do not appear in our currently-known disease modules. We cannot, therefore, penalize an algorithm that suggests proteins other than the ones we expect. Therefore, we used a scoring mechanism similar to that used by Agrawal et al. in [3], which is as follows.

Depending on the runtime of the algorithm, some selection of the 520 human-curated disease modules were randomly partitioned in to 10 sets. For each disease, for every set, that set was held out as a validation set, and the other 9 sets acted as the disease module input to the algorithm, which suggested proteins, one at a time, up to $k = 100$. We then calculated the recall-at-k, which is the fraction of proteins in the validation set that where in the first k suggestions by the algorithm. (A score of 1 means that within the first k suggestions, the algorithm was able to find all the proteins in the validation set. A score of 0 means that within the first k suggestions, the algorithm did not find any of the proteins in the validation set.) The average recall-at-k was then computed as the average over all selected diseases, and plotted against k. Note that by letting k be a large value, recall-at-k prevents penalizing "perfect" algorithms, since those algorithms have many "chances" to provide our expected proteins.

Whether iterating over the PPI nodes or over the II nodes, the algorithms using the II network are extremely computationally expensive. As a result, those algorithms use a randomly-sampled set of 10 diseases to score. To fairly compare PPI network algorithms and II network algorithms, the PPI algorithms are run on the same sample, but to get an idea of the overall performance, the PPI network algorithms were also run over the entire data set.

Results

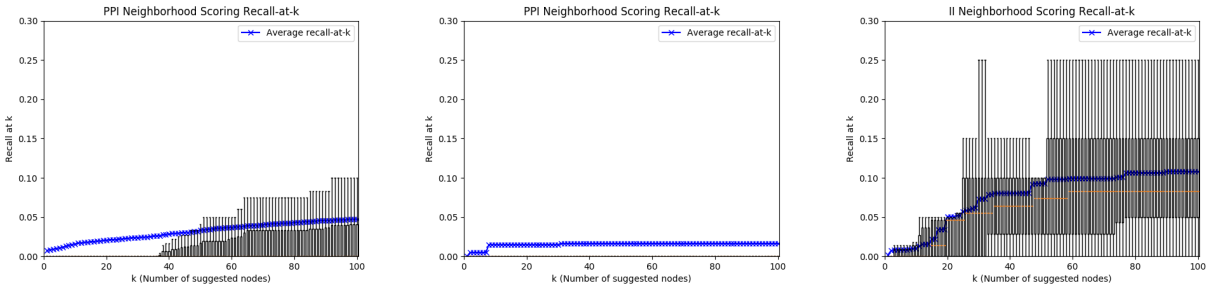
A summary of the results can be seen in Figure 4 and Table 2. Note that with a sample size of 10, the results for the II network algorithm cannot be confidently used in a statistical manner to empirically validate or deny the utility of the II network; nonetheless, we can still make some preliminary judgements. The linkage-based algorithm seems to perform better over the II network as opposed to the PPI network, but the random-walk algorithm seems to perform worse over the II network as opposed to the II network.

One hypothesis that could explain this behavior is that the II network is able to capture some extra "local structure" information about the protein, which benefits the linkage-based algorithm, since it only looks at direct neighbors. In contrast, this information could be hampering the random-walk algorithm, since the II network would cause the algorithm to focus too much on the interactions involving a diseased protein. In that case, we might need to tweak the random walk parameters in

Recall-at-100

	Linkage-based	Random-Walk
PPI Network (full)	0.0472	0.1058
PPI Network (sampled)	0.0164	0.1261
II Network (sampled)	0.1081	0.0292

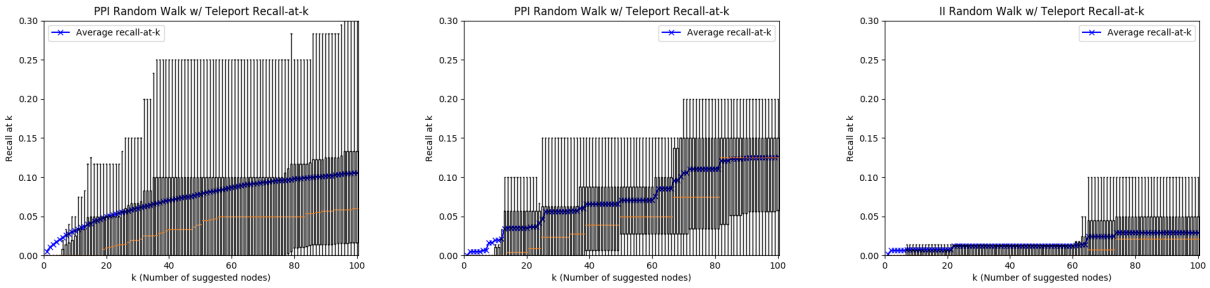
Table 2: Recall-at-100 for the different networks and strategies.



(a) Recall-at-k for linkage-based algorithm over PPI

(b) Recall-at-k for linkage-based algorithm over PPI (sampled)

(c) Recall-at-k for linkage-based algorithm over II (sampled)



(d) Recall-at-k for random-walk algorithm over PPI

(e) Recall-at-k for random-walk algorithm over PPI (sampled)

(f) Recall-at-k for random-walk algorithm over II (sampled)

Figure 4: The average recall-at-k for each algorithm and graph are plotted against k, along with a box-and-whisker plot of the distribution of recalls-at-k at each k up to 100. Outliers are considered for the average line, but are not plotted for the box-and-whisker plot. As a result, empty box-and-whisker plots mean that the upper quartile starts at 0. The y-scales for all the graphs range from 0 to 0.3.

Runtime

	Linkage-based	Random-Walk
PPI Network (over 10 diseases)	80 seconds	15 minutes
PPI Network	61 minutes	13.7 hours
II Network (over 10 diseases)	30 hours	13.6 hours
II Network (extrapolated)	65 days	29.4 days

Table 3: Runtime for the different networks and strategies.

order to give more weight to exploration so that it is more likely to traverse away from the set of diseased interactions.

Additionally, the running times of the different algorithms are shown in Table 3, in addition to the extrapolated time it would take to run the II network algorithms on the full suite of diseases.

Conclusion

In conclusion, we find mixed performance results on using the II network over the PPI network. It is possible that the II network might contain some sort of local-structure information between the proteins that may help certain algorithms perform better. However, the algorithms over the

II network is computationally expensive, and are therefore extremely difficult to train parameters over. If II networks are to be investigated further, we will require more optimized code in order to perform more sophisticated experiments, such as figuring out which diseases seem to be better explained by each network and why.

Bibliography

[1] Barabasi, A.-L., Gulbahce, N., Loscalzo, J. (2011). Network Medicine: A Network-based Approach to Human Disease. Nature Reviews. Genetics, 12(1), 56?68. http://doi.org/10.1038/nrg2918

[2] Ghiassian SD, Menche J, Barabsi AL (2015) A DIseAse MOdule Detection (DIAMOnD) Algorithm Derived from a Systematic Analysis of Connectivity Patterns of Disease Proteins in the Human Interactome. PLOS Computational Biology 11(4): e1004120. https://doi.org/10.1371/journal.pcbi.1004120

[3] Agrawal, M., Zitnik, M., Leskovec, J. (TBP). Large-scale Analysis of Disease Pathways in the Human Interactome. Pacific Symposium on Biocomputing.