Network Analysis Algorithms for Disease Gene Prioritization

Charlie Broadbent, Sam Chen, Kate Finstuen-Magro, Chris Padilla, Oscar Smith, Andy Younkins

March 11, 2020

Abstract

The compilation of protein-protein interaction (PPI) data and the application of network traversal algorithms has been shown to be an effective approach to disease gene prioritization. The effectiveness of this approach is not as certain on current, larger, more updated PPI networks. In addition, performance may differ based on the type of disease being analyzed. We analyzed the effectiveness of three different network analysis algorithms: random walk with restart, PageRank with Priors, and diffusion kernel. With modern PPI network data from the STRING database, we found promise with random walk with restart and PageRank and their use for disease gene prioritization, though their complete accuracy and performance based on disease type requires further study.

1 Introduction

The identification of genes associated with various diseases is useful to biologists and researchers in improving the quality of medical treatment. Recent advances in genetic sequencing has allowed for more information than ever on the genetic origin of diseases like cancer and diabetes; however, the influx of data, while providing a wealth of information, has made deciding which genes to analyze much more important.

One recently studied approach to the problem of disease gene prioritization is to use biological data encoded in a protein-protein interaction (PPI) network. A PPI network is an undirected graph in which each node corresponds to a protein and an edge between two nodes exists if the corresponding proteins interact with each other. Note here that genes encode for proteins, so we will refer to them interchangeably. Furthermore, a protein interaction represents physical contact or binding between proteins.

Through disease gene prioritization, we can analyze the part of the network that contains genes already known to be associated to a specific disease and rank them in importance. Additionally, this allows for the identification of previously not associated genes of importance. Thus, disease gene prioritization is a valuable tool in helping biologists narrow down candidate disease genes for further study. Recent studies [11][6][8] have used different network analysis algorithms to traverse a given PPI network and efficiently rank disease genes given a starting set of known disease genes. However, since these studies, more research in protein-protein-interaction has led to an increase in interaction data. The effect of this growth and change in PPI data on network analysis algorithms is unclear, so our goal is to analyze their effectiveness on larger, more updated PPI networks.

Several PPI databases exist, and they vary in size and interaction composition. Primary databases such as BioGRID contain only experimentally proven interactions, and meta-databases like APID contain a compilation of primary databases [7]. Predictive databases such as STRING consist of experimental as well as predictive interaction data. [7].

We investigated the accuracy and efficiency of three algorithms, random walk with restart [11], PageRank with Priors [24], and diffusion kernel [12], and their ability to rank disease genes. Although these algorithms have been implemented on PPI networks previously, our main objective was to study their performance on current, more updated networks. We used the STRING [23] database in order to rank the genes of three different kinds of diseases: lymphoma, ischaemic stroke, and endometriosis. We found that random walk with restart and PageRank perform significantly better than diffusion kernel, and that they show promise for disease gene prediction. However, the true importance of the highly ranked genes among the three diseases remains uncertain,

and requires further biological research.

2 Materials and methods

We will first describe three algorithmic approaches for ranking genes: random walk with restart (RWR) [11], PageRank with Priors (PR) [24], and diffusion kernel (DK) [12]. These methods were chosen because of their ubiquity in the existing literature [11][6][8] and our interest in investigating the performance of network analysis algorithms on large, noisy networks, such as a PPI network. Then, we will describe our data sets. These include our PPI network as well as sets of known disease genes for endometriosis, ischaemic stroke, and lymphoma.

2.1 Algorithms

The three algorithms can produce similar results given the correctly tweaked parameters. Despite this, they are all distinct in their computational techniques. Thus, it is worthwhile to analyze each of their performances and identify how their differing functionalities can lead to different or similar results.

2.1.1 Random walk with restart

Random Walk with Restart (RWR) is an algorithm that computes the closeness between any two nodes in a graph. The computation is performed by iterating the transitions of a walker traveling throughout a graph by randomly selecting adjacent nodes to travel to [11]. At each state at a certain step, there is a possibility to restart from any node in the set of root nodes rather than continuing the walk from the current node. We define RWR as

$$\mathbf{p}^{t+1} = (1-\beta)W\mathbf{p}^t + \beta\mathbf{p}^0$$

where

t is the step.

β is the probability of restart.

W is the normalized adjacency matrix of the graph, with $W[i][j]$ being the probability of moving from node j to node i.

$\mathbf{p}^t$ is the vector in which the i-th element is the probability of being at node i at step t.

$\mathbf{p}^0$ is the initial probability vector, which will be generated by assigning all known genes associated with the disease of interest an equal probability of being a start node, which sum to 1. All other nodes will be assigned a 0.

Note that the closer β is to 1, the less frequently the walker leaves the current node. For example, if $\beta = 1$, the walker never leaves the start nodes and the steady state vector will be the same as the initial state vector. Candidate genes will be ranked according to their values in the steady-state probability vector, where higher values indicate a higher likelihood of disease association. This final vector is obtained by running the algorithm until the difference between two steps in the walk $\mathbf{p^t}$ and $\mathbf{p^{t+1}}$ is less than 10^{-6} [11].

2.1.2 PageRank with Priors

PageRank (PR) was originally developed in the context of a web page connected by links to other web pages [2]. PageRank, in its initial form, ranked all nodes globally. However, since we are analyzing genes in relation to an initial disease gene set, we used PageRank with Priors, as described by Smyth and White [24]. Like PR, PageRank with Priors uses the number and quality of links to a page in order to give a numeric estimate of the importance of a website. However, instead of having a uniform probability of restarting at any node in the graph, PageRank with Priors will only restart at a set of defined root nodes, which in our case correspond to the known disease genes. Note that the set of root nodes is assigned a probability distribution, known as a prior bias vector, where only the root nodes have non-zero probabilities and all probabilities sum to one. Then we define the rank of a particular node iteratively as:

$$P^{t+1}(u) = (1-\beta)\sum_{v\in B}\frac{P^t(v)}{L(v)} + \beta p_u$$

where

β is the probability of restarting at a root node. Note that root node will denote a known disease gene for our purposes.

$P^{t+1}(u)$ is the rank of a particular node u at time-step $t+1$. Then $P^t(u)$ is the rank of a particular node u at time-step t.

B is the set of all nodes that have links to node u

$L(v)$ is the number of links going out of a node v.

p_u is the prior bias of of node u

Like RWR, the parameter for PR, β, governs the probability that a walk on the graph will restart. If $\beta = 1$, then the PR of the node u would be the same

as its prior bias. In other words, $\beta = 1$ would prevent a walk from leaving the prior bias nodes. On the other hand, if $\beta = 0$, the prior bias of a node is never factored into the calculation of a node's ranking.

Since we can define the rank of each node u in terms of a system of equations, we get the following equation:

$$\mathbf{p}^{t+1} = (1-\beta)W\mathbf{p}^t + \beta\mathbf{p}$$

where $\mathbf{p}^{t+1}$ and $\mathbf{p}^t$ are vectors of rankings at timestep $t+1$ and t respectively, W is the normalized adjacency matrix of the graph, and $\mathbf{p}$ is the prior bias vector. The starting probability vector, $\mathbf{p}^0$, has $\frac{1}{|R|}$ - where $|R|$ is the size of the known disease gene set for each known disease gene node and 0 for all other nodes.

To generate a prior bias vector, we compared our disease gene sets with disease gene sets from MalaCards [18]. If a disease gene was in both our original disease gene set and the MalaCards set, we assigned it an initial value of two. If a disease gene was in both our original disease gene set and the MalaCards set and it was rated as an elite gene by MalaCards, we assigned it an initial value of three. All other genes in our original disease gene set were assigned an initial value of one. Then, the prior bias of a given disease gene u is

$$p_u = \frac{i(u)}{\sum_{v \in R} i(v)}$$

where $i(u)$ is the importance of a given node u and R denotes the original set of known disease genes. Notice that RWR and PageRank with Priors are equivalent if the prior bias is the same for all root nodes.

Like RWR, the steady-state probability vector was obtained by repeatedly calculating $\mathbf{p}^{t+1}$ until the Euclidean distance between $\mathbf{p}^t$ and $\mathbf{p}^{t+1}$ was less than 10^{-6}.

Note that the steady-state probability vector for both RWR and PageRank with Priors can also be found by multiplying $\beta\mathbf{p}$ by $(I-(1-\beta)W)^{-1}$ where I is the identity matrix. However, calculating the vector iteratively is comparable in runtime and allows us to sidestep issues of matrix invertibility.

2.1.3 Diffusion Kernel

Diffusion kernel is a method that seeks to mimic heat flow across a surface. In the case of disease gene prioritization, the diffusion kernel method ranks genes by computing the matrix exponential of a Laplacian matrix. For any graph, represented by a matrix, the Laplacian matrix L is calculated as:

$$L = D - A$$

where D is the diagonal matrix containing each node's degree, and A is the adjacency matrix of the graph. The diffusion kernel K is obtained by calculating the matrix exponential of L, multiplied by a magnitude of diffusion β, a variable parameter:

$$K = e^{(-\beta L)}$$

Similar to RWR, β can be adjusted to control the extent heat is diffused from disease genes, where higher values correspond to more diffusion. To calculate the matrix exponential, we used eigenvector decomposition:

$$L = Q\Lambda Q^{-1}$$

where

The columns of Q are the eigenvectors of L.

Λ is a diagonal matrix, whose entries are the corresponding eigenvalues of L.

Since our graph is symmetric, the eigenvalues and eigenvectors are real. Since this decomposition is fairly common, there is a highly optimized method from LAPACK which NumPy [17] calls to compute this. Furthermore, Q is a normal matrix, so $Q^{-1} = Q^T$.

To make use of this decomposition, first note that multiplying a matrix by a constant $(-\beta)$ only changes the eigenvalues, so $-\beta L = Q(-\beta\Lambda)Q^{-1}$. Now if we expand the exponential by

$$e^x = \Sigma_{n=0}^{\infty} \frac{x^n}{n!}$$

Now note that $(YXY^{-1})^n = YX^nY^{-1}$ for any matrices X, Y, and perform the following simplification:

$$\begin{aligned}
K &= e^{-\beta L} \\
&= e^{Q(-\beta\Lambda)Q^{-1}} \\
&= \sum_{n=0}^{\infty} \frac{\left(Q(-\beta\cdot\Lambda)Q^{-1}\right)^n}{n!} \\
&= \sum_{n=0}^{\infty} \frac{Q(-\beta\cdot\Lambda)^n Q^{-1}}{n!} \\
&= Q\sum_{n=0}^{\infty} \frac{(-\beta\cdot\Lambda)^n}{n!} Q^{-1} \\
&= Qe^{-\beta\Lambda}Q^T
\end{aligned}$$

This is a significant simplification since $-\beta\Lambda$ is diagonal so it's matrix exponential is simply the element-wise exponential.

To calculate the scores of disease genes, we multiply our starting gene vector $\mathbf{v^0}$, and the kernel K, which results in a vector of scores $\mathbf{v^s}$, where each element is a score that corresponds to a gene. This multiplication is performed during the computation of K:

$$v^s = Kv^0 = Qe^{-\beta\Lambda}Q^T v^0$$

Since matrix multiplication is associative, this can be done as a series of matrix-vector multiplications significantly reducing the runtime. The resulting vector $\mathbf{v^s}$ is the desired list of gene scores.

2.2 Protein-Protein Interaction Network

We used the STRING database and its compilation of high throughput experimental data of protein-protein interactions from other databases in order to construct our PPI networks. STRING provides confidence scores between interactions based on various types of evidence of known and predicted interactions [23]. We chose to use STRING for its extensive coverage of data and evidence, as well as its ease of use.

The constructed network consisted of 19,354 nodes (proteins) and 11.7 million edges (interactions). In general, this graph is highly connected: the mean degree of the nodes is 608, the median 469, and the maximum 7645. However, the standard deviation of degrees is around 529, which suggests that the a wide degree distribution. Our network consists of interactions of every source type that is provided by STRING. These include: fusion, neighborhood, co-occurrence, experimental, text-mining, database, and co-expression [23].

2.3 Disease genes

The sets of genes we used for each of the three diseases—endometriosis, ischaemic stroke, and lymphoma—were taken from a training set used by Chen et al. [6]. Endometriosis is a disorder in which uterine tissue grows outside of the uterus, ischaemic stroke occurs when an artery to the brain is blocked, and lymphoma is a type of cancer that occurs in the lymph nodes.

We chose these diseases to investigate the efficacy of each algorithm on different types of diseases. In addition, we chose to use the same sets as Chen et al. [6], as the sets had already been shown to work in their research which allow us to better verify the correctness of our algorithms rather than using a different, less studied set of disease genes. The endometriosis gene set contains 43 genes, the lymphoma gene set contains 42 genes, and the ischaemic gene set contains 44 genes. The distribution of degrees in each gene set can be seen in Figure 3.

2.4 Validation and Performance

To evaluate the three algorithms, we used leave-one-out cross validation for parameter tuning, and constructed AUROC (Area Under the Receiving Operator Characteristics) curves to analyze their effectiveness at ranking and classifying disease genes.

Leave-one-out cross validation was performed by removing a single gene from each gene set, running an algorithm on that reduced set, and checking that the removed gene was predicted as a disease gene. A gene was considered to be predicted if it was ranked within the top 150 genes. One hundred fifty was chosen arbitrarily, but it gives some leeway in prediction given the size of the disease gene sets. In general, choosing this threshold is difficult, as it requires some knowledge of how many additional genes are expected to be predicted. This was done for each disease gene in the set, so for a disease gene set of size n, an algorithm would be run n times, where each time a different gene is removed. This process was repeated for each algorithm for each disease.

AUROC curves were generated using the optimal parameter as determined from leave-one-out by comparing predicted genes to a ground truth disease gene set taken from MalaCards [18]. A threshold was varied from 1 to the size of the disease gene set and true-positive-rate and false-positive-rate were calculated based on an algorithm's correctness in predicting the top k ranked genes, where k is the threshold, that are in the ground truth set. A gene that is above the threshold and is in the ground truth set is interpreted as a true-positive, and a gene that is above the threshold, but not in the ground truth set, is interpreted as a false-positive. A false-negative is when a gene is below the threshold and in the ground truth set. A true negative is when a gene is below the threshold and not in the ground truth set.

2.5 Software and Hardware

All software and computations were programmed and ran in Python3.7 with the help of NetworkX for creating network graphs and NumPy for the bulk of the linear algebra operations. A crucial library we used is pickle, which allows for significantly reduced run

times by allowing us to cache some expensive computations to disk and reuse them. For example, we cached the eigendecomposition of L in the diffusion kernel, allowing us to do multiple runs of different gene sets without having to recompute the eigendecomposition for every run.

All algorithms were run on a 32-core server of about 100GB of RAM. Each algorithm takes up about 14GB of RAM, and requires around 12 minutes for pre-processing of the network on an initial run, followed by 1 to 2 minutes of actual computation.

3 Results

The optimal variable parameter β, as defined in each algorithm, was determined to be 0.4 as each algorithm most often predicted removed genes with that given value, as seen in Figure 1. The algorithms performed best on ischaemic stroke, followed by endometriosis, then lymphoma. Diffusion kernel was unable to consistently predict any removed gene on any of the diseases. Furthermore, Figure 1 indicates that PR is slightly better at predicting unknown information with leave one out cross validation than RWR for endometriosis and ischaemic stroke. However, RWR slightly outperforms PR on the lymphoma data set given a parameter of 0.4.

As seen in Figure 2, RWR and PR showed significantly more success than DK in both leave-one-out and AUROC. For this reason, DK's poor performance is discussed further in our conclusions, and these results will instead mostly highlight RWR and PR.

While leave-one-out cross validation measures the ability of each algorithm to predict known information, ROC curve analysis more accurately simulates the ability of each algorithm to predict disease genes given a set of starting genes.The ROC curves (Figure 2) also show a similar distinction between algorithms, where RWR and PR significantly outperformed DK, which barely performed better than a random classifier. In addition, PR and RWR have almost identical ROC curves for all three data sets. Note that this differs from our results from leave one out - where PR outperformed RWR on the endometriosis and lymphoma - suggesting that while PR was more successful at re-predicting known information, RWR and PR perform almost identically when simulating the process of ranking disease genes. Furthermore, results from the ROC curves generated by each algorithm differed from leave-one-out, as the area under the ROC curve was the highest for endometriosis and lowest for ischaemic stroke. Recall that the percentage of left out genes correctly predicted was highest for ischaemic stroke and lowest for lymphoma across all algorithms.

The difference in performance between DK and the other two algorithms is sufficient enough to exclude the genes it predicted from our results. The top five genes predicted by PR and RWR for each disease are shown in Table 1. Notice that the similarities between the performance of PR and RWR that was seen in their respective ROC curves is reflected in the top five genes found for both algorithms as there is significant overlap between the rankings generated for each algorithm.

4 Discussion

Perhaps the biggest point of interest from the results is the poor performance of DK. Our most likely explanation is that due to the large number of interactions contained in the network, diffusion does not occur as we anticipated because of noise, as well as the presence of nodes with degrees in the thousands. As such, we will focus on the top ranked genes for only PR and RWR. Despite DK's unexpectedly poor performance, RWR and PR's results are still worthwhile in discussing. Ultimately, the genes we predicted for each disease would need to be extensively researched and verified by biologists, but there are studies that show they may be likely candidate genes.

4.1 Effects of network properties on algorithmic performance

First, we will consider the performance of DK in more detail. One possible explanation for DK's performance is that the network was too highly connected to allow for a meaningful probability distribution to be computed. On our network with 12 million interactions and 20,000 nodes, with some nodes connected to up to 5,000 neighbor nodes (Figure 3). Diffusion kernel was initially designed to model heat flow through more sparsely connected networks, so it is not surprising that it performed poorly on our graph with a highly connected structure. The disease gene sets we used were collectively connected to approximately 10,000 other nodes. Thus, heat will diffuse equally across about half of the nodes in the network, causing most genes to have a very similar probability in the final ranking. Thus, it was difficult to determine which genes had the highest rankings and could be considered disease-related. Interestingly, DK worked well in Kohler et al. when run

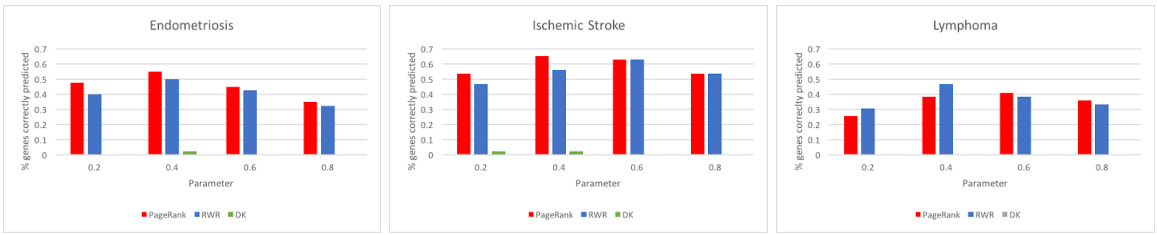


Figure 1: Leave one out cross validation results for endometriosis, ischaemic stroke, and lymphoma (left to right). The y-axis is percentage of genes correctly predicted while the x-axis represents the value of the parameter used for each algorithm. Note that the values tested were $0.2, 0.4, 0.6$ and 0.8.

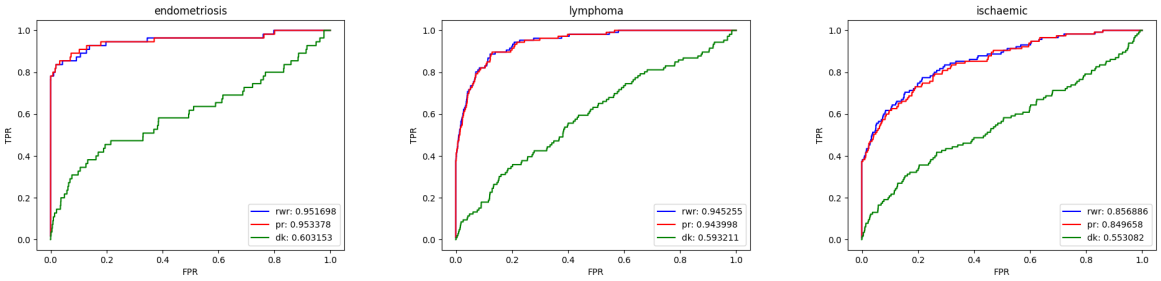


Figure 2: ROC Curves for endometriosis, lymphoma, and ischaemic stroke (from left to right) using a parameter value of 0.4 for each algorithm.

on the STRING human network.

However, at the time Kohler et al. used STRING, the human network had only 258,314 experimental interactions, significantly fewer than our 12 million. Fewer interactions may allow DK to produce a final probability vector with more variability. This difference in connectivity would explain why DK performed well in Kohler et al. but not on our network. However, the connectivity of the graph is only one possible explanation of the difference in performance of DK. It is likely that there are other factors affecting the performance that could be explored in further research.

The shape and connectivity of our network had other effects on our results. As seen in the AUROC curves, the algorithms performed the worst on ischaemic stroke. We believe this difference in gene prediction is due to an underlying structure of the disease genes within the network. Visualizing this structure is difficult due to the high degree of some of the genes. However, we assert that looking at the structure of the disease gene sets without text-mining interactions can provide additional information that is informative to the differences in algorithmic performance.

Figure 4 shows the STRING representations of the known disease genes for each disease. These representations reveal that ischemic stroke has a fundamentally different structure and connectivity than endometriosis and lymphoma, which have similar shapes. These differences in connectivity would explain the better performance of the algorithms in leave-one-out cross validation on ischemic stroke than the other two diseases. That is, the highly connected nature of ischemic stroke as compared to the long minimally connected tails of endometriosis and lymphoma will allow the algorithms to better predict a known disease gene highly in the rankings because there are more connections to each disease gene.

We see a different pattern of results for AUROC because it is not measuring just the connectivity of the known disease genes, but also an additional set of known genes that are not shown in the figures below. We look at the representations of the disease gene sets without text-mining interactions because we believe this gives a better representation of the true protein-protein interactions. Text-mining interactions are the lowest confidence interactions in STRING, so ignoring them when trying to determine if the nature of the protein-protein interactions in a disease is a valid method.

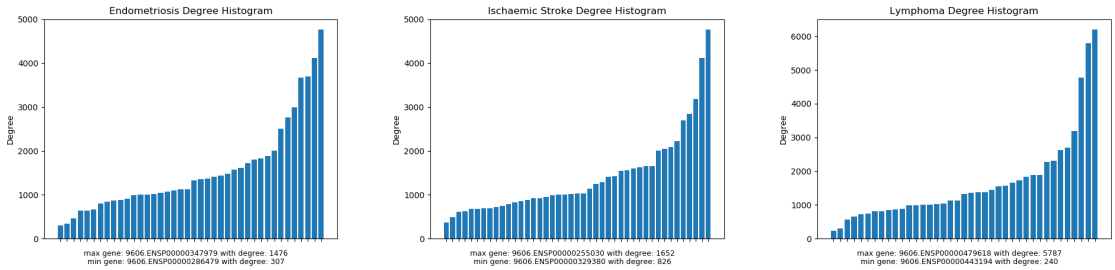


Figure 3: Degree histograms for each known disease gene set - endometriosis, ischaemic stroke, and lymphoma from left to right. Note that the y-axis is degree while each value on the x-axis corresponds to a known disease gene.

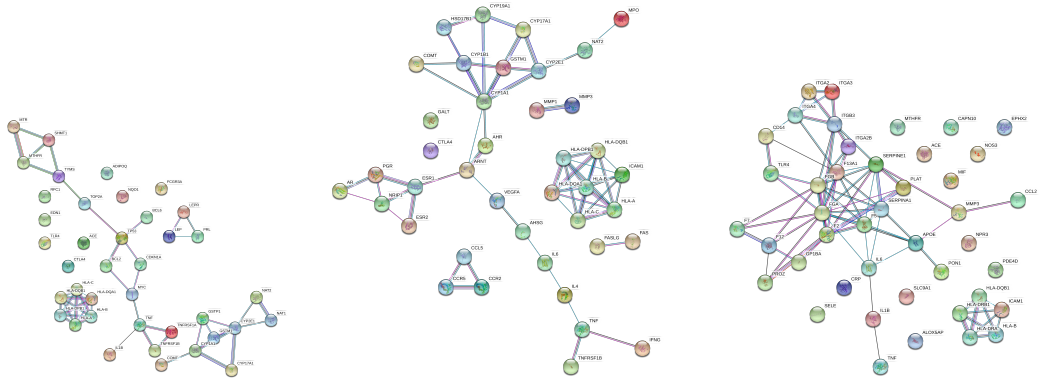


Figure 4: STRING graph of disease genes without text-mining interactions for lymphoma, endometriosis, and ischemic stroke, from left to right.

4.2 Candidate genes for future study

In order to better understand the performance of PR and RWR for each disease gene set, we investigated the top five genes that were not in the original disease gene set found by both algorithms. We did not consider the top five genes found by DK because of its poor performance.

4.2.1 Endometriosis

From Table 1, we can see that both PR and RWR had CYP2B6 as the most highly ranked gene that was not already in the known disease gene set. This gene corresponds to an enzyme that is involved in drug metabolism and synthesis of cholesterols and lipids. While there are no papers directly linking CYP2B6 to endometriosis, the importance of synthesizing lipids connects CYP2B6 to several other genes in the set of known endometriosis genes [19].

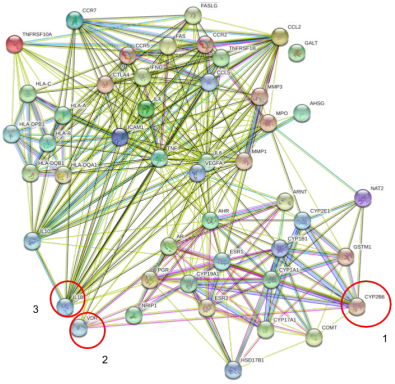
Furthermore, both PR and RWR ranked VDR and IL1B within the top five disease gene candidates for endometriosis. VDR encodes for a vitamin D receptor while IL1B encodes for a protein that is important in immune system response and inflammation. Although VDR is not directly linked to endometriosis, it is known to play an important role in reproductive health [5]. On the other hand, IL1B has an association with endometriosis but its role in the immune system also implicates the gene in around 300 other diseases as well [19].

4.2.2 Ischaemic Stroke

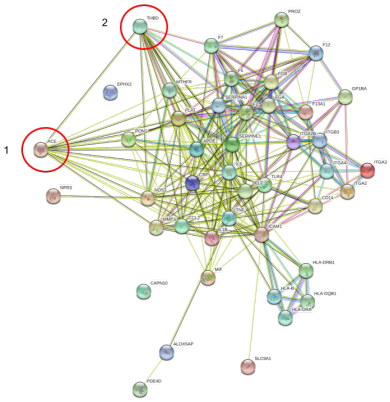
For ischaemic stroke, we again observed that the top ranked gene, THBD, was the same between PR and RWR. THBD is involved with processes related to blood clotting. Besides THBD, PageRank and RWR also both ranked Angiostensin converting enzyme within their top five disease gene candidates. This protein helps control blood pressures. Both Angiostensin converting enzyme and THBD are known to be related to ischaemic stroke [19]. Note that the predicted disease genes all correspond to processes involved with blood pressure and blood coagulation

Endometriosis		Lymphoma		Ischaemic Stroke	
RWR	**PageRank with Priors**	**RWR**	**PageRank with Priors**	**RWR**	**PageRank with Priors**
CYP2B6	CYP2B6	CYP2B6	CYP2B6	THBD	THBD
TNFRSF10A	VDR	Angiostensin converting enzyme	APOE	SELP	Angiostensin converting enzyme
VDR	IL1B	MPO	IL6	Angiostensin converting enzyme	CST3
IL1B	CCL2	APOE	MPO	VCAM1	F9
CCR7	IL10	ICAM1	TNFRSF10A	PROCR	SERPINB2

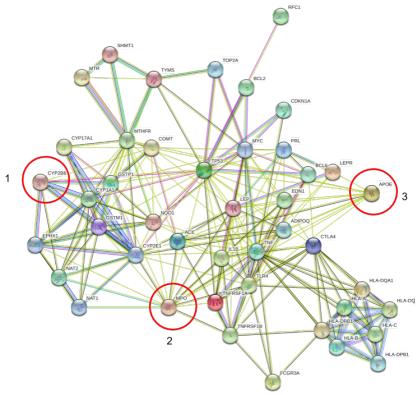
Table 1: The top ranked genes for RWR and PageRank with Priors such that they were not in the original gene set. These rankings were found when running RWR and PageRank with Priors with an $r - 0.4$ and $\beta = 0.4$ respectively.



(a) Endometriosis subgraph. Predicted genes are labeled as follows: 1 - CYP2B6, 2 - VDR, 3 - IL1B.



(b) Ischaemic stroke subgraph. Predicted genes are labeled as follows: 1 - THBD, 2 - Angiostensin converting enzyme



(c) Lymphoma subgraph. Predicted genes are labeled as follows: 1 - CYP2B6, 2 - MPO, 3 - APOE.

Figure 5: The induced subgraphs of the known endometriosis, lymphoma, and ischaemic stroke disease genes. Circled genes were ranked within the top five genes for both PageRank and RWR.

which is consistent with our expectations since ischaemic stroke is caused by a blocked artery in the

brain.

4.2.3 Lymphoma

Again, CYP2B6 was the top rated gene for both PR and RWR. Although it has a slight association with other types of cancer such as breast cancer and leukemia, there is not yet any association found with lymphoma [19]. In addition, APOE and MPO were both within the top five ranked genes for RWR and PR. They are involved in fat storage and blood function respectively. While APOE is not associated with lymphoma, MPO has a slight association with lymphoma [19].

While both algorithms highly ranked genes that had some association with the diseases, many of the genes found were highly involved in several different biological processes and therefore had strong associations with several diseases. For instance, IL1B is heavily involved in immune system response and may be more likely to be highly connected to the known disease gene set and involved in several different diseases.

5 Conclusions

Although the highly ranked genes are consistent with previous studies on the three diseases as well as what is currently known about the genes, there is no guarantee of their importance until they are verified by biologists. Additionally, until these results can be verified, the effectiveness of the algorithms cannot be stated with complete certainty.

Despite this, there are still several clear improvements that can be made in future studies and network analyses. The first is the construction of a smaller, less connected network. In our results it was clear that noise played a large role in affecting the results, especially for DK. A network should consist of interactions that have been experimentally verified, or predicted, such that interactions from other sources such as text mining, are not accounted for. This will result in a more accurate network that helps prevent noise from affecting any results.

Another clear problem was the existence of hub genes, or genes of extremely high degree. These genes likely have very important functions since they interact with such a large number of other genes. This is presumably a product of ascertainment bias, meaning these important genes are studied more than others, and thus there exists more interaction data for them. The problem arises when they are present in computation, and can skew the ranks of closely related genes. Again, this is especially problematic for DK.

More analyses also must be done on many different types of diseases. This would allow for more insight on whether specific types of diseases perform differently on different algorithms, as well as whether a PPI network analysis approach is even viable in identifying genes diseases.

5.1 Software

All software and data sets used in this paper are available for download and free use here: https://github.com/oscardssmith/Disease-Gene-Network-Analysis.git

6 Acknowledgements

We would like to thank our advisor, Layla Oesper, for her guidance. Additionally, we would like to thank Mike Tie for helping us install the proper software on our server and with server issues in general. Finally, we would like to thank Dr. Max Leiserson from University of Maryland for talking to us about PPI networks.

References

[1] Aerts, S. et al. Gene prioritization through genomic data fusion. *Nature Biotechnology*; New York 24, 537–44 (2006).

[2] Brin, S., & Page, L. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, (1998).

[3] Calderone, A., Castagnoli, L. & Cesareni, G. mentha: a resource for browsing integrated protein-interaction networks. *Nat Methods* 10, 690–691 (2013).

[4] Cao, M. et al. New directions for diffusion-based network prediction of protein function: incorporating pathways with confidence. *Bioinformatics* 30, i219–i227 (2014).

[5] Cermisoni, G. C., Alteri, A., Corti, L., Rabellotti, E., Papaleo, E., Viganò, P., and Sanchez, A. M. (2018). Vitamin D and Endometrium: A Systematic Review of a Neglected Area of Research. *International Journal of Molecular Sciences*, 19(8), 2320. https://doi.org/10.3390/ijms19082320

[6] Chen, J., Aronow, B. J. & Jegga, A. G. Disease candidate gene identification and prioritization using protein interaction networks. *BMC Bioinformatics* 10, 73 (2009).

[7] De Las Rivas, J. & Fontanillo, C. Protein–Protein Interactions Essentials: Key Concepts to Building and Analyzing Interactome Networks. *PLoS Comput Biol* 6, (2010).

[8] Guney, E. & Oliva, B. Analysis of the robustness of network-based disease-gene prioritization methods reveals redundancy in the human interactome and functional diversity of disease-genes. *PloS one*, 9, 4 (2014). https://doi.org/10.1371/journal.pone.0094686

[9] Iván, G. & Grolmusz, V. When the Web meets the cell: using personalized PageRank for analyzing protein interaction networks. *Bioinformatics* 27, 405–407 (2011).

[10] Jin, W., Jung, J. & Kang, U. Supervised and extended restart in random walks for ranking and link prediction in networks. *PLOS ONE* 14, e0213857 (2019).

[11] Köhler, S., Bauer, S., Horn, D. & Robinson, P. N. Walking the Interactome for Prioritization of Candidate Disease Genes. *The American Journal of Human Genetics* 82, 949–958 (2008).

[12] Kondor, R.I., & Lafferty, J.D. Diffusion kernels on graphs and other discrete input spaces. *In ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning* 315–322 (ICML, 2002).

[13] Li, J. et al. A computational method using the random walk with restart algorithm for identifying novel epigenetic factors. *Mol Genet Genomics* 293, 293–301 (2018).

[14] Li, L., Wang, Y., An, L., Kong, X. & Huang, T. A network-based method using a random walk with restart algorithm and screening tests to identify novel genes associated with Menière's disease. *PLOS ONE* 12, e0182592 (2017).

[15] Ni, J. et al. Disease gene prioritization by integrating tissue-specific molecular networks using a robust multi-network model. *BMC Bioinformatics*, 17 (2016).

[16] Nitsch, D., Gonçalves, J. P., Ojeda, F., de Moor, B. & Moreau, Y. Candidate gene prioritization by network analysis of differential expression using machine learning approaches. *BMC Bioinformatics* 11, 460 (2010).

[17] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E.A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. (2020) SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, in press.

[18] Rappaport, N., Twik, M., Plaschkes, I., Nudel, R., Iny Stein, T., Levitt, J., Gershoni, M., Morrey, C.P., Safran, M., & Lancet, D. MalaCards: an amalgamated human disease compendium with diverse clinical and genetic annotation and structured search. *Nucleic Acids Research*, 45(D1):D877-D887, (2017).

[19] Stelzer, G., Rosen, R., Plaschkes, I., Zimmerman, S., Twik, M., Fishilevich, S., Iny Stein, T., Nudel, R., Lieder, I, Mazor, Y., Kaplan, S., Dahary, D., Warshawsky, D., Guan - Golan, Y., Kohn, A., Rappaport, N., Safran, M., & Lancet, D. The GeneCards Suite: From Gene Data Mining to Disease Genome Sequence Analysis. *Current Protocols in Bioinformatics*, 54:1.30.1 - 1.30.33, (2016).

[20] Szklarczyk, D. et al. STRING v11: protein-protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic Acids Res.* 47, D607–D613 (2019).

[21] Titeca, K., Lemmens, I., Tavernier, J. & Eyckerman, S. Discovering cellular protein-protein interactions: Technological strategies and opportunities. *Mass Spectrometry Reviews* 38, 79–111 (2019).

[22] Tong, H., Faloutsos, C. & Pan, J. Fast Random Walk with Restart and Its Applications. in *Sixth International Conference on Data Mining* (ICDM'06) 613–622 (IEEE, 2006). doi:10.1109/ICDM.2006.70.

[23] von Mering, C., Jensen, L.J., Kuhn, M., Chaffron, S., Doerks, T., Krger, B., Snel, B., & Bork, P. STRING 7–recent developments in the integration and prediction of protein interactions. *Nucleic Acids Res.* 35 (2008).

[24] White, S. & Smyth, P. Algorithms for Estimating Relative Importance in Networks. *In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 266–275 (ACM, 2003). doi:10.1145/956750.956782.

7 Appendix

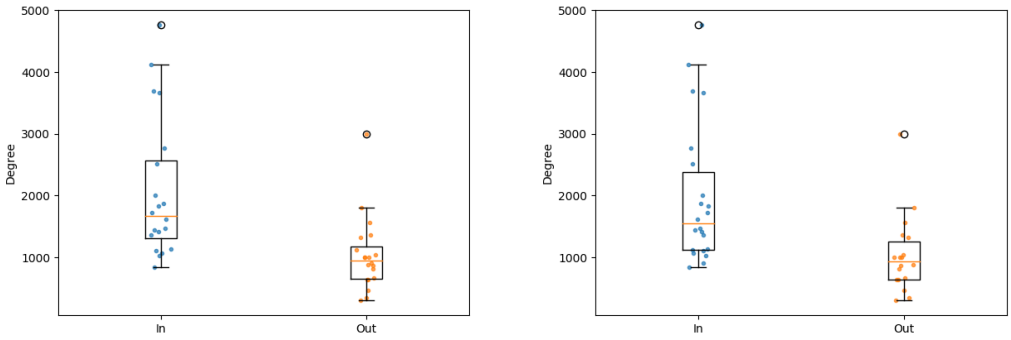


Figure 6: Box plots of input vs. output gene degrees for endometriosis. Left: RWR. Right: PageRank.

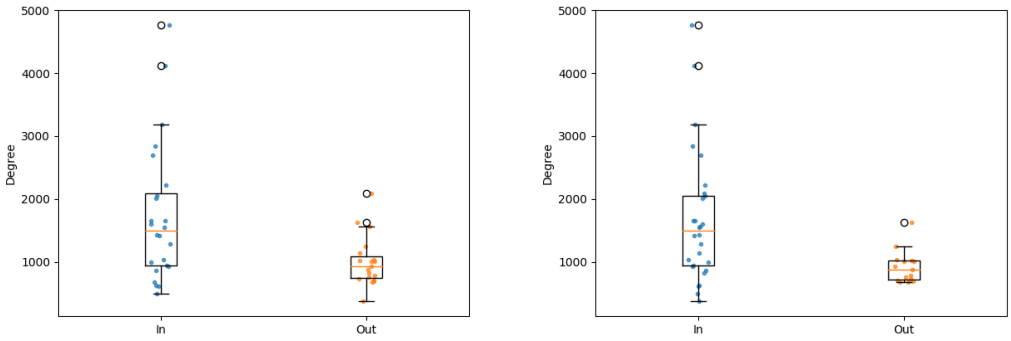


Figure 7: Box plots of input vs. output gene degrees for ischaemic stroke. Left: RWR. Right: PageRank.

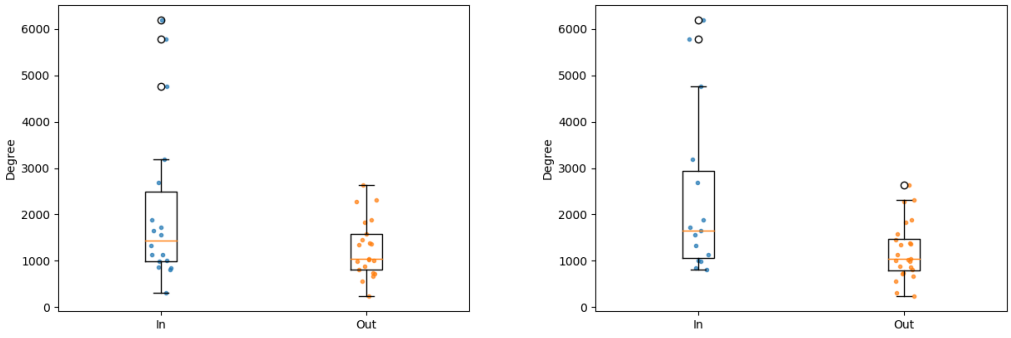


Figure 8: Box plots of input vs. output gene degrees for lymphoma. Left: RWR. Right: PageRank.