DBDM Assignment 4

Topology based Protein-Protein Interaction Network Analysis for Prioritization of Candidate Genes associated with Ménière's Disease

Paul Couzy (s1174347), Irene Haasnoot (s1258745)
Eva van Weenen (s1376969), Alex van Vorstenbosch (s1526146)

July 18, 2019

Abstract

In this paper we study a Protein-Protein Interaction network of 19576 unique proteins to discover which genes are most likely related to Ménière's disease. We use the Random Walk with Restart algorithm and use a permutation-, interaction- and enrichment test to generate a list of the 100 genes associated with these proteins, which are most likely linked to Ménière's disease.

1 Introduction

Ménière's disease (MD) is a disorder of the inner ear. People carrying this disease experience dizziness, ringing in the ear and hearing loss for episodes that last generally a few minutes to a few hours. The cause of MD is not fully understood and is thought to be involved by both genetic as environmental factors. In this assignment the involvement of proteins with Ménière's disease is investigated.

A Protein-Protein Interaction (PPI) Network is constructed to link proteins to each other. A PPI network $\boldsymbol{N}$ (for further details see section 2.2) is modeled as a graph where each node is a protein and each edge is a physical interaction between two proteins. The links are weighted with a score indicating the strength of the interaction. Based on a set of genes known to be associated with MD (corresponding to a set S consisting of 106 proteins), additional candidate genes associated to the disease are identified.

In section 2 we describe the data set used to mine the genes associated with MD and explain our algorithm and the choices we made to find the most interesting genes. In section 3 we present the candidate genes we find with our algorithm. In sections 4 and 5 we present a discussion and conclusion.

2 Methods

In this section the system-parameters, the code, the data-set, and the methods and processes used to analyze the data are discussed.

2.1 System Parameters

The following systems parameters are used to conduct the experiments:

Setup – STRW University

- CPU: 3.20GHz Intel(R) Core(TM) i5-3470, 32K L1d cache, 32K L1i cache, 256K L2 cache, 6144K L3 cache
- Main memory: 8GB
- Disk: 500GB HDD
- OS: Fedora 21.0

The data used in this setup was stored on a network disk on a different STRW server that was mounted on the computer used. This disk has a storage capacity of 1.5 TB.

2.2 Data set

In this assignment the data used to model the PPI Network $\boldsymbol{N}$ for Homo Sapiens is the STRING database version 10.5 (Szklarczyk et al., 2017). In this data set each interaction contains two proteins, represented by their `Ensembl_IDs`, and a score that indicates the strength of the interaction between the two proteins. This version of the database contains 11353056 protein-protein interactions and 19576 unique proteins.

A second set S contains 106 genes which are known to be associated with MD. These two sets are used as input information to infer candidate genes associated with MD.

The set G consists of 43 novel genes that are probably associated with MD and the set V is a subset of G containing the genes that were biologically validated. These genes were identified in the research of Li et al. (2017) by following a similar approach as we will present here. In the approach of Li et al. (2017) another version of the STRING database (containing more PPI's) is used. We expect to find (a part of) this set G in our set C, a set of 100 candidate genes that will be inferred in this assignment.

2.3 Pre-processing

To construct the PPI efficiently, we use a bijection to map the genes from the sets $\boldsymbol{N}$, S, G and V to integers, in such a way that each integer corresponds with only one gene. This mapping can be used at the end of the data-mining to return to the original genes. Furthermore, we remove 32 genes from set S that are not in our PPI network $\boldsymbol{N}$. The set of genes G was created with a version of the STRING database that contains more PPIs than we use in this research. Therefore remove the one gene of G and V that cannot be found in $\boldsymbol{N}$. Our resulting set S now contains 74 genes that are known to be associated with MD, and our sets G and V contain 42 and 14 genes respectively.

2.4 RWR-Algorithm

To retrieve a set C of candidate genes from the PPI network, we use the Random Walk with Restart (RWR) algorithm. It is a type of ranking algorithm, that uses a random walk to rank the importance of nodes in a graph. We start from one of the 74 known genes in set S associated with MD. From this point we take a random walk on the weighted graph of the PPI-network. The probability of each edge is given by the occurrence of the protein-protein interaction between the two nodes.

The probability vector P_i indicates the probability of being associated with MD for all unique genes in the network. The vector is updated after each iteration i of the random walk, with P_0 being initiated as a seed with a probability of 1/74 for each of the known genes and zero otherwise. The restart probability c, is the probability that the random walk restarts to the original genes, c is tested for a range of values between 0.3 and 0.7. This restart probability is used to characterize the importance of the genes in set S $\boldsymbol{A}$ is the transition-matrix of the PPI-network, or the representation of the weighted graph. This gives the following update-rule for P_{i+1}, if we would follow the random walk:

$$P_{i+1} = (1-c)\boldsymbol{A}^\top P_i + cP_0 \tag{1}$$

The random walk is terminated after 100 iterations, or when it converges $||P_{i+1} - P_i|| < 10^{-6}$. When c is chosen to be high, the algorithm will terminate earlier than the maximum number of iterations. When the RWR-algorithm is terminated, the probability vector P_i is the list of probabilities of genes to be linked to MD. Of this list, we only select the genes with probabilities above a threshold of 10^{-5}.

2.5 Selection rules

The RWR-algorithm may return false positives: genes that have a high probability of being linked to MD, but in reality are not. The results of the algorithm depend on the topology of the PPI-network, thus favoring strongly connected genes above others. To reduce the amount of false positives, the following selection rules are used: permutation test, interaction test and enrichment test.

2.5.1 Permutation test

The permutation-test is used to correct for the bias of the RWR-algorithm towards strong connections. We first generate 1000 different sets, where each set is filled with 74 genes, randomly picked from the `Ensembl_IDs` from the network. For each set we run the RWR-algorithm, with these 74 genes as the seed for the algorithm, and check which genes are associated with MD as follows. For each candidate from the RWR-algorithm we calculate its p-value. The p-value is calculated by comparing the probability of the RWR-algorithm of the 74 genes in set S, which are associated with MD with the 1000 probabilities of our random ensemble;

$$p = \frac{\Theta}{1000} \tag{2}$$

where Θ is the count of where the probability of a gene is higher in a random ensemble than from the 74 associated genes.

A cut-off of $p < 0.05$ is used. A high p-value indicates that a gene is likely to be a candidate gene due to random chance, and not actually linked to MD. Therefore, all genes with a p-value equal or greater than 0.05 are eliminated in this process.

2.5.2 Interaction test

To mine the most related candidate genes, two tests, namely, the interaction test and the enrichment test, are implemented to directly or indirectly measure the association between the candidate genes and MD.

The interaction test is applied on the original PPI Network (before the RWR Algorithm was applied). The test is based on the widely accepted idea that two proteins that strongly interact with each other will have similar functions. Therefore, a gene that interacts strongly with a known gene associated to MD will be more likely be a candidate gene.

For this test the maximum interaction score (MIS) is computed by selecting the maximum score between a gene g and the genes from set S. A threshold of 900, the cutoff of highest confidence in the STRING database, is used and only candidate genes with a MIS greater than or equal to this threshold are considered.

2.5.3 Enrichment test

The enrichment test is also based on the relation of the genes to the known genes associated with MD. Instead of looking at the interaction score, this test is built based on the Gene Ontology (GO) terms [Wang et al. (2015)] and Kyoto Encyclopedia of Genes and Genomes (KEGG) pathways [Kanehisa and Goto (2000)]. The idea behind this test is that a candidate gene is likely to be a novel MD-related genes if it has a similar relationship with GO terms and KEGG pathways as genes that are known to be associated with MD. According to the enrichment theory [Yang et al. (2014)] these terms and pathways can be converted to a numeric vector $FV(g)$. This vector is a combination of the GO enrichment and the KEGG enrichment:

$$\begin{align} FV(g) =& (S_{GO}(g,t_1),...,S_{GO}(g,t_n), \tag{3}\\ & S_{KEGG}(g,P_1),...,S_{KEGG}(g,P_k)) \tag{4}\end{align}$$

where the GO enrichment equals

$$S_{GO}(g,t_j) = -log_{10}\left(\sum_{k=m}^{n}\frac{\binom{M}{m}\binom{N-M}{n-m}}{\binom{N}{n}}\right) \tag{5}$$

with N the number of overall proteins of the Homo Sapiens, M the number of proteins that have the term GO_j, n the number of direct neighbours of g, and m the number of direct neighbours of g that have the term GO_j. The KEGG enrichment equals

$$S_{KEGG}(g,P_j) = -log_{10}\left(\sum_{k=m}^{n}\frac{\binom{M}{m}\binom{N-M}{n-m}}{\binom{N}{n}}\right) \tag{6}$$

with N the number of overall proteins of the Homo Sapiens, M the number of proteins in pathway P_j, n the number of direct neighbors of g, and m the number of direct neighbours of g that are in pathway P_j. N in equations 5 and 6 should not be confused with our Protein-Protein Interaction network $\boldsymbol{N}$.

The distance between the two genes g and g',where g' is a gene from set `S`, is then defined by the cosine of the vectors $FV(g)$ and $FV(g')$. Similar to the MIS, the maximum enrichment score (MES) can be computed.

2.6 Code

This project was carried out using `Python 3.7` (Python Core Team, 2018). The implementation of the RWR algorithm is based on the Personalized PageRank (PPR) implementation of Jung (2018). Furthermore python packages `numpy`, `scipy`, `networkx`, `goatools`, `biotools`, `bioservices` and `sharepathway` were used.

The implementation of the enrichment test is not completed. Following the GO enrichment analysis presented by Dessimoz and Skunca, p-values are obtained and a selection is made based on the Bonferroni correction (p-value < 0.01). A start is made to implement the KEGG Enrichment analysis (T. Cokelaer (2012-2017)), but it is not used for our final results.

3 Results

To determine the restart probability c, we vary over c and determine the amount of genes in sets S, G and V that are still candidate genes after the Random Walk with Restart algorithm, the permutation test, the interaction test and the GO enrichment test respectively. These results can be found in Table 1. We find that a lower c results in more candidates generated by the RWR algorithm. However, the subsequent tests remove a lot of candidates and all c's end up with the same number of candidates in S, G and V after the interaction test. Having a large number of candidates does negatively impact the run time of our tests, therefore we find a restart probability of $c = 0.5$ to be the best trade-off.

The Random Walk with Restart algorithm returns all genes of S, G and V as candidate genes. The intersection between our candidate genes and G and V is significantly reduced after the permutation test – note that the performance of the permutation test depends on the PPI network. The interaction test reduces the number of candidate genes intersecting with S the most (54%) and subsequently, this intersection is reduced with 33% by the GO enrichment test.

The final number of candidate genes found with the RWR algorithm after the three statistical tests is then 244. We sort this list of candidate genes by the probability obtained from the Random Walk and select the top 100 genes from the list. The table of candidates genes can be found in the attached file `C.txt`. The genes of set G and V and their positions in `C.txt` can be found in Table 2.

4 Discussion

Our list C of top 100 genes associated MD, contains less genes from the set G than (Li et al., 2017). This is not strange, as the PPI network used by those authors contains more PPI interactions and lists more genes associated with MD. As the RWR algorithm and permuation test depends heavily on the network, the version of the STRING database affects the results in a significant way. Unfortunately the implementation of the KEGG enrichment was more complicated than we anticipated. As the KEGG enrichment was too advanced to implement, we used the GO-enrichment to give back a p-value for the proteins and use this a selection rule for the proteins. If we could combine the GO enrichment with the KEGG enrichment our results should improve. Further research could also be conducted to improve the ranking of the genes by the use of a different restart probability c.

5 Conclusion

A PPI network is generated from the STRING database. The RWR algorithm is applied to this network to find 10594 candidate genes which could be linked to MD. As the RWR algorithm depends on the topology of the network, it favours highly connected nodes. To mitigate this bias the permutation test is used. After the permutation test 1083 candidate genes remain. For this selection the interaction test is used to detect the likeliness of the proteins to interact, this returns a list of 366 genes. Next, the GO enrichment test and the Bonferroni method are used to reduce the candidates to a list of 244 genes. Finally, from these candidates the top 100 candidate genes are presented, sorted by their probability of the RWR algorithm.

	Random Walk with Restart				**Permutation test**				**Interaction test**				**GO Enrichment test**			
c	C_{RWR}	$\cap S$	$\cap G$	$\cap V$	C_{perm}	$\cap S$	$\cap G$	$\cap V$	C_{int}	$\cap S$	$\cap G$	$\cap V$	C_{GO}	$\cap S$	$\cap G$	$\cap V$
0.3	12821	74	42	14	1232	74	37	11	405	33	33	10	252	22	29	9
0.4	11650	74	42	14	1161	74	37	11	388	33	33	10	250	22	29	9
0.5	10594	74	42	14	1083	74	37	11	366	33	33	10	244	22	29	9
0.6	9512	74	42	14	1020	74	37	11	347	33	33	10	238	22	29	9
0.7	8294	74	42	14	975	74	37	11	329	33	33	10	232	22	29	9

Table 1: Number of candidates for different values of the restart probability c. C_X indicates the number of candidates of set C for test/algorithm X where X is the Random Walk with Restart (RWR) algorithm, the permutation test (perm) or the interaction test (int). $\cap Y$ indicates the size of the intersection between set C_X and set Y where Y is the set S, G or V.

Ensembl_ID	**Name**	**Rank in C**
ENSP00000258743	IL6	27
ENSP00000353874	TLR9	-
ENSP00000305651	CXCL10	65
ENSP00000392398	GPX5	-
ENSP00000260010	TLR2	-
ENSP00000346103	GPX4	-
ENSP00000354901	CXCL9	-
ENSP00000011653	CD4	-
ENSP00000256646	NOTCH2	-
ENSP00000233946	IL1R1	-
ENSP00000280357	IL18	75
ENSP00000412237	IL10	-
ENSP00000356438	PTGS2	57
ENSP00000225831	CCL2	45
ENSP00000292303	CCR5	55
ENSP00000361359	CD40	61
ENSP00000363822	AR	-
ENSP00000264832	ICAM1	38
ENSP00000306512	IL8	-
ENSP00000252321	KCNA5	-
ENSP00000296871	CSF2	52
ENSP00000216797	NFKBIA	-
ENSP00000155840	KCNQ1	-
ENSP00000162749	TNFRSF1A	82
ENSP00000320084	CD276	-
ENSP00000364114	HLA-DRB5	-
ENSP00000250151	CCL4	-
ENSP00000294728	VCAM1	48
ENSP00000264246	CD80	79
ENSP00000332049	CD86	64
ENSP00000226730	IL2	-
ENSP00000384273	RELA	-
ENSP00000379110	CXCL1	-
ENSP00000329411	IRF7	90
ENSP00000227507	CCND1	-
ENSP00000306245	FOS	-
ENSP00000328511	KCNA4	-
ENSP00000365380	FOXP3	-
ENSP00000311032	CASP3	40
ENSP00000369293	IL2RA	76
ENSP00000264657	STAT3	47
ENSP00000361405	MMP9	60

Table 2: Table of genes G, their Ensembl_IDs, names, and position in C, the list of 100 candidates for Ménières Disease after the GO enrichment test. Subset V of biologically validated genes is highlighted in this table.

Bibliography

C. Dessimoz and N. Skunca. *The Gene Ontology Handbook.*

J. Jung. Python implementation for random walk with restart (rwr), 2018. URL https://github.com/jinhongjung/pyrwr.

M. Kanehisa and S. Goto. Kegg: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.*, 2000.

Lin Li, YanShu Wang, Lifeng An, XiangYin Kong, and Tao Huang. A network-based method using a random walk with restart algorithm and screening tests to identify novel genes associated with ménière's disease. *PLOS One*, 2017.

Python Core Team. *Python: A dynamic, open source programming language, version 3.6.6.* Python Software Foundation, 2018. URL https://www.python.org/.

D. Szklarczyk, J.H.Morris, H. Cook, M. Kuhn, S. Wyder, M. Simonovic, A. Santos, N.T. Doncheva, A. Roth, P. Bork, L.J. Jensen, and C. von Mering. The string database in 2017: quality-controlled protein–protein association networks, made broadly accessible. *Nucleic Acids Res.*, 2017. URL hstring-db.org.

J. Serra-Musach D. Pultz T. Cokelaer, L.M. Harder, 2012-2017. URL https://bioservices.readthedocs.io/en/master/kegg_tutorial.html.

Baoman Wang, Fei Yuan, Xiangyin Kong, Lan-Dian Hu, , and Yu-Dong Cai. Identifying novel candidate genes related to apoptosis from a protein-protein interaction network. *Computational and Mathematical Methods in Medicine*, 2015.

Jing Yang, Lei Chen, Xiangyin Kong, Tao Huang, and Yu-Dong Cai. Analysis of tumor suppressor genes based on gene ontology and the kegg pathway. 2014.