Computational Health Laboratory project report

Dalla Noce Niko, Ristori Alessandro, Zuppolini Andrea
Master Degree in Computer science.
n.dallanoce@studenti.unipi.it, a.ristori5@studenti.unipi.it, a.zuppolini@studenti.unipi.it.
Computational Health Laboratory, Academic Year: 2021/2022
Date: 16/05/2022
`https://github.com/nikodallanoce/ComputationalHealthLaboratory`

Abstract

Starting from one or more genes, extract from interaction databases the genes they interact with. Using the expanded gene set, perform pathway analysis and obtain all disease pathways in which the genes appear. Merge the pathways to obtain a larger graph. Perform further network analysis to extract central biomarkers and communities beyond pathways. Compute a distance between the initial gene set and the various pathways (diseases).

Contents

List of Figures

1 Introduction

1.1 Setting and case study

The Zhu-Tokita-Takenouchi-Kim (ZTTK) syndrome is a recently discovered disease caused by loss of functions in the gene SON [4]. The heterozygous mutations in this gene is autosomal dominant, with high probability of being inherited.

The most common feature of the ZTTK are intellectual disability, facial dysmorphism, brain malformation and other features linked to brain and the development of the individual but also ocular, facial and physiological features appears to be linked. There are two main ways to detect ZTTK: Brain imaging and WES (whole exome sequencing). The former is a common methodology and the patient showed many abnormalities in form and shape of the brain, but those characteristics are widely shared with many other neurological diseases. The latter is more precise and guarantees good results in identifying mutations in the gene, being more expensive.

The difficulties in diagnosis and the poor quantity of data available make the ZTTK a hard disease both to study and identify. Since the only known responsible, up to now, is the SON gene, it becomes tough to circumscribe the set of causes and effects of the ZTTK. In those cases, techniques like geneset augmentation, pathway finding, pathway analysis, clustering and community detection, can really provide powerful tools to the researchers on the field to simplify their analysis and give them relevant statistics helpful to their experimenting.

1.2 Project roadmap

This project report follows the roadmap we present here:

1. **Geneset-expansion**, starting from the gene SON and its interactions, we have expanded our set using 1^{st} and 2^{nd} order neighbourhood, querying the BioGRID [2] database. The network was now big enough to find other pathways;

2. **Pathway enrichment**, using the GSEApy [1] package we have performed pathway enrichment with the data in the DisGeNET [3] to associate the genes with the disease pathways they contribute to;

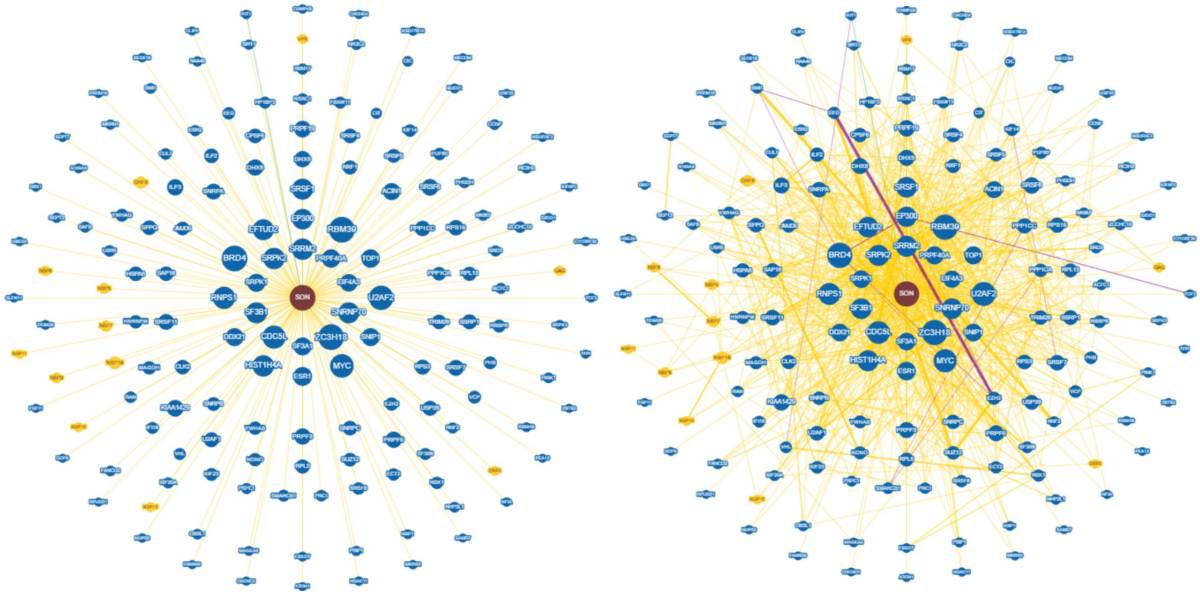
3. **Network analysis**, after building the network, two metrics have been deployed to analyze the diseases on it, such as the distance among the pathway components, the largest pathway component size. Afterwards, we have used the node's degree inside the network to extract the most significant biomarkers;

4. **Community analysis**, as first step we decided on using the Louvain method for detecting the communities, we found between $8 - 11$ communities starting from the graph we have built, weighting the edges using the number of shared diseases between the couples of nodes. Then we compared the pathways in the community we have found for the SON gene with the known relevant effect of the ZTTK syndrom, along with many other metrics.

[1] https://github.com/zqfang/GSEApy

2 Geneset Expansion

As we said in section 1, the only known responsible of the ZTTK disease, up to know, is the SON gene, even though it is a well known gene that contributes to the RNA splicing and the overall cell cycle. In order to study possible concurrent factor to the syndrome we have first enlarged the geneset under our focus (which we retrieved into a csv format from BioGRID [2] and cleaned by hand) by looking at all of the genes that interact with SON and, iteratively, also the second order neighbourhood. The request for the 1^{st} order neighbourhood retrieved 146 proteins and their (eventual) mutual interactions as we can see in Figure 2.1 (b), the second request enlarged our network to 12863 nodes.



(a) The 146 starting interactions

(b) The interactions at the first order

Figure 2.1: Starting interactions from the SON gene and all the first order interactions.

All the information retrieved has been post-processed:

- All the genes have been upper cased, the BioGRID dataset is mantained by its users, so we needed to uniform them;
- All the duplicated interactions were removed, there were 25281 of such interactions and we did not them since our graph is not directed;
- All the self loops were removed, we are not interested in a protein interaction with itself.

As we can see in Figure 2.1, the shape of the network is well-defined, SON is at the center of the interactions and all of the nodes are distributed around it at a distance path of 2, while two generic proteins can have a maximum distance of 4. This graph is the starting point of our analysis, we have now a significant amount of data to perform a satisfying pathway enrichment

3 Pathway Enrichment

In this section, we start from the gene set obtained in section 2 to find all the disease pathways linked to it, to accomplish that we have used GSEApy, a python package that performs pathway enrichment from many sources. At first we tried with the KEGG and Reactome human datasets with no success since we would have to manually remove all those pathways not related to any disease and there was no way to filter them out, then we found the DisGeNET dataset [3], which satisfied our needs.

Passing to the GSEApy *enrichr* method all the nodes in our graph, it performed pathway enrichment on our behalf, then, we filtered the disease pathways by keeping those having a p-value lower than 0.1, totalling 589 pathways which will be used during our future network and community analysis.

An example of disease pathway is in Figure 3.1, a big number of proteins are do not interact between each other, this is totally normal given the size of our graph and the fact that we centered it around a specific gene and limited it at the second order.

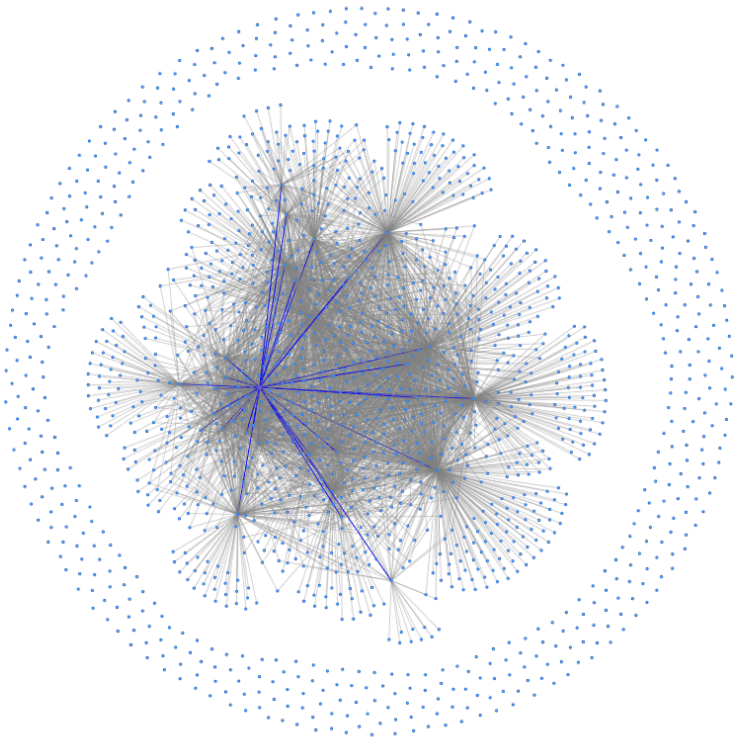


Figure 3.1: A disease pathway with the SON gene and its interactions highlighted in blue.

The GSEApy package, no matter the dataset, always returns a dataframe with following columns (Figure 3.2 shows such dataframe):

- **Term**, the disease pathway name.
- **Overlap**, the ratio of the disease's genes that are in our graph over the entire number of its gene, the former is more useful than the latter for our analysis.
- **P-value**, how much the result is trustable.
- **Adjusted P-value**, the P-value obtained over all the significant tests.
- **Genes**, a string formed by all the genes of our graph that belong to the pathway, they are comma separated.

Term	Overlap	P-value	Adjusted P-value	Genes
Chronic otitis media	55/69	0.005896	0.098950	IGHM;CD81;WIPF1;FMR1;DOCK8;CHD7;JMJD1C;COMT;GT...
Inadequate arch length for tooth size	47/58	0.005953	0.099228	AMER1;SETD5;NOTCH3;TRIO;RPL10;SATB2;GNAI3;PLOD...
Tooth Crowding	47/58	0.005953	0.099228	AMER1;SETD5;NOTCH3;TRIO;RPL10;SATB2;GNAI3;PLOD...
Tooth mass arch size discrepancy	47/58	0.005953	0.099228	AMER1;SETD5;NOTCH3;TRIO;RPL10;SATB2;GNAI3;PLOD...
Tooth size discrepancy	47/58	0.005953	0.099228	AMER1;SETD5;NOTCH3;TRIO;RPL10;SATB2;GNAI3;PLOD...

Figure 3.2: An example of disease pathways retrieved by using the GSEApy package.

Since working with the dataframe genes attribute was not so convenient due to it being a string which we would have to split each time we would have to work with the pathways, we decided to build a dict indexed by the disease's indexes (the first column of the dataframe as seen in Figure 3.2) and using the term as the "name" and the splitted genes as a list attribute.

4 Network Analysis

4.1 Building the protein-to-protein graph

Starting from the interaction dataset built after the geneset expansion (as we have seen in section 2) we developed our protein-to-protein network thanks to the networkx[2] python package. First we inserted the nodes into the graph and named them with the protein they represent, and, at the same time we "colored" them with the disease pathways they belong to. Then, we created the edges from the interaction dataset and we weighted them based on the number of shared diseases (the ones retrieved during the pathway enrichment explained in section 3) between couple of nodes.

We noticed that a relevant number of nodes/proteins does not belong to any disease, more precisely 5101 of the 13010 proteins. Since we did not want to end with possibly a non-connected graph, we decided to keep them. The same can also be seen for the edges, since we have observed that 45000 of them had no shared diseases between the couple of nodes they connect, but the same reasoning we did for the nodes stands for them, so we kept them too. Figure 4.1 shows the how we computed such nodes and edges after building our protein-to-protein graph.

```
nodes_no_disease = list()
for node in protein_graph.nodes:
    if len(protein_graph.nodes[node]["diseases"])==0:
        nodes_no_disease.append(str(node))
```

```
print("Nodes without diseases: {0}".format(len(nodes_no_disease)))
```

```
Nodes without diseases: 5101
```

(a) Nodes with no color/disease

```
edges_no_disease = list()
for edge in protein_graph.edges:
    if protein_graph.edges[edge]["weight"]==0:
        edges_no_disease.append(str(edge))
```

```
print("Edges without diseases: {0}".format(len(edges_no_disease)))
```

```
Edges without diseases: 45404
```

(b) Edges with no weight/disease

Figure 4.1: Jupyter cells that compute the number of nodes and edges that do not belong to any disease pathway.

4.2 Disease network metrics

Once we built the protein-to-protein graph (we talked about it in subsection 4.1) we computed some of the metrics proposed by Agrawal et al. [1] to have a better overview on the effect of each disease pathway on our newly built graph.

Size of largest pathway component *Fraction of disease proteins that lie in the disease's largest pathway component (i.e., the relative size of the largest connected component (LCC) of the disease)* [1]. It indicates how prevalent is the biggest component of genes inside a disease pathway and, therefore, how its genes are connected between them.

Distance of pathway components *For each pair of pathway components, we calculate the average shortest path length between each set of proteins, and then, the average of this is taken*

[2]https://github.com/networkx/networkx

over all pairs of the components [1]. Even though this metric tells us how distant are a disease's genes and, therefore, how long is the path between them, it is not valuable in our case since, as we explained at the end of section 2, our graph is centered around the SON gene and, for that reason, the maximum distance of each node from one another is 4.

4.3 Biomarkers identification

A main issue, not the hardest but for sure the most annoying and tedious one, was related to plotting the graph, which had too many nodes and interactions to be properly drawn. Moreover, we were interested in knowing which were the biomarkers, the "central" nodes of our graph, the ones that have more "impact" on its connectivity. Solving the latter would have also made the plotting far easier, so we it was a logical step at the end of our network analysis.

By using the centrality algorithm based on the nodes' degree from the networkx package, we made a method that retrieves a user-specified number of biomarkers plus one protein of their choice (in our case the SON gene). Figure 4.2 shows the 30 biomarkers identified whose centrality is the ratio between the node degree over the entire number of nodes that compose the protein-to-protein graph.

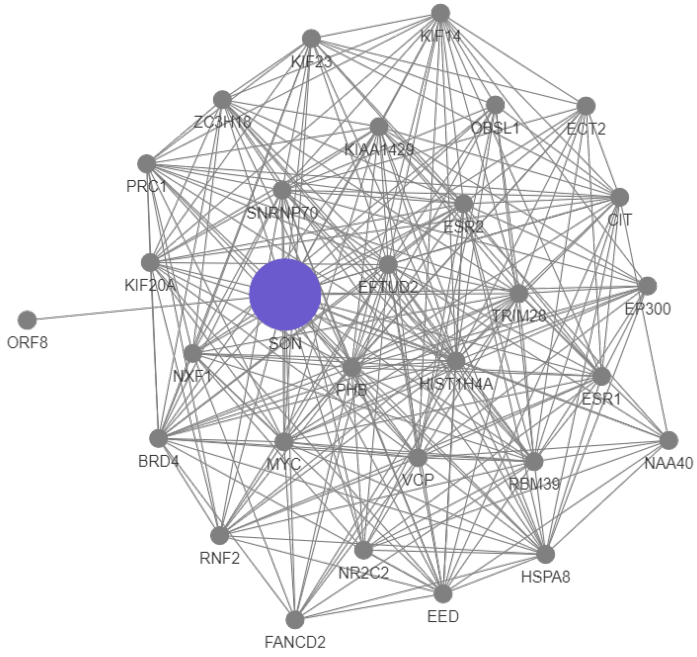


Figure 4.2: Biomarkers with the SON gene highlighted.

5 Community Analysis

5.1 Community detection

At this point, a protein graph is available, weighted using the shared disease (subsection 4.1), that could be helpful to reveal potential clusters of nodes with similar features. However, since our goal is to find non trivial relationships of genes and pathways with the SON gene, the most interesting features in our study case are the hidden features. Namely, the ones which are not directly available but have to be extracted working with the network, using methods like community detection.

Since our network is quite large in terms of both nodes and especially edges, we have decided to use the *Louvain method* for community detection having time complexity $\mathcal{O}(n \cdot \log n)$, with n being the number of nodes. The Louvain method is a greedy algorithm that optimizes the network modularity (5.1):

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j) \tag{5.1}$$

This approach distinguishes group of nodes with high density of connections and delimits the community whenever there is a drop of density in the number of edges. The algorithm stops whenever a local maximum of network modularity is reached, therefore different execution can lead to different detection of the communities.

The algorithm gives back around 6900 different communities, but most of those are made of singletons of only one gene. Performing a cross-check with the initial protein-graph we have noticed that most of those single-gene communities are involved in a minimal number of pathways, in average 2 as seen in Figure 5.1 (a), or are not involved in any pathways at all. Usually, running the algorithm, only between $[8 - 11]$ communities have a significant number of genes with around 680 genes per community (Figure 5.1 (b)), while the remaining ones, the singletons, are sort of outliers, therefore we have decided to leave them out. Up to now, the remaining communities cover more than 6100 genes and we have implemented many methods to highlight their characteristics.

```
mean_diseases_one_node = mean_diseases_communities_size_n(louvain_communities, protein_graph)
print("Mean diseases for those communities with one node: {0}".format(str(mean_diseases_one_node)))
```

```
Mean diseases for those communities with one node: 2.2324127906976745
```

(a) Average singleton community size

```
print("Mean size of communities: {0}".format(str(mean_size_communities(communities))))
```

```
Mean size of communities: 681.1111111111111
```

(b) Average community size

Figure 5.1: Number of diseases for single-gene communities and mean size of kept communities.

Such communities were then plotted with the pyvis package[3] by assigning different colors to better visualize them as we can see from Figure 5.2, we highlighted the SON gene by giving it a bigger size than all the other nodes in the graph. More graphs will be shown in section 6 to better illustrate the results.

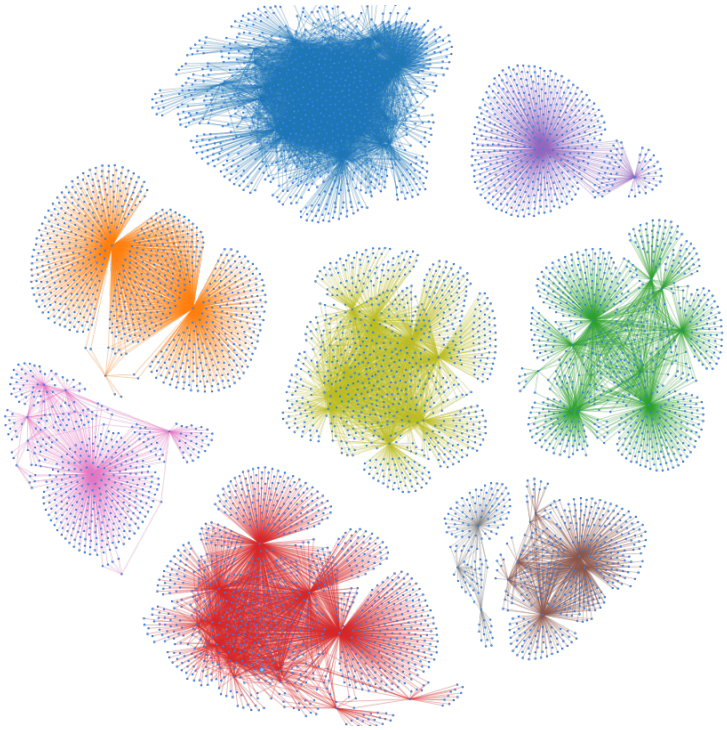


Figure 5.2: Communities after the pruning.

5.2 Community evaluation

In this section there will be the evaluation of the communities, which, in our example, there are 9 of them (Figure 5.2). Every community contains a number of nodes which is involved in many different disease pathways and, to evaluate the goodness of the various clusters, we have implemented several metrics:

- **Ratio disease**, measuring the ratio between the genes of the given community which participate to a disease pathway over the whole size of that pathway. This metric is particularly useful to avoid that the largest pathways are always the major contributors of the community. For example the "Malignant neoplasm of breast", which has around 3400 genes, without this metric would be the most relevant disease in almost every community. Scaling it with its size, gives a better perspective of its actual contribution in the interval $(0, 1]$.

$$R_d = \frac{n_c}{n_d} \tag{5.2}$$

[3] https://github.com/WestHealth/pyvis

- **Ratio community**, again a ratio, this time between the number of genes in a given pathway over the size of the community.This other metric puts in relation the contribution of every pathway in a given community with the community size, rewarding pathways with a lot of contribution in the community and has value in the interval $(0, 1]$, this metrics solves the problem with small pathways entirely contained in the same community.

$$R_c = \frac{n_c}{|V_c|} \tag{5.3}$$

- **Relevance**, represents the best of both worlds. It combines the absolute contribution of the pathway with the community size, and is obtained multiplying the previous 2 metrics. It has, again, values in the interval $(0, 1]$.

$$Relevance = R_d \times R_c \tag{5.4}$$

Community	Disease	Shared genes	Disease genes	Community size	Ratio disease	Ratio community	Relevance
0	Tooth size discrepancy	5	47	1084	0.106383	0.004613	0.000491
1	Tooth size discrepancy	2	47	510	0.042553	0.003922	0.000167
2	Tooth size discrepancy	3	47	894	0.063830	0.003356	0.000214
3	Tooth size discrepancy	5	47	759	0.106383	0.006588	0.000701
4	Tooth size discrepancy	8	47	1085	0.170213	0.007373	0.001255
5	Tooth size discrepancy	4	47	434	0.085106	0.009217	0.000784
7	Tooth size discrepancy	12	47	524	0.255319	0.022901	0.005847

Figure 5.3: The community-disease dataframe with the computed metrics, we choose a disease as an example.

By taking into account the computed metrics we can also determine the distance of every community to all the other ones. Since the distance is based on disease-related metrics, it also represents a similarity value between the communities.For our purpose this metric is not relevant since we needed to focus only on the SON community, but it could be useful for future improvement of an hypothetical pathway analysis package.

```
son_comm = look_for_gene_community("SON", communities)
for i in range(len(communities)):
    print("Distance to community {0}: {1}".format(i, communities_distance(communities_rank, son_comm, i)))
```

```
Distance to community 0: 0.14864256552637062
Distance to community 1: 0.15653731097698675
Distance to community 2: 0.1526746051195029
Distance to community 3: 0.11742705447044309
Distance to community 4: 0.0
Distance to community 5: 0.13705091001054495
Distance to community 6: 0.10726077869655141
Distance to community 7: 0.08689236367490395
Distance to community 8: 0.1406843209622718
```

Figure 5.4: Distance between the SON gene's community towards the other ones, the lower the value the better.

6 Results

In this section we show the results of the community analysis performed in the SON gene's community by applying the metrics presented before, in subsection 5.2.

6.1 Relevant diseases in SON community

In Figure 6.1 we show the top 10 most relevant diseases inside the SON community:

Disease	Ratio disease	Ratio community	Relevance ▲
Intellectual Disability	0.231748	0.514742	0.119290
Mental and motor retardation	0.296758	0.292383	0.086767
Mental Retardation	0.283688	0.294840	0.083643
Poor school performance	0.302294	0.275184	0.083187
Cognitive delay	0.298153	0.277641	0.082780
Mental deficiency	0.294194	0.280098	0.082403
Global developmental delay	0.279859	0.293612	0.082170
Dull intelligence	0.304775	0.266585	0.081248
Low intelligence	0.304775	0.266585	0.081248
Short stature	0.279310	0.199017	0.055588
Generalized hypotonia	0.261053	0.152334	0.039767
Strabismus	0.312693	0.124079	0.038799
Failure to gain weight	0.269136	0.133907	0.036039
Pediatric failure to thrive	0.268473	0.133907	0.035950
Undergrowth	0.270202	0.131450	0.035518
Genetic Diseases, Inborn	0.256684	0.117936	0.030272
Acquired scoliosis	0.275000	0.108108	0.029730
Curvature of spine	0.271565	0.104423	0.028358
Low set ears	0.274262	0.079853	0.021900
Dilated ventricles (finding)	0.329268	0.066339	0.021843
Cerebellar Hypoplasia	0.361345	0.052826	0.019088
Feeding difficulties	0.266667	0.063882	0.017035

Figure 6.1: Diseases found in the SON community ordered by Relevance

Those results seems to be coherent with the state-of-the-art knowledge of the ZTTK syndrome, since most of the diseases presented in Figure 6.1 are related to neurological pathologies or cognitive difficulties, coherently with the ZTTK collateral aspects. Especially for *Intellectual Disability*, encountered in 100% of the ZTTK diagnoses, our results have evidenced the strong link between the two pathologies. Also, as another example, for the case of *global development delay* most of the patients studied in the ZTTK paper have presented such disease.

With respect to all the other diseases, their relation with SON should be experimentally verified even though, apparently, they seem to align, more or less, with all the others.

Those results in Figure 6.1 are ordered in terms of Relevance, because of the unbalancing that could be caused by the other two metrics as discussed in subsection 5.2:

- if we sort the rows using the "Ratio disease" we are rewarding bigger disease pathways such as *Malignant breast carcinoma*, and penalizing smaller pathologies even if their overall contribution is bigger.
- On the other hand, sorting by "Ratio community" would only emphasize how much the disease contributes to a community without considering its pathway size, thus favoring those small disease contained entirely in a community.

6.2 Diseases with SON in their pathways

As seen in subsection 6.1, most of the relevant pathologies inside SON community are well-known effects of the syndrome under inspection, now we need to focus on the disease pathways in which SON belongs to and plot them in order to confirm what we have seen until now.

Let's visualize directly on Figure 6.2, Figure 6.3 and Figure 6.4 the results of our analysis, evidencing the three disease pathways inside SON gene's community. Just to point out a needed observation, the diamond nodes represents genes that belong to such pathway and we colored in orange the edges that connect them, meanwhile the SON gene was highlighted in blue to better distinguish it from the other proteins.

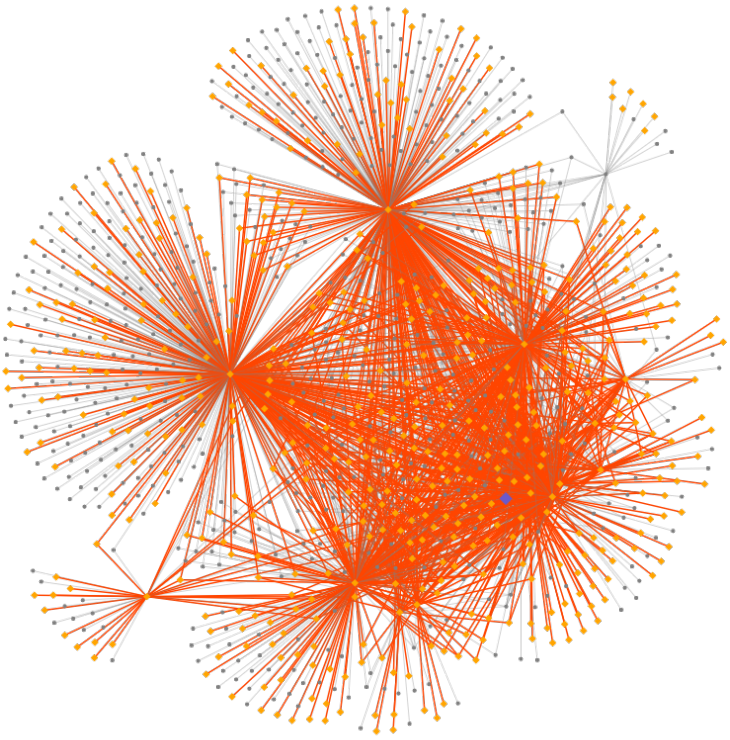


Figure 6.2: Intellectual disability disease pathway within the SON community.

Inside the community there are, more or less, 700 nodes and it can also be noticed that the pathways are well integrated inside the community, as seen for *Intellectual disability* in Figure 6.2, since its genes are connected to many hub nodes, but this was expected as that disease has the highest "Relevance" value.

Let's now plot in Figure 6.3 and Figure 6.4 two more disease pathways that have SON as their gene but do not have a big "Relevance" value as the previous one.

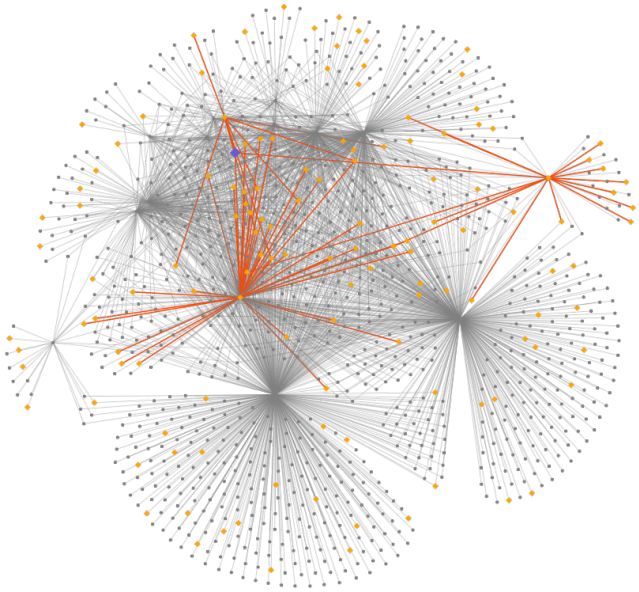


Figure 6.3: Undergrowth disease pathway within the SON community.

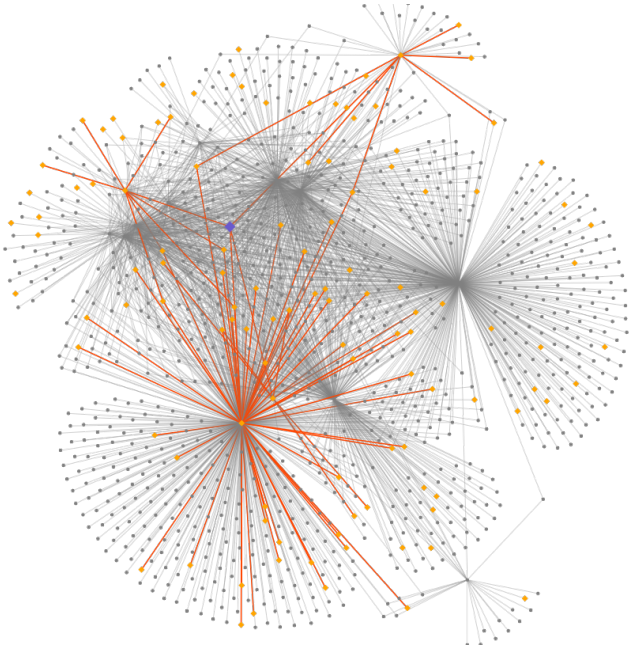


Figure 6.4: Strabismus disease pathway within the SON community.

6.3 Diseases without SON in their pathways

To investigate potential undiscovered correlations between genes in the community and the SON gene, we will now try to explore the diseases which do not contain the latter but manifest a strong interaction in its community. In particular, it is very useful plotting these disease pathways in order to see which of them interact with SON. In Figure 6.5 we show the collected top 20 diseases that don't have the SON gene in their pathway, ordered by the "Relevance" metric.

Disease	Ratio disease	Ratio community	Relevance
Small head	0.342553	0.197789	0.067753
Seizures	0.215686	0.216216	0.046635
Epilepsy	0.203390	0.206388	0.041977
Hypoplastic mandible condyle	0.297125	0.114251	0.033947
Retrusion of lower jaw	0.297125	0.114251	0.033947
Decreased projection of lower jaw	0.297125	0.114251	0.033947
Decreased projection of mandible	0.297125	0.114251	0.033947
Aplasia/Hypoplasia of the mandible	0.295238	0.114251	0.033731
Micrognathism	0.292453	0.114251	0.033413
Malignant neoplasm of breast	0.086157	0.362408	0.031224
Hyperreflexia	0.326180	0.093366	0.030454
Epileptic encephalopathy	0.263305	0.115479	0.030406
Primary microcephaly	0.471154	0.060197	0.028362
Muscle Spasticity	0.286232	0.097052	0.027779
Breast Carcinoma	0.081211	0.332924	0.027037
Mitochondrial Diseases	0.275618	0.095823	0.026411
Cryptorchidism	0.253165	0.098280	0.024881
Muscle hypotonia	0.244776	0.100737	0.024658
Microcephaly	0.301508	0.073710	0.022224
Fetal Growth Retardation	0.266129	0.081081	0.021578

Figure 6.5: The top 20 diseases disconnected from SON, but in its same community.

Beware, many of the disease we have found, do not have a known relation with ZTTK, hence their link should be experimentally verified. For the *Epilepsy* disease this could be a significant result, since it has been encountered in some of the patients with ZTTK, and those results are a good signal indicating a potential relation. Even if the SON gene is not present in that pathway, at this point, it is reasonable to say that there could be a correlation and hence ZTTK patients could benefit of some of the knowledge the medical community has on epilepsy.

Some of the rows in Figure 6.5 present widely studied pathologies such as various types of cancers, but their presence, as already mentioned many times (subsection 5.2), can also derive from the influence that such big pathways have inside strict communities, even though the "Relevance" metric has a mitigation effect on it. Moreover, from the literature, we recall that there is not any proof of correlation between the ZTTK and cancer.

Furthermore we present some plots (Figure 6.6, Figure 6.7 and Figure 6.8) to evidence our results with diseases that do not have SON in their pathways but that could correlated.

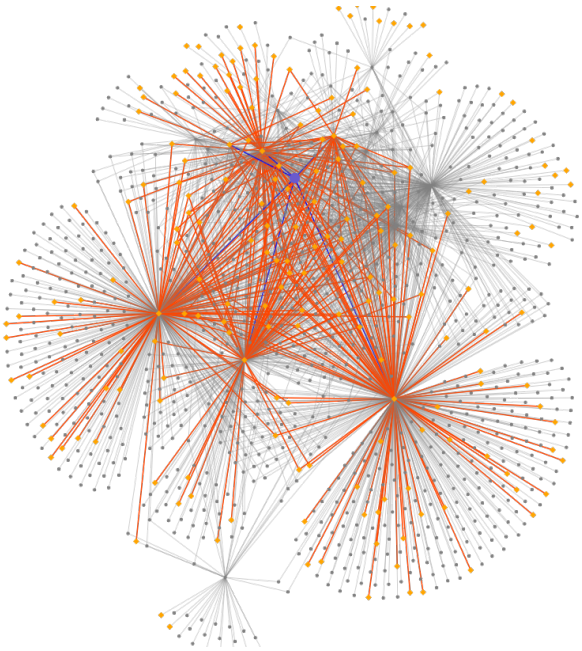


Figure 6.6: Small head disease pathway within the SON community. Blue edges highlights the interaction between the SON gene and the genes inside the pathway.

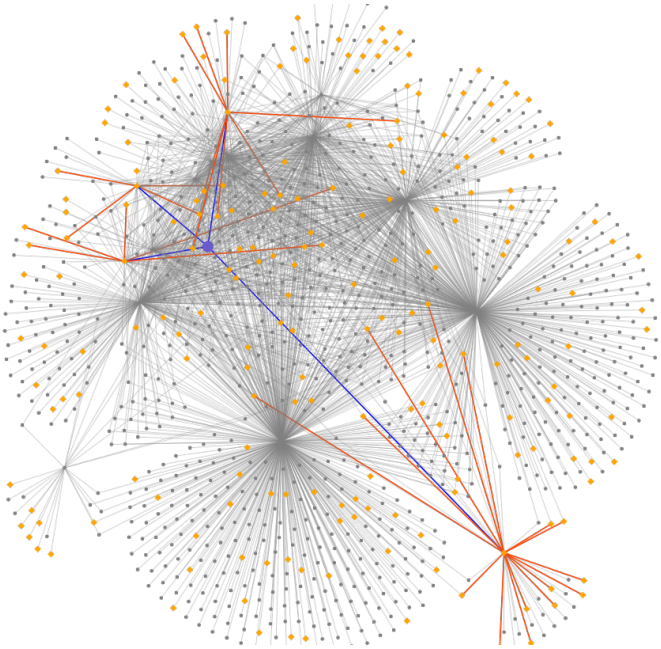


Figure 6.7: Epilepsy disease pathway within the SON community. Blue edges highlights the interaction between the SON gene and the genes inside the pathway.

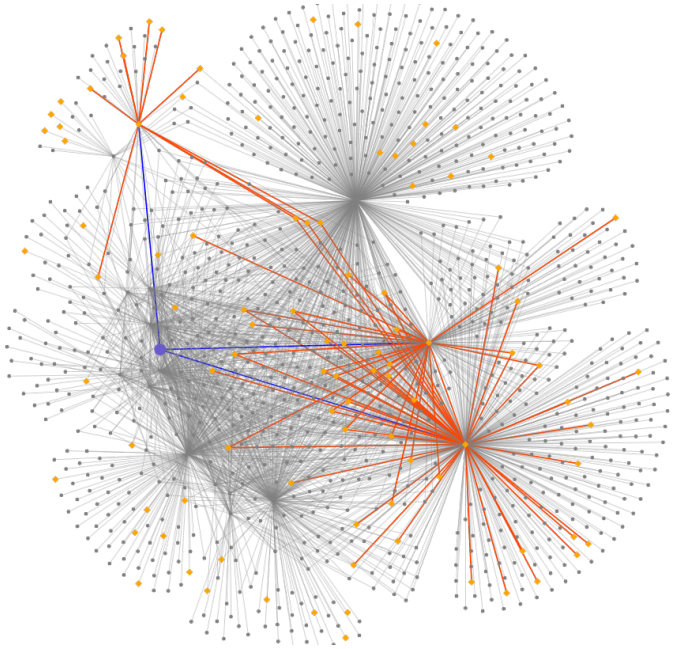


Figure 6.8: Hyperreflexia disease pathway within the SON community. Blue edges highlights the interaction between the SON gene and the genes inside the pathway.

Figure 6.6, Figure 6.7 and Figure 6.8 could be particularly useful for experts because they point out the candidates genes to investigate on (the ones that interact with SON), we present them in Table 1 for two chosen disease pathways.

SON interactions with genes in disease pathways			
Small head disease		**Epilepsy**	
Identifier	**Name**	**Identifier**	**Name**
CIT	Citron Rho-Interacting Serine/Threonine Kinas	PRDM16	Histone-Lysine N-Methyltransferase
PRDM16	Histone-Lysine N-Methyltransferase	UBE2A	Ubiquitin-Conjugating Enzyme E2A
EFTUD2	Elongation Factor Tu GTP Binding Domain Containing 2	NDUFAF2	Myc-Induced Mitochondrial Protein
DCPS	Histidine Triad Nucleotide-Binding Protein 5		
FANCD2	Fanconi Anemia Complementation Group D2		

Table 1: Genes of two diseases, *Small head* and *Epilepsy*, that interact with the SON gene

6.4 Final considerations

For most of us this has been the first approach to bioinformatics, as such we faced many difficulties, especially in the absence of a uniform notation for the data, in the different ways each database uses to represent the knowledge.

Difficulties can also be challenging, after the first, quite annoying, phase of data preparation, the following steps towards the pathway analysis were way more interesting, particularly getting in touch with real world problems and their incredible complexity. Working at something that could (in potential) be lifechanging for someone has a total different feedback and has given us much more motivation than many other works.

Putting aside all the difficulties, we were able to achieve what was requested and possibly discovering other diseases that might be correlated to the SON gene as shown in subsection 6.3, so we can say that our month-long work was useful.

A How to run the project

The project is hosted on a github repository, so to work with there is just need to clone it.

```
git clone https://github.com/nikodallanoce/ComputationalHealthLaboratory
```

Then, create a new python or conda environment (higly recommended) or install the required packages under an already existing environment.

```
pip install -r requirements.txt
```

After that you can run the the jupyter notebooks, which follow the roadmap explained in subsection 1.2:

- **0_Pathway_Enrichment.ipynb**, deals with section 2 and section 3.
- **1_Network_Analysis.ipynb**, deals with section 4.
- **2_Community_Analysis.ipynb**, deals with section 5.
- **3_Plots.ipynb**, methods to plot the protein, disease and community graphs.
- **4_Project_CHL.ipynb**, all the previous notebooks above combined.

It is also extremely important to modify the *config.yml* with a valid token to access the BioGRID [2] datasets. You can also check the source methods inside the *src* directory to have a better look on our work.

References

[1] Monica Agrawal, Marinka Zitnik, and Jure Leskovec. Large-scale analysis of disease pathways in the human interactome. In *Pacific Symposium on Biocomputing*, volume 23, pages 111–122, 2018.

[2] BioGRID. The biological general repository for interaction datasets (biogrid) is a public database that archives and disseminates genetic and protein interaction data from model organisms and humans. `https://thebiogrid.org/`, 2022.

[3] DisGeNET. Disgenet - a database of gene-disease associations. `https://www.disgenet.org/`, 2022.

[4] Sulagna Tina Kushary, Anya Revah-Politi, Subit Barua, Mythily Ganapathi, Andrea Accogli, Vimla Aggarwal, Nicola Brunetti-Pierri, Gerarda Cappuccio, Valeria Capra, Christina R Fagerberg, et al. Zttk syndrome: Clinical and molecular findings of 15 cases and a review of the literature. *American Journal of Medical Genetics Part A*, 185(12):3740–3753, 2021.

[5] Thanh-Phuong Nguyen, Corrado Priami, and Laura Caberlotto. Novel drug target identification for the treatment of dementia using multi-relational association mining. *Scientific reports*, 5(1):1–13, 2015.