Human Disease Network

Omar Ghetti 793181

Contents

1 Introduction

This project aims to make a complete rundown over the Human Disease Network, exploring the data and evaluating clustering algorithms to find correlations between different nodes, that in this precise case are representing the different diseases. Two diseases are connected to each other only if they share at least one gene in which mutations are associated with both diseases.Different State-of-the-art analysis over the network proved that some of those genes tend to group in well distinct clusters, while a big part of them tend to isolate themself at the borders of the network, or to avoid to group in general. The aim of the project is to validate those results and to test some community-detection algorithms over the network, to see if the results are comparable to the state-of-the-art analisys. The network presented by the researchers presents 22 defined diseases classes, defining an initial groundtruth useful for the final comparison with the algorithms results.

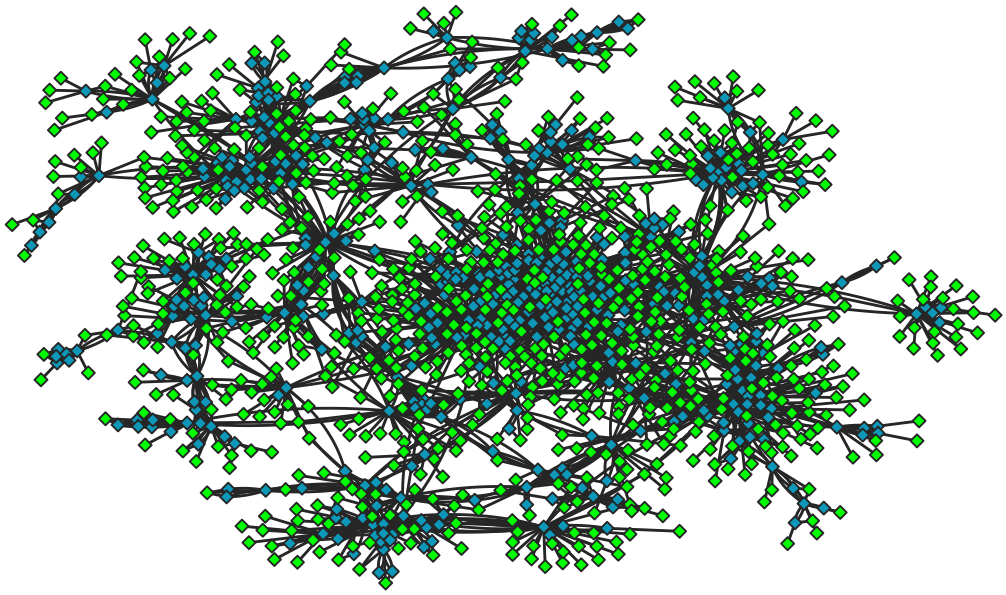
2 Data Exploration

2.1 Network Exploration

First of all, a panoramic view on the data presented in the dataset is useful to start looking at the network, so a console printing of the data is presented, followed by a first network representation made with ggraph library

```
## IGRAPH d5f45a8 DNW- 1419 3926 --
## + attr: name (v/c), label (v/c), X0 (v/c), X1 (v/c), Type (e/c), id
## | (e/n), weight (e/n)
## + edges from d5f45a8 (vertex names):
##  [1] 1285->3211 468 ->2914 416 ->3825 126 ->2566 126 ->1329 288 ->3831
##  [7] 407 ->2203 473 ->3720 282 ->1338 282 ->1339 282 ->2851 686 ->2814
## [13] 575 ->3609 575 ->1348 199 ->3160 199 ->3161 199 ->3248 199 ->3292
## [19] 112 ->3780 113 ->3780 113 ->2309 113 ->1361 62  ->3571 923 ->3931
## [25] 542 ->1631 466 ->2105 466 ->3730 77  ->3712 856 ->1394 856 ->3830
## [31] 856 ->1396 643 ->3554 545 ->1402 545 ->2974 664 ->1515 157 ->2966
## [37] 30  ->1420 30  ->1421 30  ->3553 30  ->3539 30  ->1424 30  ->3985
## + ... omitted several edges
```

Human Disease Network



In this graph, genes are highlighted in Green, while diseases are highlited in Blue. It's possible to see that the network is quite wide, consisting of more than 3000 nodes and more than 1000 vertices. the gene class is also more represented than the disease one, presenting two times the number of gene. It's also important to note that we are working with a directed graph.

```
## [1] "Vertices Number: 1419"
```

```
## [1] "Genes Number: 903"
```

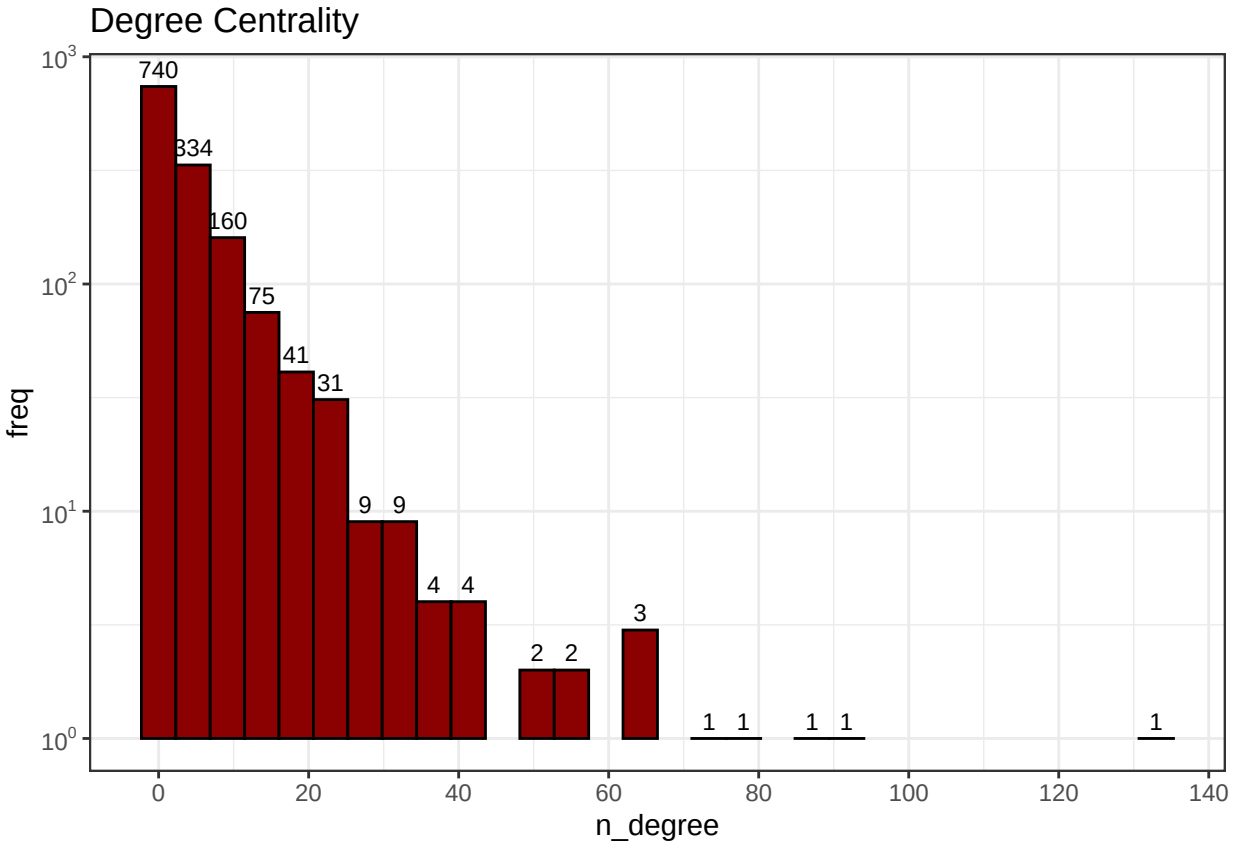
```
## [1] "Diseases Number: 516"
```

```
## [1] "Edges Number: 3926"
```

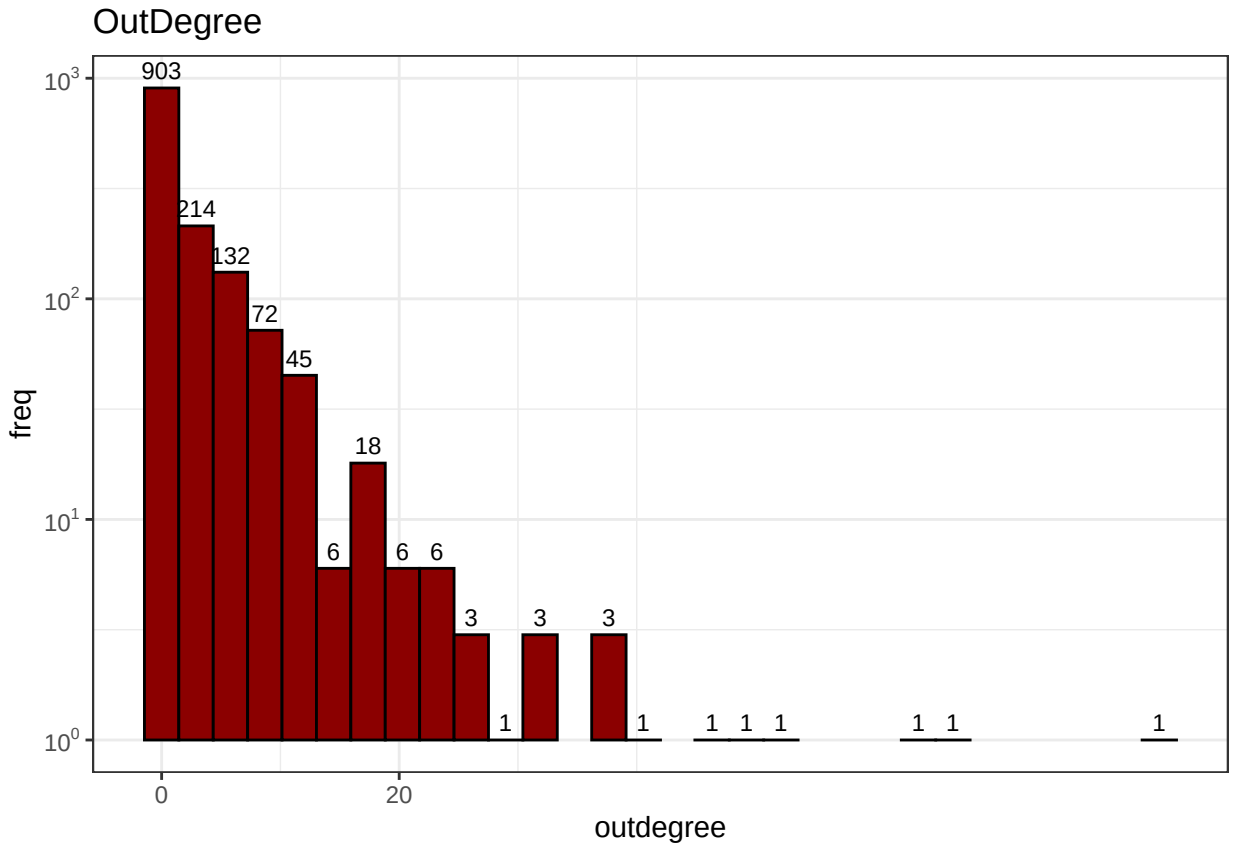
3 Centrality Analisys

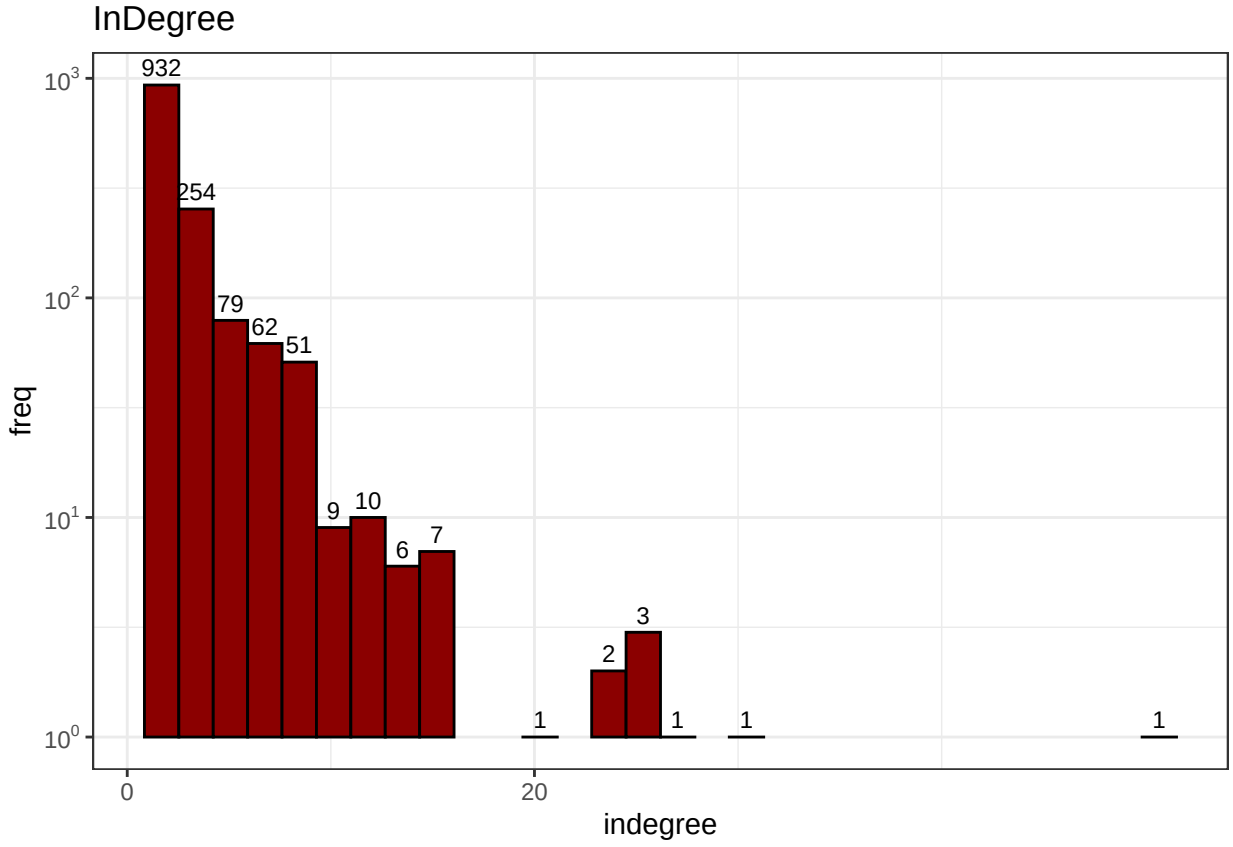
3.1 Degree Centrality

giving the fact that the graph is a directed graph, analisys over degree will take count of Outdegree and Indegree for a certain node, cause in the case of a directed graph both are present. Nodes with high degree will be considered as Hubs for the network, and the idea is that they will be central in clusters. values for degree centrality are normalized.



analyzing the results, it's clear that a lot of nodes have a centrality value close to 1, and, given the number, it's fair to say that genes are the main cause of this behaviour cause the majority of them have 1 as degree.





looking specifically at indegree numbers and outdegree numbers, it's possible to see that no one of the genes

present an outgoing edge, and they just have one entering edge. it's now presented a rank of the top and bottom 10 nodes for degree.

```
## [1] "TOP 10 NODES FOR DEGREE"
```

```
##  [1] "Colon cancer"         "Deafness"             "Leukemia"
##  [4] "Breast cancer"        "Diabetes mellitus"    "Gastric cancer"
##  [7] "Thyroid carcinoma"    "Retinitis pigmentosa" "Cardiomyopathy"
## [10] "Pancreatic cancer"
```

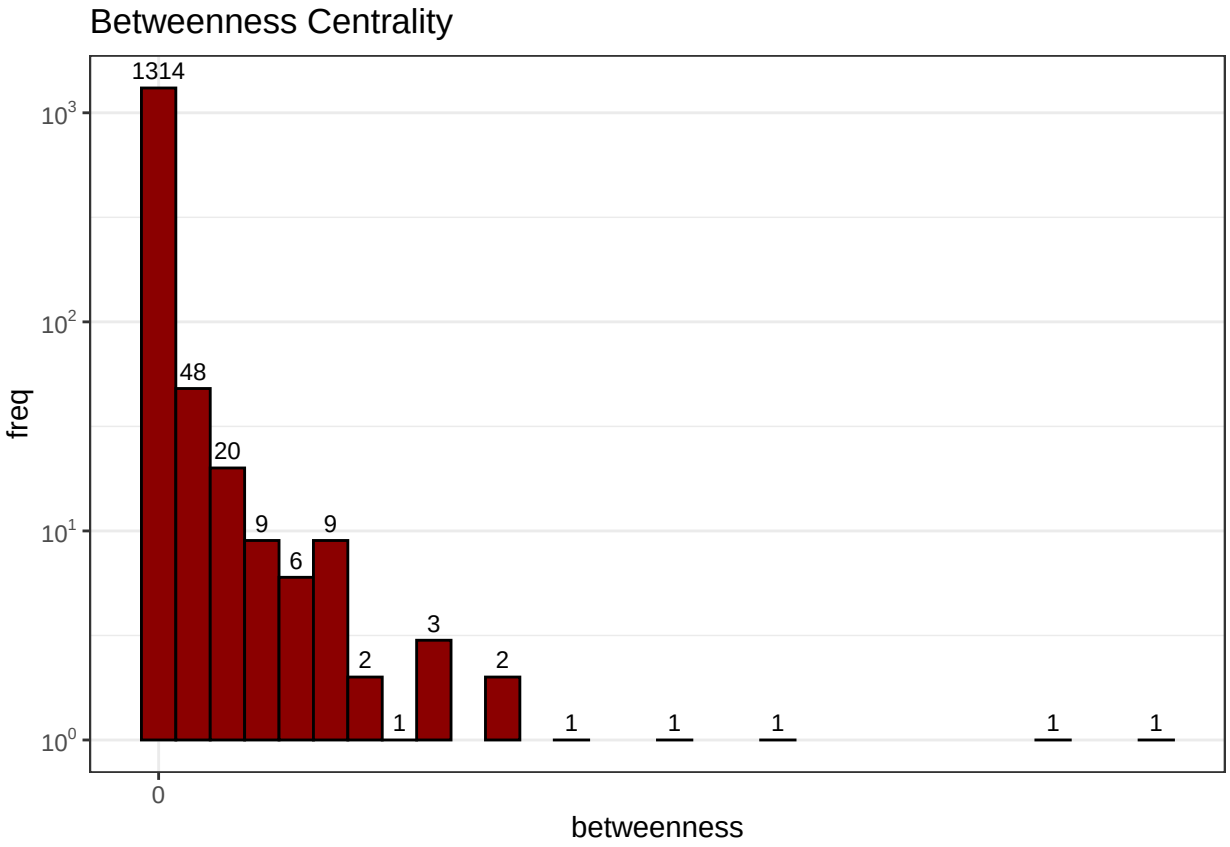
```
## [1] "BOTTOM 10 NODES FOR DEGREE"
```

```
##  [1] "MTP"   "CNGA3" "GNAT2" "SSTR5" "PLAG1" "OCA2"  "TYRP1" "GFAP"  "A2M"
## [10] "APBB2"
```

as expected from the histograms, all the top 10 nodes are diseases, while all the bottom 10 are genes.

3.2 Betweenness

Betweenness is a measure that represents how central is a node based on how many shortest paths on the totality of them present the selected node. this is a more robust metric to evaluate centrality of a node because is not only based on the node itsefl. The measure is calculated in his normalized variant.

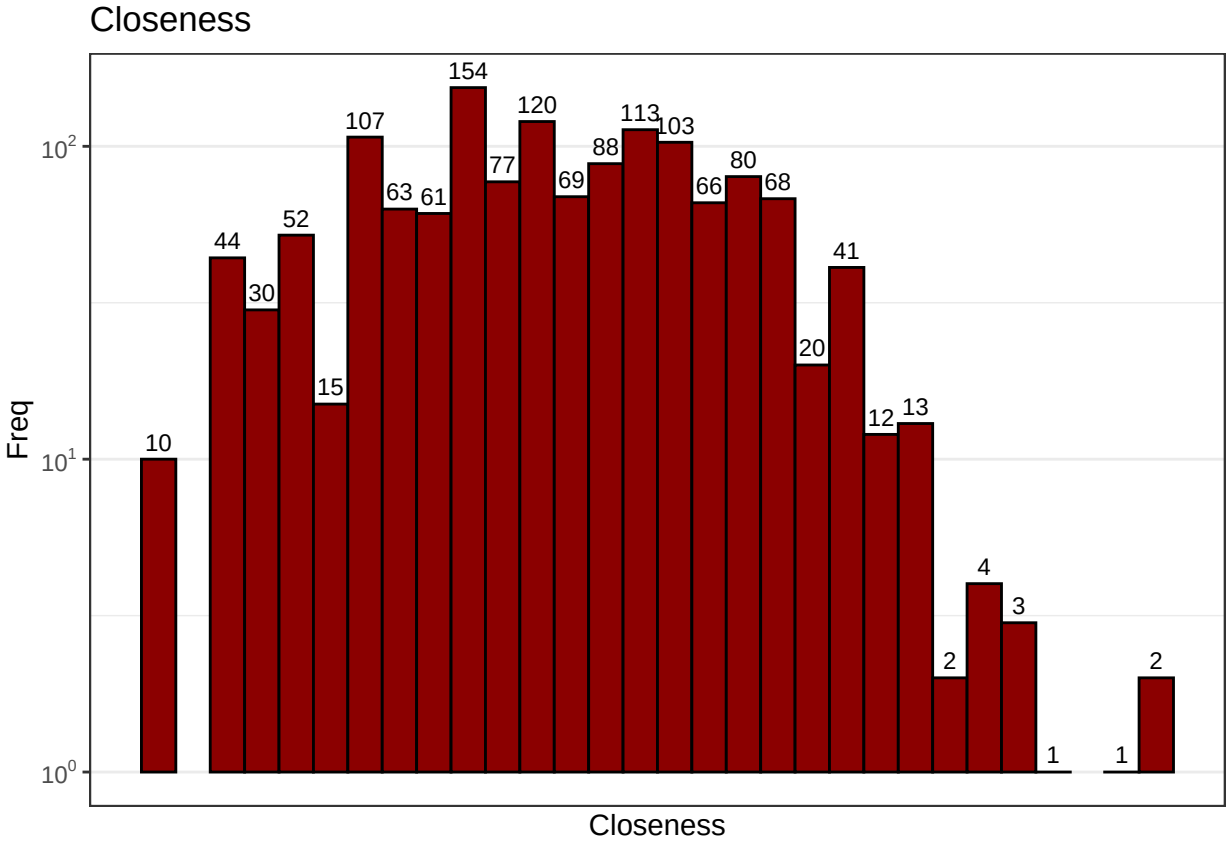


as for the degree measure, a lot of nodes sits on a low value of betweenness. Just 5 of the nodes have a quite higher value of Betweenness, and it's expected that they will be hubs. Those are the top nodes for Betweenness:

```
##  [1] "Cardiomyopathy"    "Lipodystrophy"     "Diabetes mellitus"
##  [4] "Glioblastoma"      "Deafness"          "Myopathy"
##  [7] "Cataract"          "Leukemia"          "Colon cancer"
## [10] "Alzheimer disease"
```

3.3 Closeness

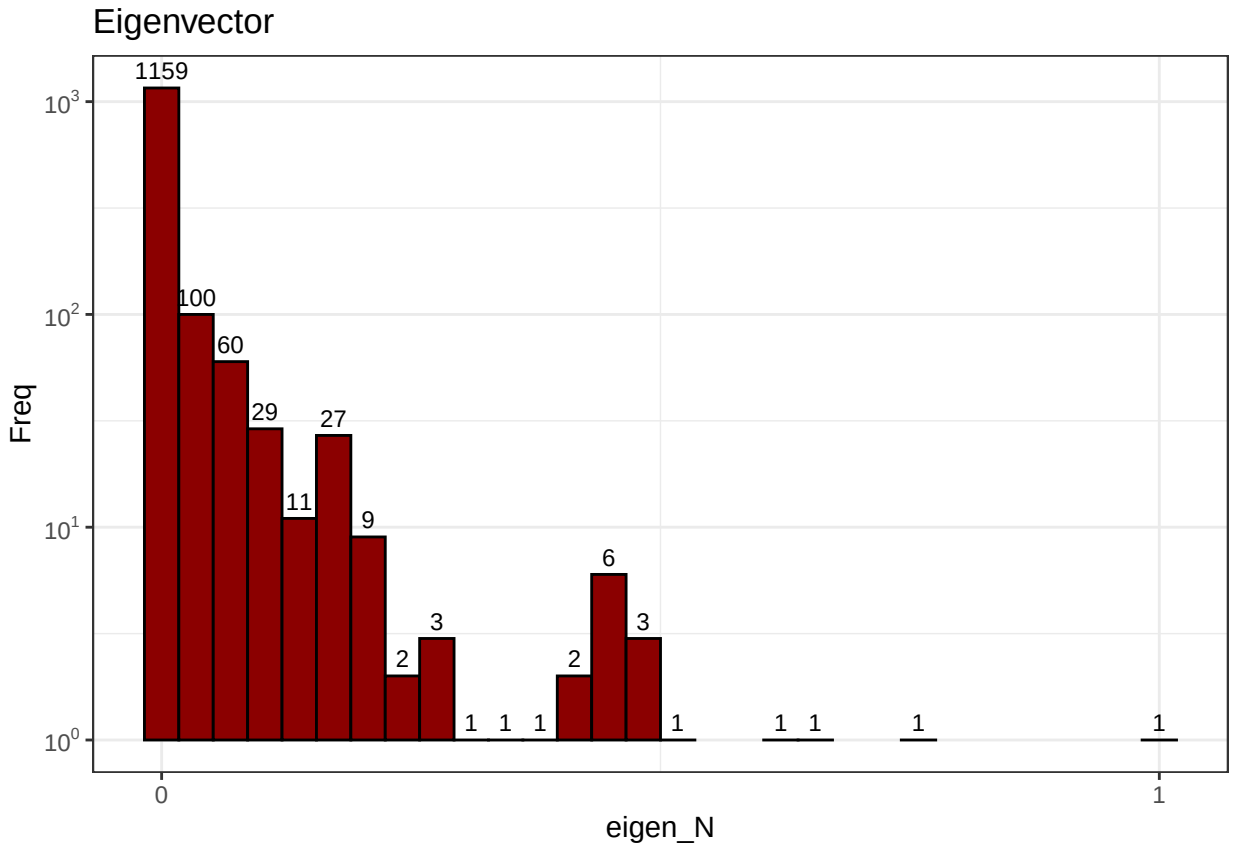
Closeness is a measure that highlights the ability of a node to spread informations effectively and quickly. Is based on the average farness of a node from the other ones. It's expected that nodes with high closeness have short distances from other nodes. This measure is pretty unrelevant when the network is undirected, but we are working over a dircted network so it's worth having this measure.

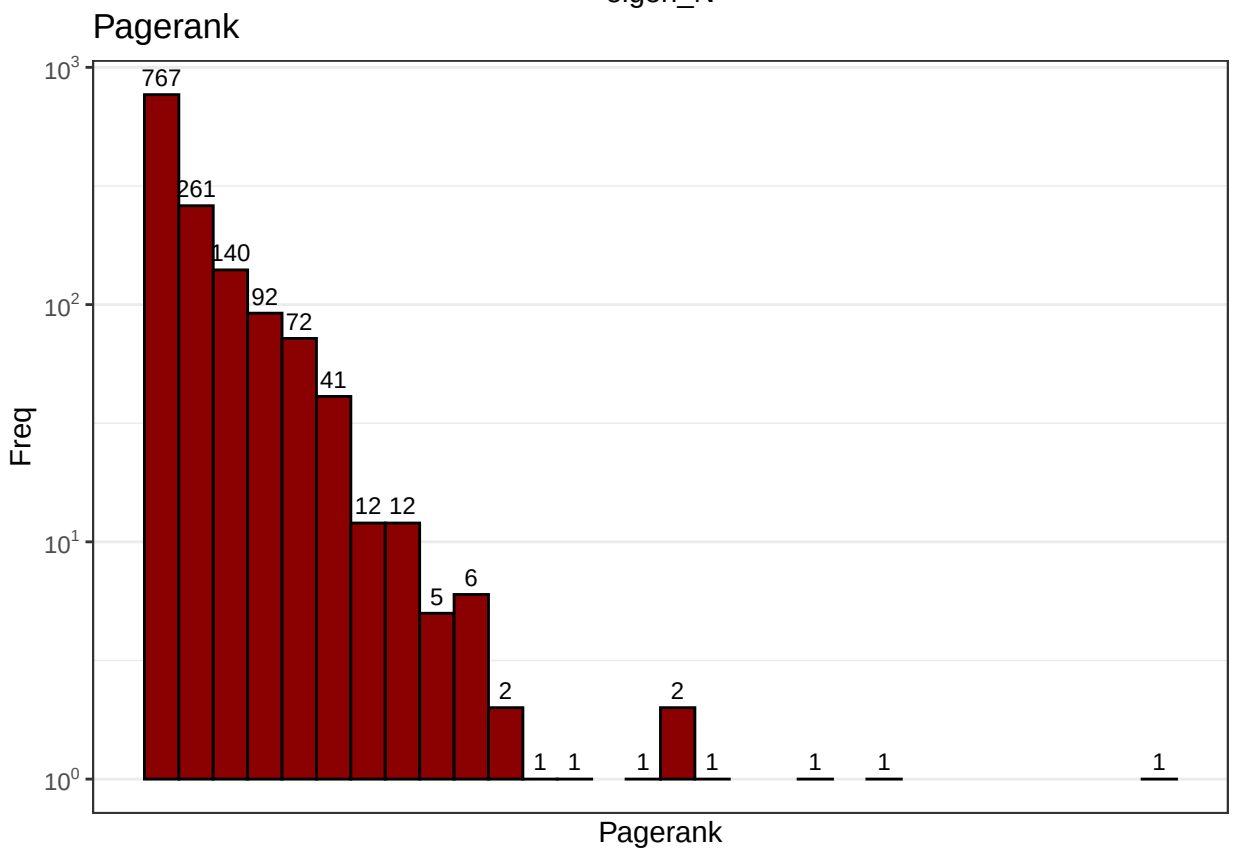


The values of normalized closeness centrality are always between 0 and 1. In this case, values are balanced in the middle of the scale, with just 10 nodes scoring 0.

3.4 Pagerank & Eigenvector Centrality

Pagerank and Eigenvector are two similar measures that are useful to understand how nodes influence other nodes in the network. Nodes with high Eigenvector scores are more likely to have lot of influence on other nodes in the network. The procedure is close to the degree centrality one, but it goes one step further, checking also connections of the nodes connected to the node in exam. A node can have an high degree but low Eigenvector, because it's connections could be with other low scoring nodes. Pagerank, invented by Google, differs from eigenvector because scores nodes on the incoming links, and so it's pretty unrelevant in undirected networks.

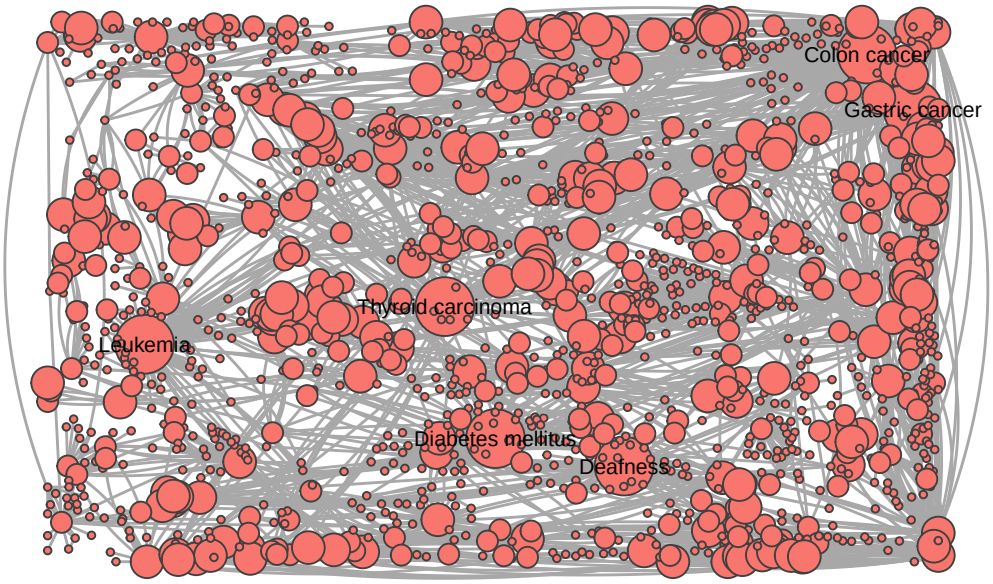




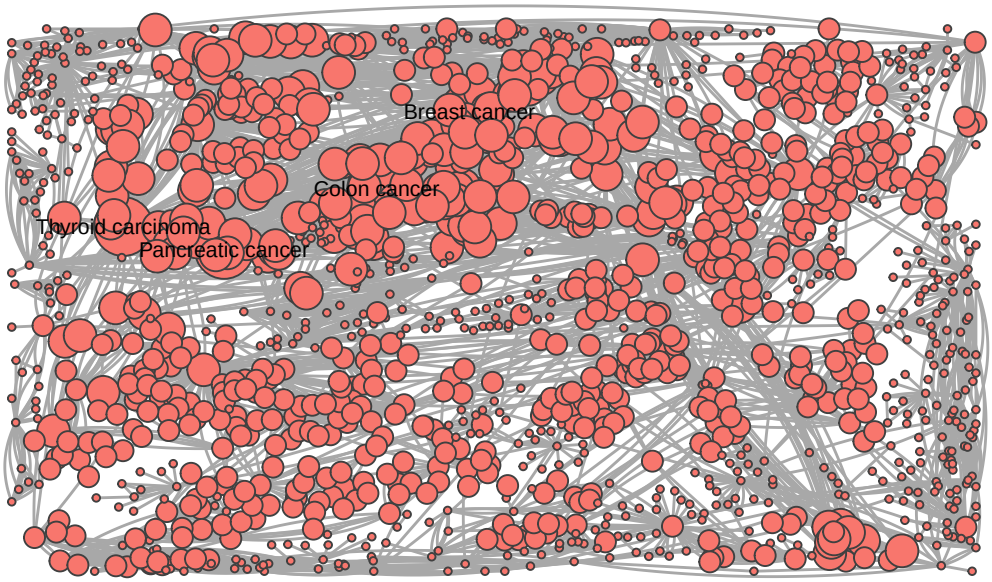
as expected, a lot of nodes are scoring really low on both the measures, and there are just 4-5 nodes that are

reaching high values. It's expected that those nodes will be very important inside the network. Graph is now plotted with node dimensions regulated by those two measures:

Pagerank Centrality



Eigenvector Centrality



Top 5 nodes for Eigenvector and Pagerank are also presented:

```
## [1] "top 10 nodes for Pagerank"
```

```
##  [1] "Colon cancer"       "Deafness"           "Leukemia"
##  [4] "Diabetes mellitus"  "Thyroid carcinoma"  "Gastric cancer"
##  [7] "Breast cancer"      "Pancreatic cancer"  "Mental retardation"
## [10] "Prostate cancer"
```

```
## [1] "top 10 nodes for Eigenvector"
```

```
##  [1] "Colon cancer"          "Breast cancer"         "Thyroid carcinoma"
##  [4] "Pancreatic cancer"     "Hepatic adenoma"       "TP53"
##  [7] "Li-Fraumeni syndrome" "Osteosarcoma"          "Prostate cancer"
## [10] "Gastric cancer"
```

From the results, it's possible to see that a lot of the diseases that have high values for this measures are related to various form of cancer. Colon Cancer takes the firts place for both Pagerank and Eigenvector.

4 Clustering

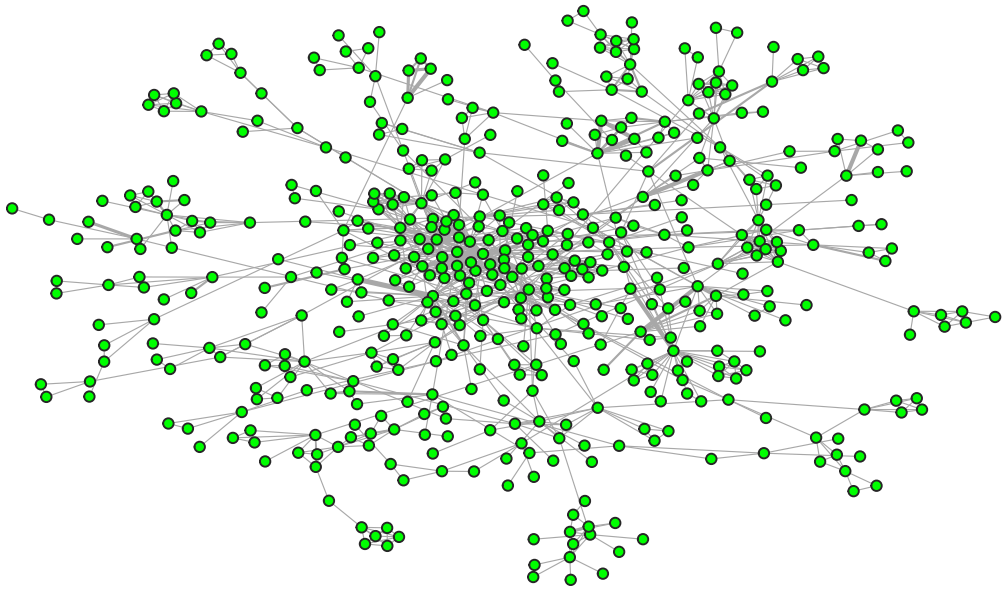
4.1 Clustering Coefficient

Clustering Coefficient is used to measure how dense the network is. High values of Clustering Coefficient means that the network presents a high number of edges, and that is likely to have a lot of clusters in the network

	Values
Local Transitivity Average	0.3126
Global Transitivity	0.251

Values for the network are quite low for both local and global transitivity values, and so evaluating community detection algorithms on this network could be quite difficult. Giving the fact that, for every measure of centrality analyzed, nodes representing genes had always the lowest scores, it's better to remove them from the network, cause probably they are just contributing to create confusion during the clustering phase. Moreover, in this particular network two nodes are connected only if the diseases represented are sharing a common gene, so the presence of the genes itself became redundant. On top of that, edges weight has been tweaked to represent the number of genes shared by the nodes connected by a certain edge and the network has been transformed to an undirected one to facilitate operations.

Human Disease Network With Genes Removed



```
## [1] "Nodes:  516"
```

```
## [1] "Edges:  1188"
```

The network in this state is well manageable and easily visualizable, with less nodes and less edges. It's possible to see that some of the clusters are becoming more visibile. It's also in a better state regarding Clustering Coefficient measures, as the new results are way better than the previous one.

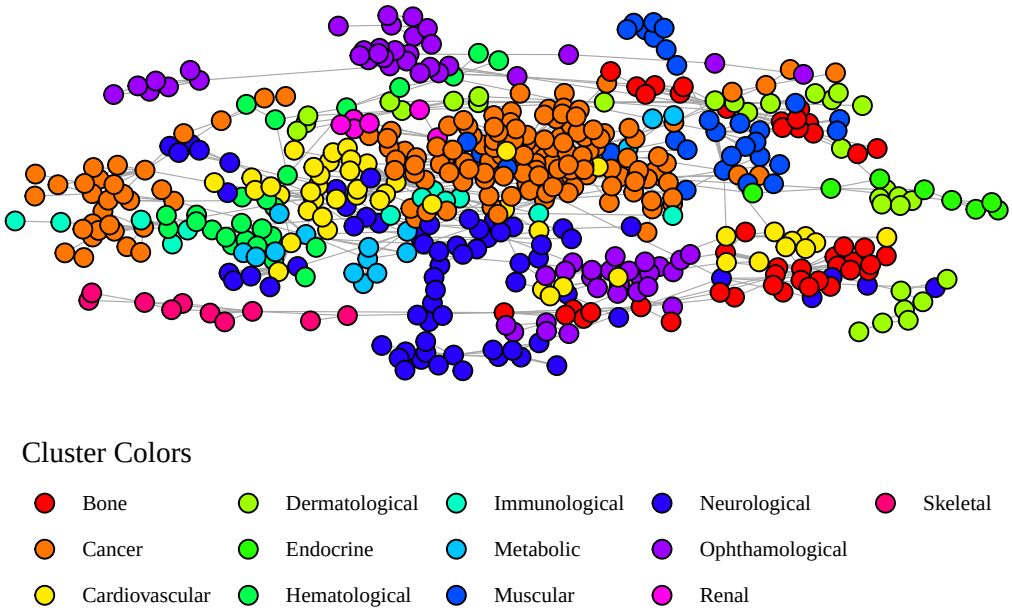
	Values
Local Transitivity Average	0.6358
Global Transitivity	0.4305

4.2 Algorithms

Without any attribute related to nodes, it's impossbile to go for traditional clustering algorithms, and so community detection and graph partitioning ones are the way to go. The chosen algorithms for this project are: - Girvann-Newmann - Louvain - Leiden - Fastgreedy - Markov - Leading Eigenvector Those algortithms have been chosen after looking at some state-of-the-art works on clustering techniques. Prior to show result of the analisys, it's worth noting that every cluster will be labeled via the most frequent label inside of it, and that the final results will be stored in a matrix that will be used for a Groundtruth comparison.

4.3 Girvan-Newmann

Girvan-Newmann algorithm works with the idea that edges that are connecting different clusters of the network should have a high value of edge betweenness. It's considered an Hierarch-centric algorithm




```
## [1] "GN Number Of Communities: 29"
```

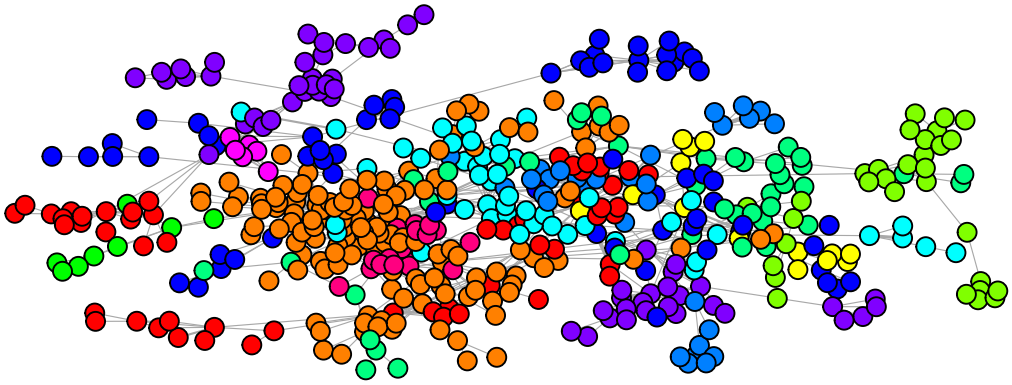
4.4 Louvain and Leiden

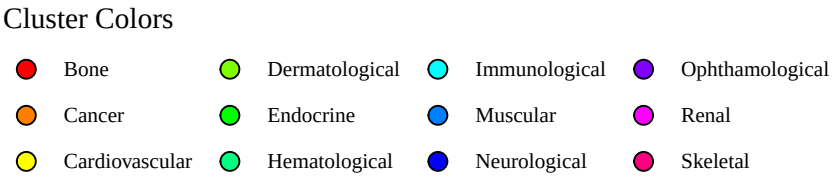
Those two algorithms are presented toghether cause Leiden is considered a natural evolution of Louvain. Louvain is a clustering algorithm based on the concept of modularity maximization. Leiden is a more recent and updated version of this algorithm, way faster and with the possibility of decide pre execution the number

of clusters. It's expected to be the most performant algorithm from those selected for this project for those reasons.

```
## [1] "Louvain Number Of Communities: 27"
```

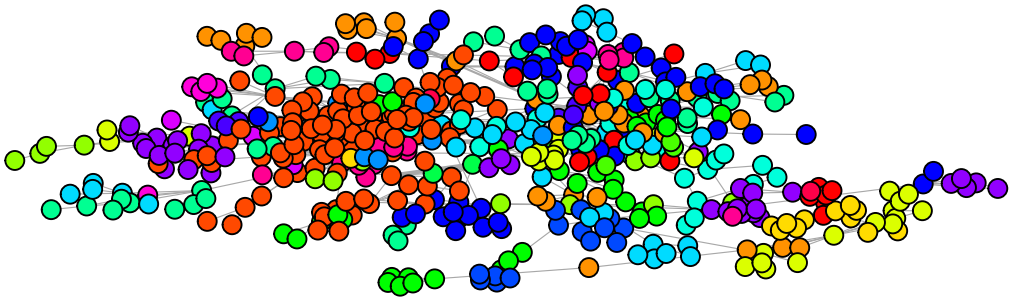
Louvain

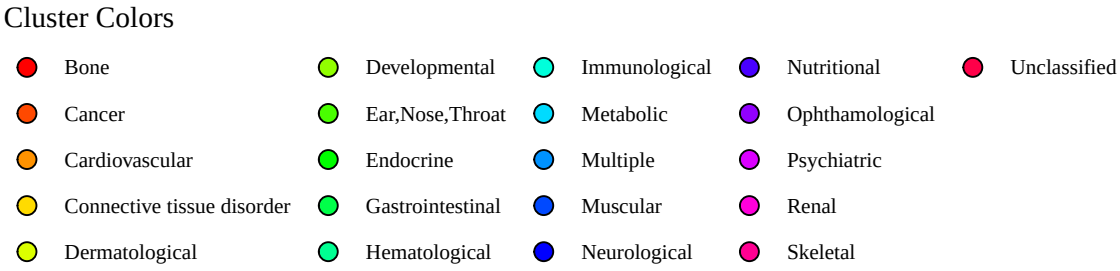





```
## [1] 224
```

Leiden

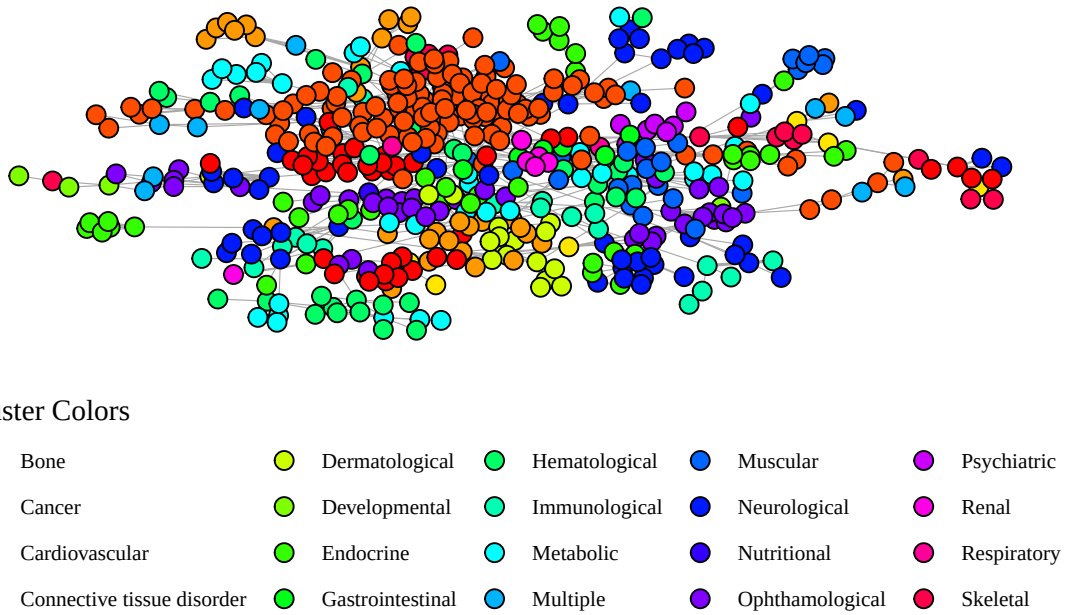




from the results, we can see that the improvement is tangible, with Leiden detecting a lot more cluster than Louvain. This will surely impact on measures like purity

4.5 Markov

```
## [1] "Markov Communities: 169"
```



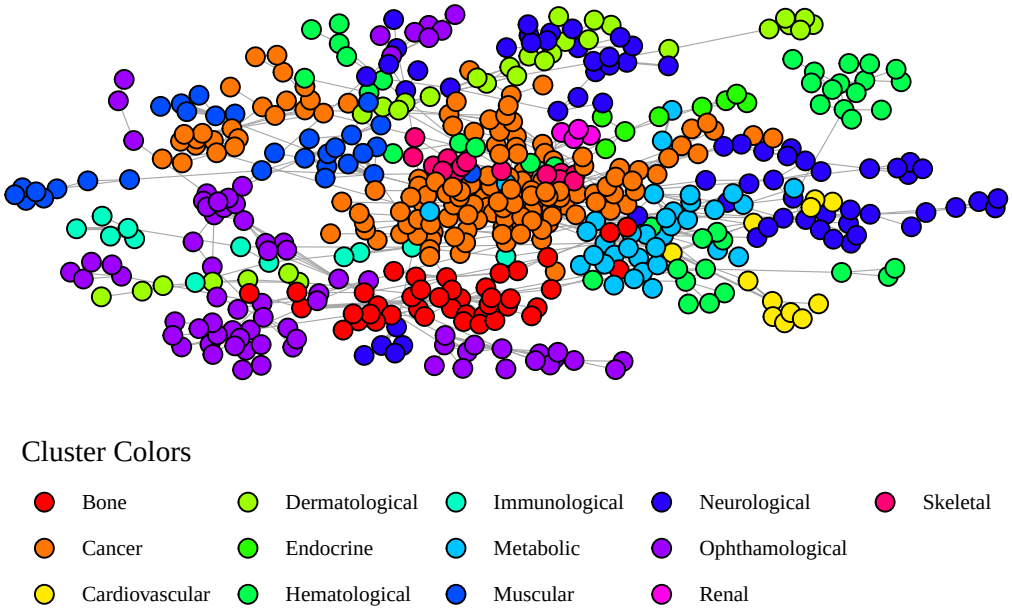
also in this case, it's possible to see that a the number of clusters identifed is way higher than Louvain and Girvann-Newmann: Markov algorithm works on the principle that a random walk in a graph G that visits a cluster will exit from it only when it has visited the majority of his edges.

4.6 Fastgreedy

Fastgreedy is another algorithm that works on Modularity Maximization. It starts by taking every node as a community, and iterating over the network it groups nodes to get to the limit of the modularity.

```
## [1] "Fastgreedy communities:  26"
```

Fastgreedy



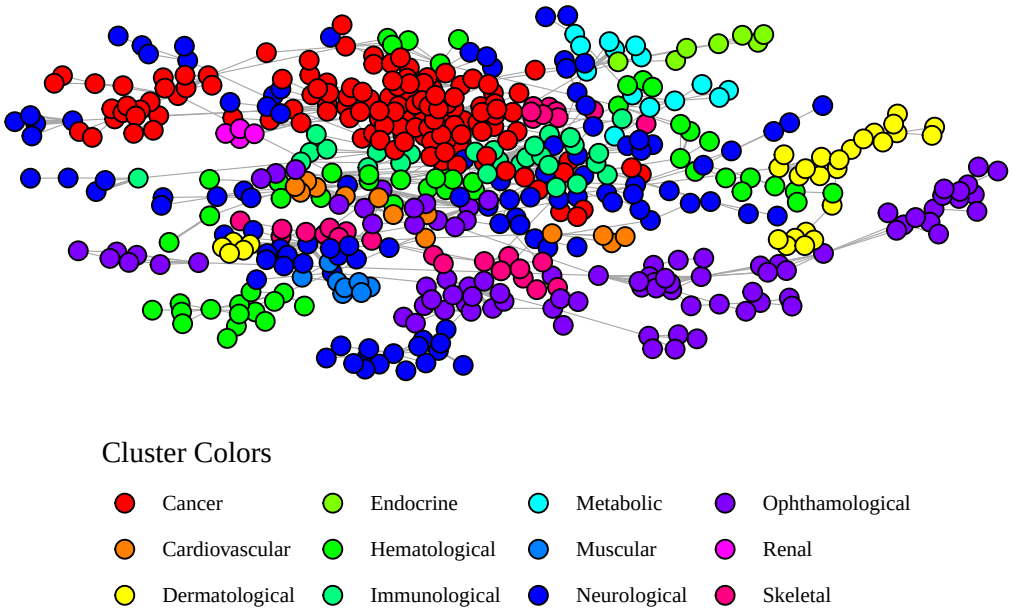
results, as for the other algorithms related to modularity maximization, are pretty low, with just 26 clusters found.

4.7 Leading Eigenvector

Leading Eigenvector is another modularity maximization based algorithm that works on the principle of dividing the network in two components at every iteration, reaching the maximum limit of modularity. As for the other modularity maximization algorithms, it's expected to perform badly because of the difficulty of recognizing clusters with just 1 element

```
## [1] "Leading Eigen Communities:  27"
```

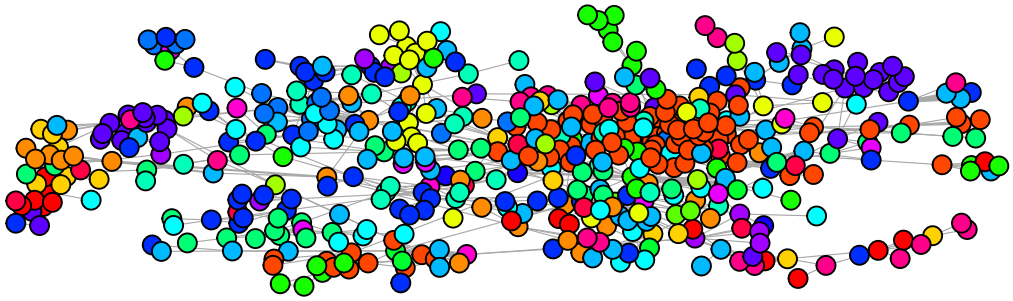
Leading eigenvector Clusters



4.8 Groundtruth

as said before, the network presents 22 different classes of diseases that nodes are associated with. Plotting a graph highlighting those classes can be useful to evidence if nodes belonging to the same class are close to each other.

Groundtruth Clustering



as expected, nodes associated with the cancer class are very close, with few outliers, while for example nodes associated with Nutritional and Neurological classes are more sparse in the network

5 Final Comparison

5.1 Purity

Purity is a measure that indicates how well a community can represent an entire class. In other words, purity indicates how many nodes for a class have been classified correctly in a cluster. Value ranges from 0 to 1, with one only reached if a cluster is completely representing an entire class. Looking and the numbers from the various algorithms used, it's expected that Leiden will have the higher value of purity, because it's detecting even the smallest clusters in the network.

	newmann	louvain	leiden	markov	fastgreedy	lead_eigen	avg
Ear,Nose,Throat	0.0000000	0.0000000	1.0000000	0.0000000	0.0000000	0.0000000	0.16666667
Cancer	0.9636364	0.9339623	0.9897959	0.9514563	0.9629630	0.9411765	0.95716505
Ophthamological	0.8666667	0.8666667	0.9024390	0.9024390	0.9148936	0.9183673	0.89524539
Endocrine	0.2962963	0.2962963	0.8571429	0.7812500	0.2962963	0.2592593	0.46442350
Cardiovascular	0.6562500	0.4137931	0.8235294	0.7500000	0.3571429	0.3928571	0.56559542
Neurological	0.6885246	0.6885246	0.7500000	0.6949153	0.6885246	0.7857143	0.71603389
Hematological	0.4857143	0.5675676	0.8648649	0.7142857	0.5675676	0.6052632	0.63421053
Nutritional	0.0000000	0.0000000	1.0000000	0.3333333	0.0000000	0.0000000	0.22222222
Muscular	0.8947368	0.8947368	0.7500000	1.0000000	0.8947368	0.2666667	0.78347953
Respiratory	0.0000000	0.0000000	0.0000000	0.4000000	0.0000000	0.0000000	0.06666667
Immunological	0.4166667	0.5555556	0.7083333	0.5833333	0.4166667	0.3076923	0.49804131
Dermatological	0.7600000	0.7600000	0.8518519	0.6956522	0.7600000	0.7500000	0.76291734
Psychiatric	0.0000000	0.0000000	0.5000000	0.5555556	0.0000000	0.0000000	0.17592593
Connective tissue disorder	0.0000000	0.0000000	0.5882353	0.3333333	0.0000000	0.0000000	0.15359477
Metabolic	0.2352941	0.0000000	0.7352941	0.6388889	0.2857143	0.2857143	0.36348428
Gastrointestinal	0.0000000	0.0000000	0.4000000	0.6000000	0.0000000	0.0000000	0.16666667
Bone	0.8000000	0.9200000	0.7647059	0.9166667	0.6666667	0.0000000	0.67800654
Skeletal	0.2173913	0.4814815	0.6666667	0.4583333	0.4615385	0.5714286	0.47613997
Renal	0.4545455	0.4545455	0.5000000	0.5000000	0.4000000	0.4000000	0.45151515
Developmental	0.0000000	0.0000000	0.9090909	0.3750000	0.0000000	0.0000000	0.21401515
Purity Measure	0.6143411	0.6065891	0.8139535	0.7480620	0.6124031	0.5813953	0.66279070

as expected, Leiden is scoring 0.81, while especially modularity maximization based algortihms are scoring a maximum of 0.6 in purity measure. Markov is the second top scorer, with a value of 0.74, close to the Leiden one.

5.2 Normalized Mutal Information

Normalized Mutual Information is the normalized version of Mutual Information. Mutual Information measures the mutual dependence between two clusters in the network.

	NMI
newmann	0.4577439
louvain	0.4715064
leiden	0.7254328
markov	0.6419089
fastgreedy	0.4751145
lead_eigen	0.4230969

Leiden has a pretty high value for Mutual Information compared to the other Modularity Maximization

algorithms, scoring above 0.7, Markov came second for this measure too, with a value of 0.64, close to the Leiden one.

5.3 Adjusted Rand Index

The Adjusted Rand Index is the corrected-for-chance version of the Rand index. Such a correction for chance establishes a baseline by using the expected similarity of all pair-wise comparisons between clusterings specified by a random model.

	Adjusted Rand Index
newmann	0.4509450
louvain	0.4595655
leiden	0.7055712
markov	0.5974628
fastgreedy	0.4766206
lead_eigen	0.4140901

even for this measure, Leiden and Markov are still the top performers, scoring values above 0.5 and getting closer to 1.

6 Concusions and Further Developments

The analysis presented proved that finding clusters on this network was a pretty hard task: the majority of the algorithms proposed in this paper scored badly for clustering measures and just Leiden algorithm and Markov Algorithm came close to a good result in finding clusters. The analisys was mainly conducted focusing on old state-of-the-art Modularity Maximization based algorithms, that proved to not be a good choice overall. if a choice has to be made, Leiden is the right algorithm for a task like this one, considering the fact that is a pretty new discovery (2019) and it's an improved version of Louvain algorithm, another famous state-of-the-art algorithm. Further developments on this task include trying other categories of algorithms, like the graph partitioning ones, or maybe try a complete analisys over the original network, without removing genes, to prove the fact that they are just adding more confusion in the clustering process