Group Project

Web Analytics on Diseases

- Introduction
- Data Preprocessing
- Community Detection
- Analysis on each community
- Key player Detection
- Visualizations and Insights in Gephi
- Appendix I and II

Contribution	Data Preprocessing	Community Detection & Analysis	Key player Detection	Visualization
Sun Qianqian (G1700670L)	✓			✓
Saumya Agarwal (G1702632H)		✓	✓	
Ye Jialiang (G1700587H)	✓			✓
Wang Yanhang (G1700683L)		✓	✓	

1. Introduction

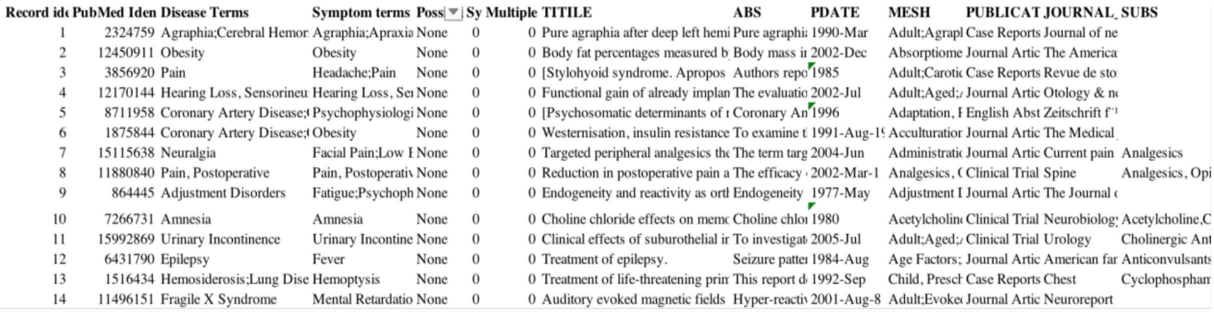
The past decades have brought remarkable advances in our understanding of human disease. While progress on the genetic and proteomic aspects has been impressive, most aspects of the relation between genotype and phenotype still remain unclear, especially for complex diseases.

We use the dataset from 1000 patient's records about the diseases and their corresponding symptoms, in order to analyze the entangled relationships between different diseases: for example, the probability for a patient to get diseases in same communities when already had some symptoms of a certain diseases. Furthermore, the diseases in same community may have some underlying relationships formed by similar problems on genes, organs or protein, and even some human body systems. The analytics can help the biologist and medical scientist to detect these complex relationships and to make some efficient medicines for some disease affected by same protein. And then the community can be exploited to predict gene function or be used to transfer detailed knowledge of model organisms to interpret and predict associated phenomena in human.

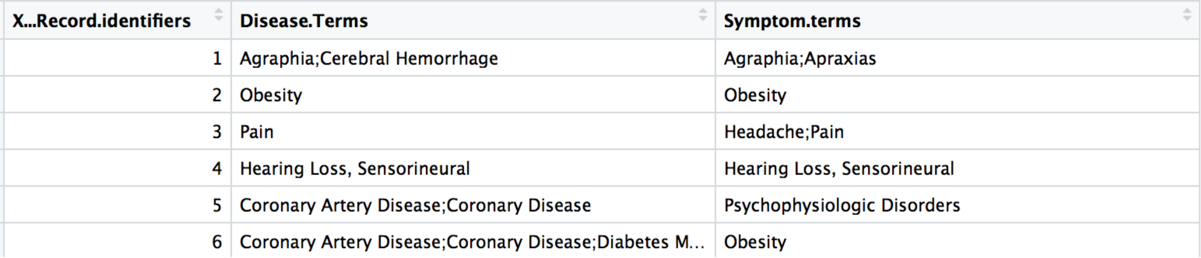
2. Data pre-processing

2.1 data selection

There are 14 attributes in the original file. In our project, we only selected 3 attributes (Record id, Disease term and Symptom term) to make a disease-disease network base on the similarities of diseases' symptoms.



Record id	PubMed Iden	Disease Terms	Symptom terms	Poss	Sy	Multiple	TITILE	ABS	PDATE	MESH	PUBLICAT	JOURNAL_	SUBS
1	2324759	Agraphia;Cerebral Hemor	Agraphia;Apraxia	None	0	0	Pure agraphia after deep left hemi	Pure agraphia	1990-Mar	Adult;Agrapl	Case Reports	Journal of ne	
2	12450911	Obesity	Obesity	None	0	0	Body fat percentages measured b	Body mass ii	2002-Dec	Absorptiome	Journal Artic	The America	
3	3856920	Pain	Headache;Pain	None	0	0	[Stylohyoid syndrome. Apropos	Authors repo	1985	Adult;Carotic	Case Reports	Revue de sto	
4	12170144	Hearing Loss, Sensorineu	Hearing Loss, Sei	None	0	0	Functional gain of already implan	The evaluatio	2002-Jul	Adult;Aged;	Journal Artic	Otology & n	
5	8711958	Coronary Artery Disease;	Psychophysiologi	None	0	0	[Psychosomatic determinants of r	Coronary An	1996	Adaptation, I	English Abst	Zeitschrift f	
6	1875844	Coronary Artery Disease;	Obesity	None	0	0	Westernisation, insulin resistance	To examine t	1991-Aug-1	Acculturation	Journal Artic	The Medical	
7	15115638	Neuralgia	Facial Pain;Low I	None	0	0	Targeted peripheral analgesics th	The term targ	2004-Jun	Administratic	Journal Artic	Current pain	Analgesics
8	11880840	Pain, Postoperative	Pain, Postoperati	None	0	0	Reduction in postoperative pain a	The efficacy	2002-Mar-1	Analgesics, (	Clinical Trial	Spine	Analgesics, Opi
9	864445	Adjustment Disorders	Fatigue;Psychoph	None	0	0	Endogeneity and reactivity as ort	Endogeneity	1977-May	Adjustment I	Journal Artic	The Journal (	
10	7266731	Amnesia	Amnesia	None	0	0	Choline chloride effects on memc	Choline chlo	1980	Acetylcholin	Clinical Trial	Neurobiolog	Acetylcholine,C
11	15992869	Urinary Incontinence	Urinary Incontine	None	0	0	Clinical effects of suburothelial ir	To investigat	2005-Jul	Adult;Aged;	Clinical Trial	Urology	Cholinergic Ant
12	6431790	Epilepsy	Fever	None	0	0	Treatment of epilepsy.	Seizure patte	1984-Aug	Age Factors;	Journal Artic	American fai	Anticonvulsants
13	1516434	Hemosiderosis;Lung Dise	Hemoptysis	None	0	0	Treatment of life-threatening prin	This report d	1992-Sep	Child, Presch	Case Reports	Chest	Cyclophospham
14	11496151	Fragile X Syndrome	Mental Retardatio	None	0	0	Auditory evoked magnetic fields	Hyper-reactiv	2001-Aug-8	Adult;Evoke	Journal Artic	Neuroreport	



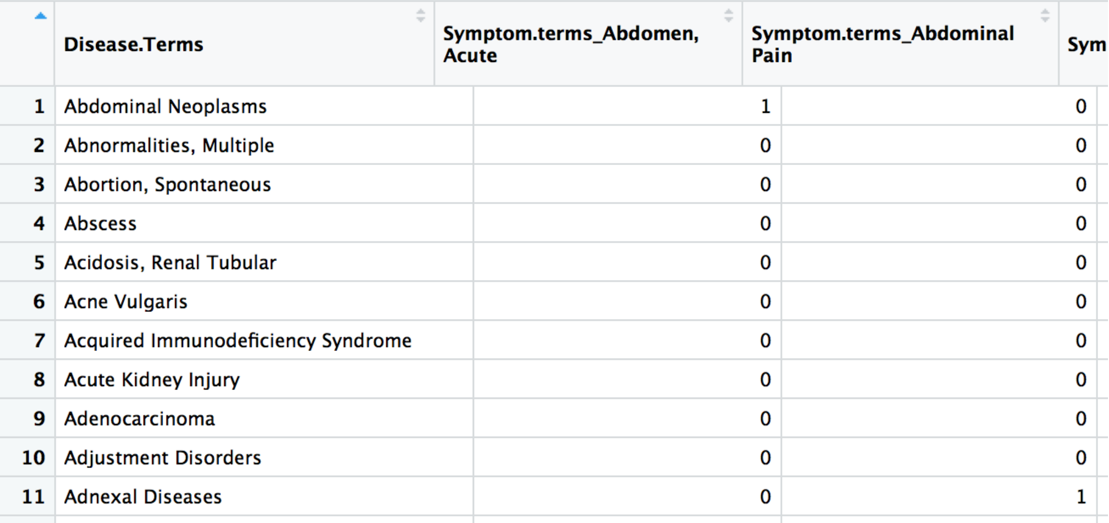
X...Record.identifiers	Disease.Terms	Symptom.terms
1	Agraphia;Cerebral Hemorrhage	Agraphia;Apraxias
2	Obesity	Obesity
3	Pain	Headache;Pain
4	Hearing Loss, Sensorineural	Hearing Loss, Sensorineural
5	Coronary Artery Disease;Coronary Disease	Psychophysiologic Disorders
6	Coronary Artery Disease;Coronary Disease;Diabetes M...	Obesity

Then, we want to obtain the network of diseases to do the further analysis.

2.2 One-hot Encoding

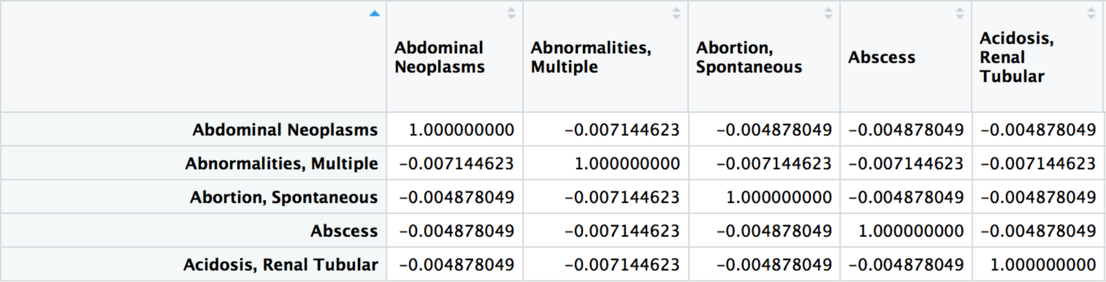
Then we want to calculate the similarities on diseases based on their corresponding symptoms, but the format of the original symptoms was quite hard to deal with. To solve this problem, we decided to use One-hot Encoding to show the existence or nonexistence of symptoms for each disease, that means, if one certain symptom was shown up for a certain disease, in the row which represents the disease we mark the exist symptom as "1", and if this symptom was not shown up for that disease, we mark the value as "0".

We first split the diseases and symptoms by ";". After that each row will only have one disease with one symptom. We use 0 and 1 to represent the symptom.



	Disease.Terms	Symptom.terms_Abdomen, Acute	Symptom.terms_Abdominal Pain	Sym
1	Abdominal Neoplasms	1	0	
2	Abnormalities, Multiple	0	0	
3	Abortion, Spontaneous	0	0	
4	Abscess	0	0	
5	Acidosis, Renal Tubular	0	0	
6	Acne Vulgaris	0	0	
7	Acquired Immunodeficiency Syndrome	0	0	
8	Acute Kidney Injury	0	0	
9	Adenocarcinoma	0	0	
10	Adjustment Disorders	0	0	
11	Adnexal Diseases	0	1	

Then we sum the "1" for each disease. There are 698 diseases and 206 symptoms. Using the vector to represent each disease, we calculated the similarities between each pair of diseases and obtained the similarity matrix.

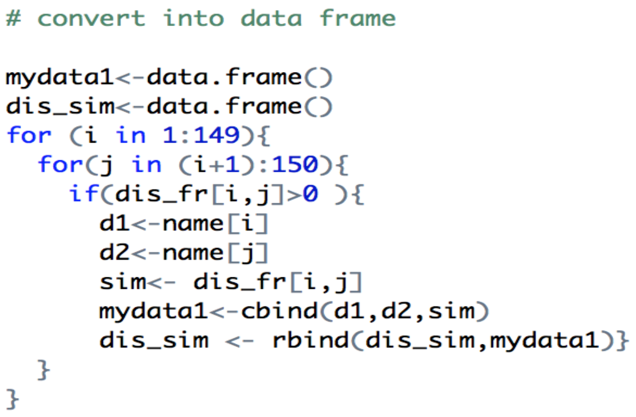


	Abdominal Neoplasms	Abnormalities, Multiple	Abortion, Spontaneous	Abscess	Acidosis, Renal Tubular
Abdominal Neoplasms	1.000000000	-0.007144623	-0.004878049	-0.004878049	-0.004878049
Abnormalities, Multiple	-0.007144623	1.000000000	-0.007144623	-0.007144623	-0.007144623
Abortion, Spontaneous	-0.004878049	-0.007144623	1.000000000	-0.004878049	-0.004878049
Abscess	-0.004878049	-0.007144623	-0.004878049	1.000000000	-0.004878049
Acidosis, Renal Tubular	-0.004878049	-0.007144623	-0.004878049	-0.004878049	1.000000000

There is a matrix of 689 rows and 689 columns. Because the Symmetry, we only considered half of the matrix, which contains 243951 records. However, there were still too many records and would take a long time to rewrite into a standard form. So here we decided to only consider 150 diseases.

2.3 Rewrite into standard form

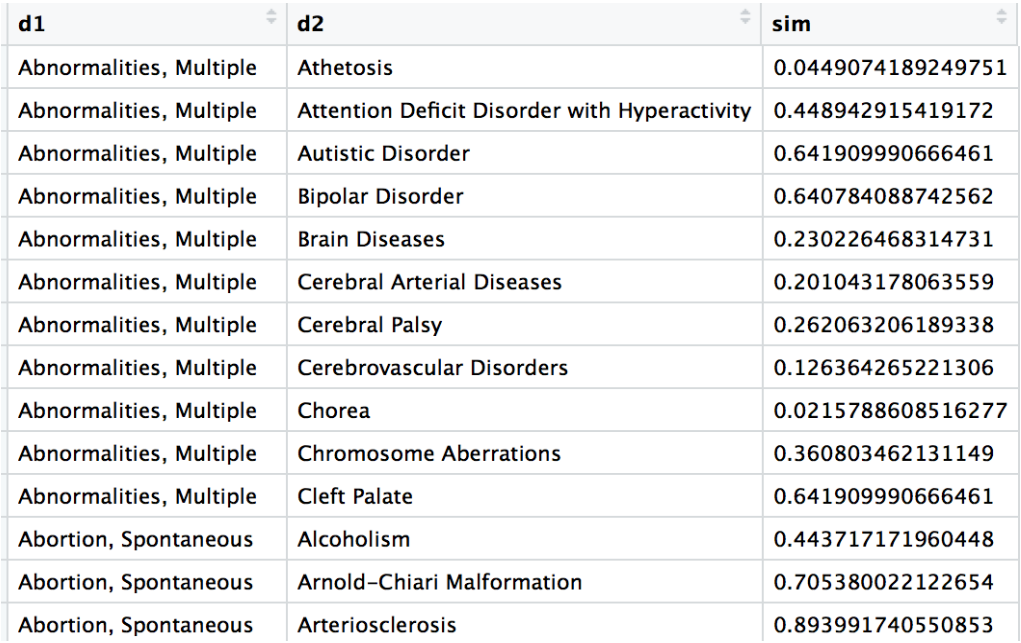
Firstly, we set two data frames and use a loop to rewrite the matrix into a source-target-weight table which can easily apply to the community detection methods. Here we ignored the negative similarity and only selected the positive similarities. Finally, we got 503 rows of data which represent 503 network edges, and put the results into a csv file.



```
# convert into data frame

mydata1<-data.frame()
dis_sim<-data.frame()
for (i in 1:149){
  for(j in (i+1):150){
    if(dis_fr[i,j]>0 ){
      d1<-name[i]
      d2<-name[j]
      sim<- dis_fr[i,j]
      mydata1<-cbind(d1,d2,sim)
      dis_sim <- rbind(dis_sim,mydata1)}
  }
}
```

The content of this csv file is list below:



d1	d2	sim
Abnormalities, Multiple	Athetosis	0.0449074189249751
Abnormalities, Multiple	Attention Deficit Disorder with Hyperactivity	0.448942915419172
Abnormalities, Multiple	Autistic Disorder	0.641909990666461
Abnormalities, Multiple	Bipolar Disorder	0.640784088742562
Abnormalities, Multiple	Brain Diseases	0.230226468314731
Abnormalities, Multiple	Cerebral Arterial Diseases	0.201043178063559
Abnormalities, Multiple	Cerebral Palsy	0.262063206189338
Abnormalities, Multiple	Cerebrovascular Disorders	0.126364265221306
Abnormalities, Multiple	Chorea	0.0215788608516277
Abnormalities, Multiple	Chromosome Aberrations	0.360803462131149
Abnormalities, Multiple	Cleft Palate	0.641909990666461
Abortion, Spontaneous	Alcoholism	0.443717171960448
Abortion, Spontaneous	Arnold-Chiari Malformation	0.705380022122654
Abortion, Spontaneous	Arteriosclerosis	0.893991740550853

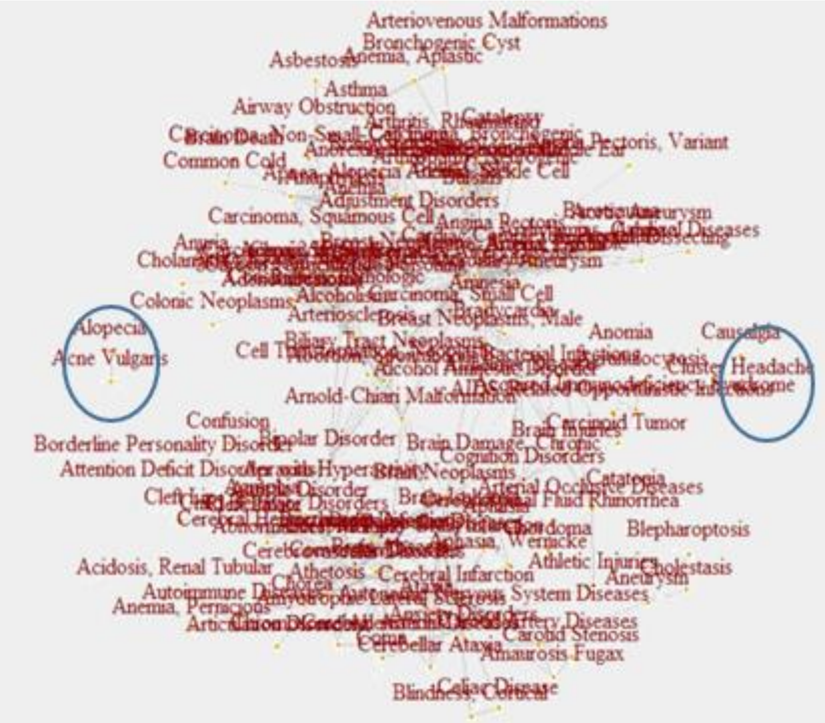
3. Community Detection

In the study of networks, a network is said to have community structure if the nodes of the network can be easily grouped into (potentially overlapping) sets of nodes such that each set of nodes is densely connected internally. In the particular case of non-overlapping community finding, this implies that the network divides naturally into groups of nodes with dense connections internally and sparser connections between groups. But overlapping communities are also allowed.

Finding an underlying community structure in a network, if it exists, is important for a number of reasons. Communities allow us to create a large-scale map of a network and makes its study easier. Being able to identify these sub-structures within a network can provide insight into how network function and topology affect each other. A very important reason that makes communities important is that they often have very different properties than the average properties of the networks. Thus, only concentrating on the average properties usually misses many important and interesting features inside the networks.

Finally, an important application that community detection has found in network science is the prediction of missing links and the identification of false links in the network. During the measurement process, some links may not get observed for a number of reasons. Similarly, some links could falsely enter into the data because of the errors in the measurement. Both these cases are well handled by community detection algorithm since it allows one to assign the probability of existence of an edge between a given pair of nodes.

Below is a picture of our Disease Network. We started with 150 diseases and after removing the diseases with negative similarity values, we are left with 127 connected nodes and 4 unconnected ones (highlighted).



3.1 Basic Data Exploration and Graph Characteristics

Vertex: A vertex (plural vertices) or node is the fundamental unit of which graphs are formed.

Transitivity: A property very important in social networks, and to a lesser degree in other networks, is transitivity. If refers to the extent to which the relation that relates two nodes in a network that are connected by an edge is transitive. Perfect transitivity implies that, if *x* is connected (through an edge) to *y*, and *y* is connected to *z*, then *x* is connected to *z* as well.

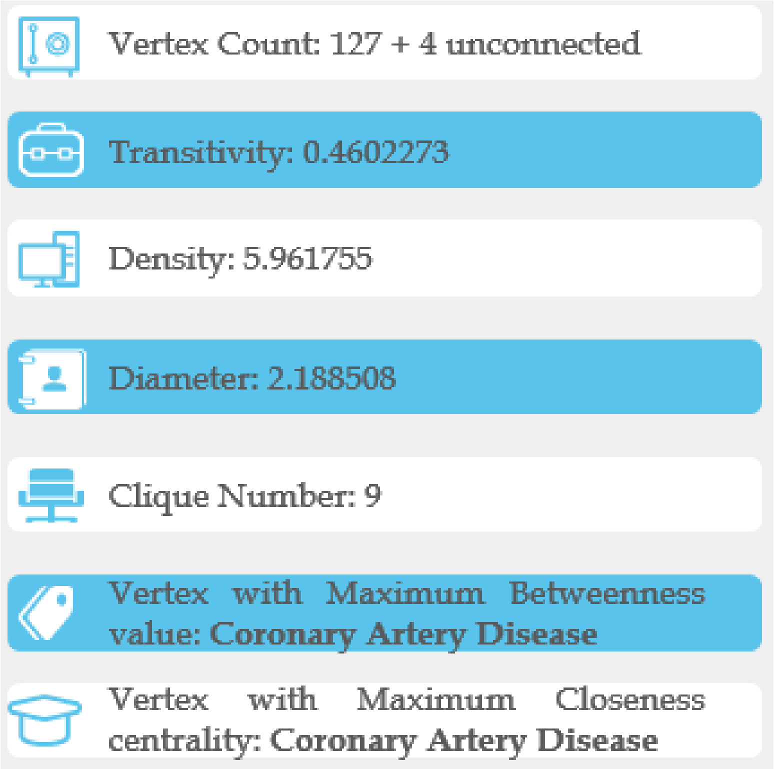
Density: The proportion of present edges from all possible edges in the network.

Diameter: A network diameter is the longest geodesic distance (length of the shortest path between two nodes) in the network.

Clique: A clique is a subset of vertices of an undirected graph such that every two distinct vertices in the clique are adjacent; that is, its induced subgraph is complete

Betweenness centrality: It is an indicator of a node's centrality in a network. It is equal to the number of shortest paths from all vertices to all others that pass through that node.

Closeness Centrality: Closeness is a measure of the degree to which an individual is near all other individuals in a network. It is the inverse of the sum of the shortest distances between each node and every other node in the network. Closeness is the reciprocal of farness



Vertex Count: 127 + 4 unconnected

Transitivity: 0.4602273

Density: 5.961755

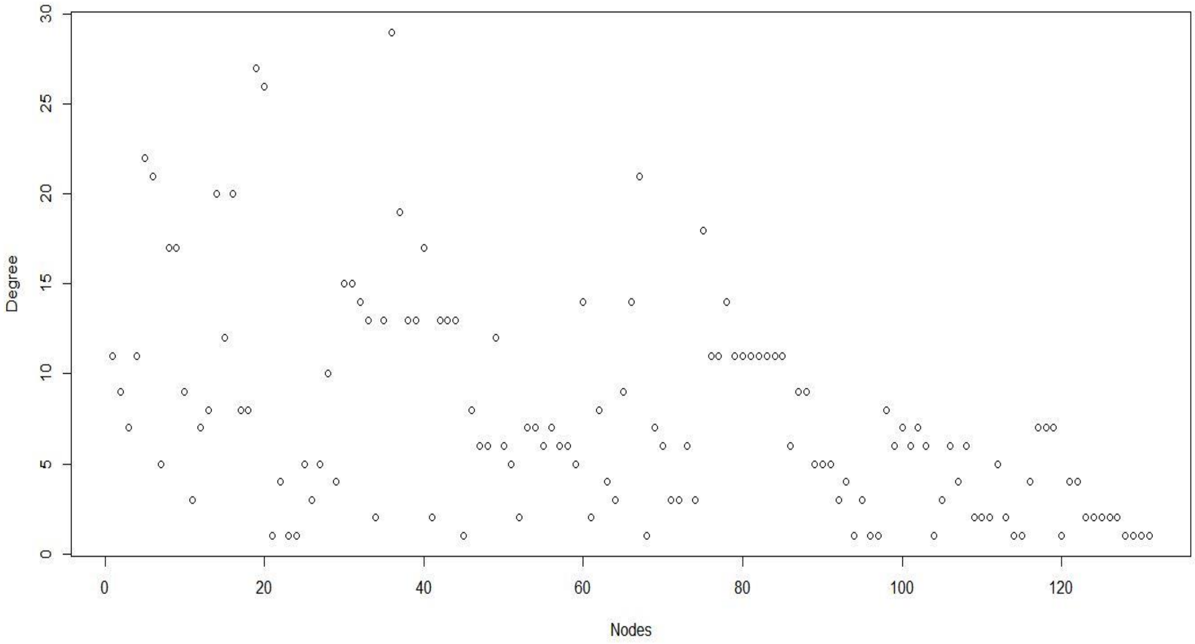
Diameter: 2.188508

Clique Number: 9

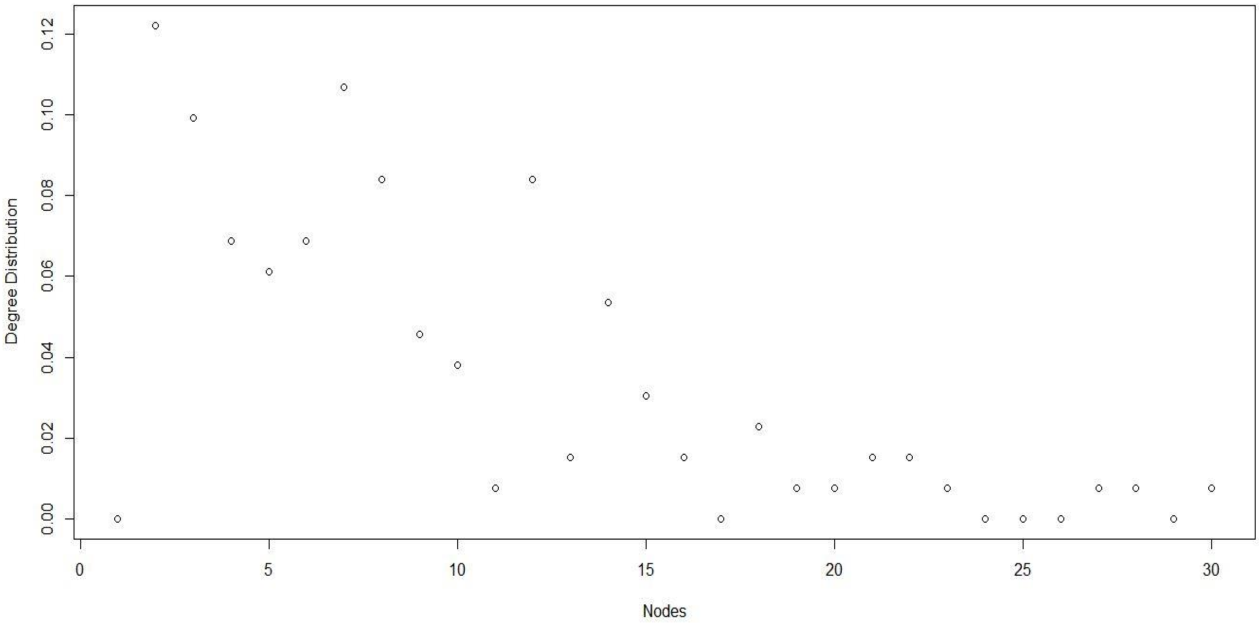
Vertex with Maximum Betweenness value: Coronary Artery Disease

Vertex with Maximum Closeness centrality: Coronary Artery Disease

Degree: The degree dv of a vertex v, in a network graph G = (V,E), counts the number of edges in E incident upon v. We can see that most of the nodes have degree in the range of 0-10.



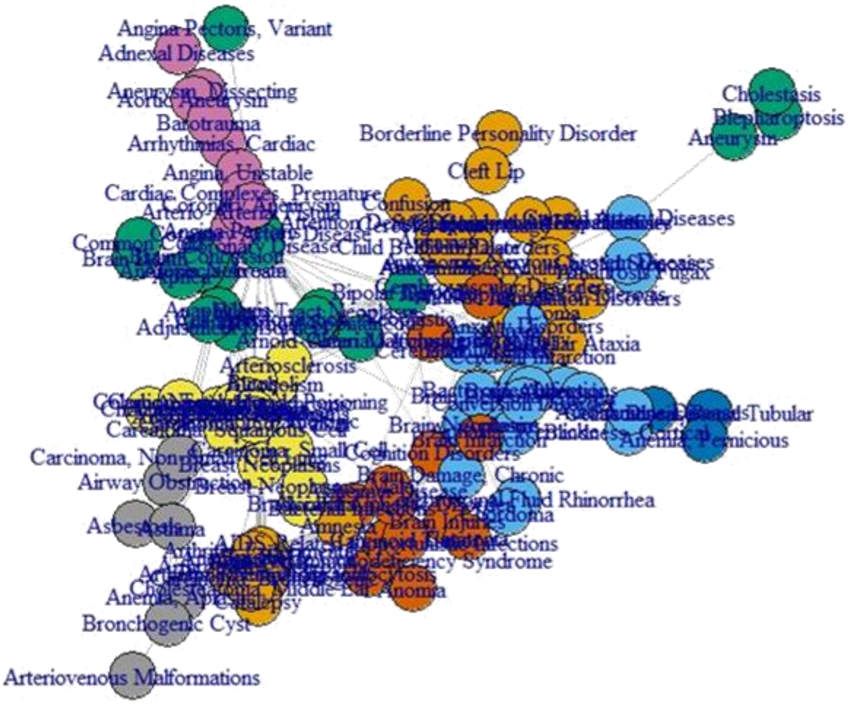
Degree Distribution: Given a network graph G, we define fd to be the fraction of vertices v ∈ V with degree dv = d. The collection {fd}d≥0 is called the degree distribution of G, and is simply a rescaling of the set of degree frequencies.



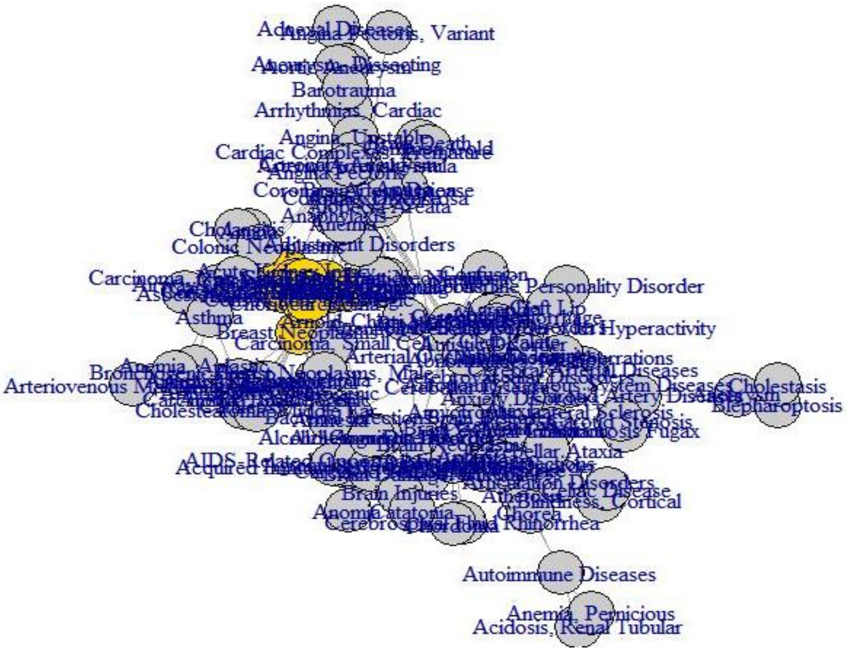
3.2 Clique Based Community Detection

A clique, C, in an undirected graph G = (V, E) is a subset of the vertices, C ⊆ V, such that every two distinct vertices are adjacent. This is equivalent to the condition that the induced subgraph of G induced by C is a complete graph.

All the cliques present in the network are as follows:



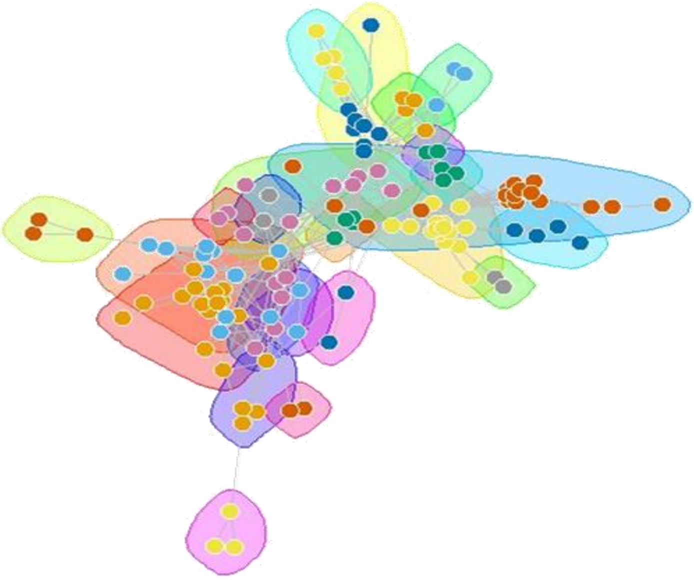
The Largest clique found in the Disease network is highlighted in yellow below:



3.3 Label propagation community Detection

The label propagation algorithm uses an iterative process to find stable communities in a graph. The method begins by giving each node in the graph a unique label. Then, the algorithm iteratively simulates a process in which each node in the graph adopts the label most common amongst its neighbours. The process repeats until the label of every node in the graph is the same as the label of maximum occurrence amongst its neighbors.

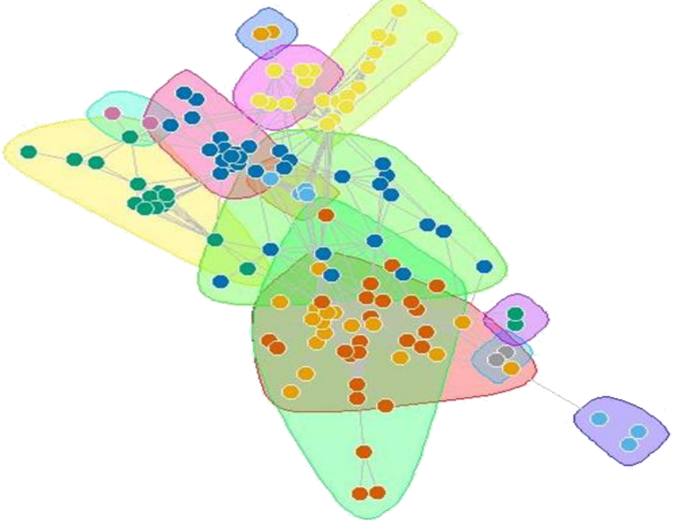
This method gave us a modularity of ***0.6769024***.



3.4 Leading Eigenvector Community Detection

The leading eigenvector method works by calculating the eigenvector of the modularity matrix for the largest positive eigenvalue and then separating vertices into two community based on the sign of the corresponding element in the eigenvector. If all elements in the eigenvector are of the same sign that means that the network has no underlying community structure.

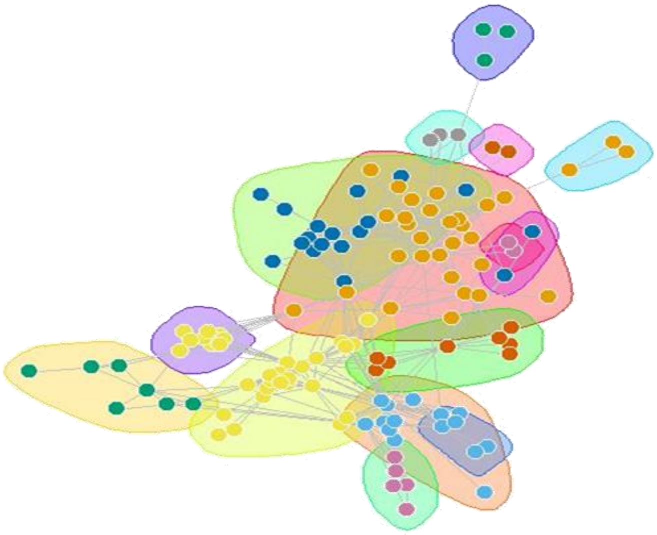
This method gave us a modularity of ***0.6743737.***



3.5 Walktrap Community Detection

This algorithm finds densely connected subgraphs by performing random walks. The idea is that random walks will tend to stay inside communities instead of jumping to other communities.

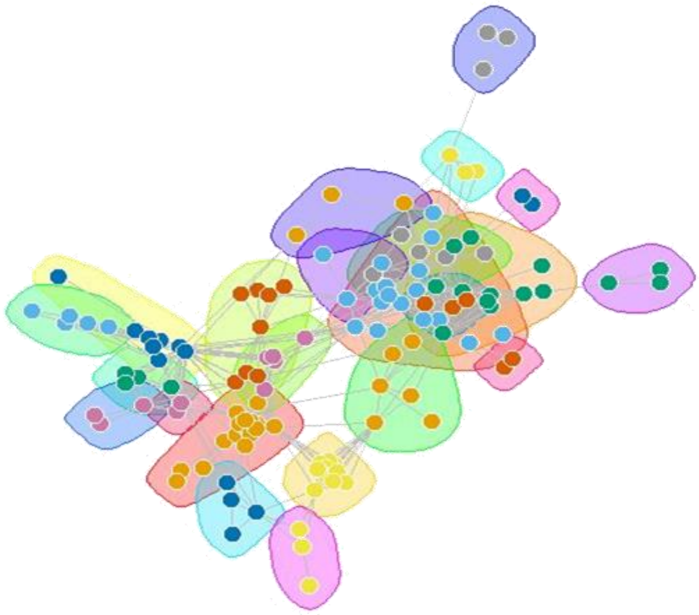
This method gave us a modularity of ***0.6910088.***



3.6 Infomap Community Detection

The Infomap algorithm is based on the principles of information theory. Infomap characterizes the problem of finding the optimal clustering of a graph as the problem of finding a description of minimum information of a random walk on the graph. The algorithm maximizes an objective function called the Minimum Description Length, and in practice an acceptable approximation to the optimal solution can be found quickly.

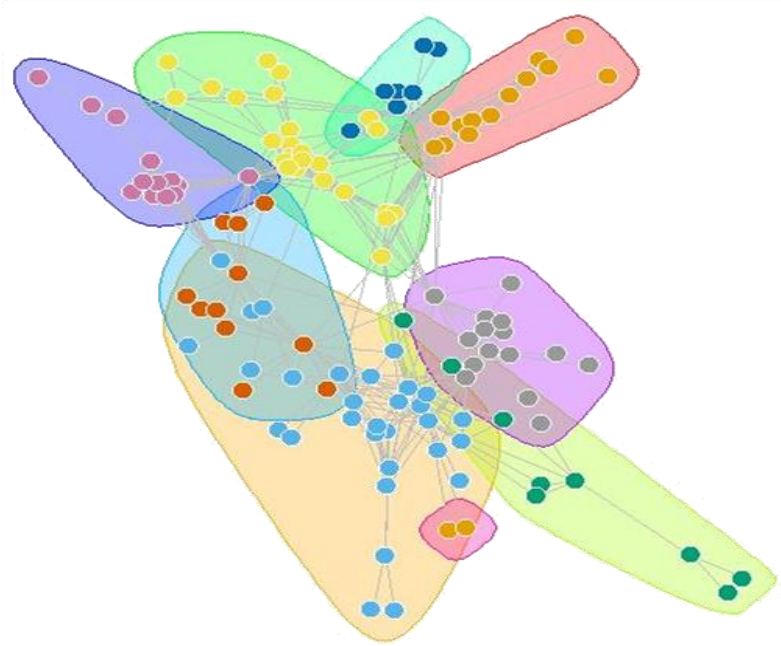
This method gave us a modularity of ***0.6810401.***



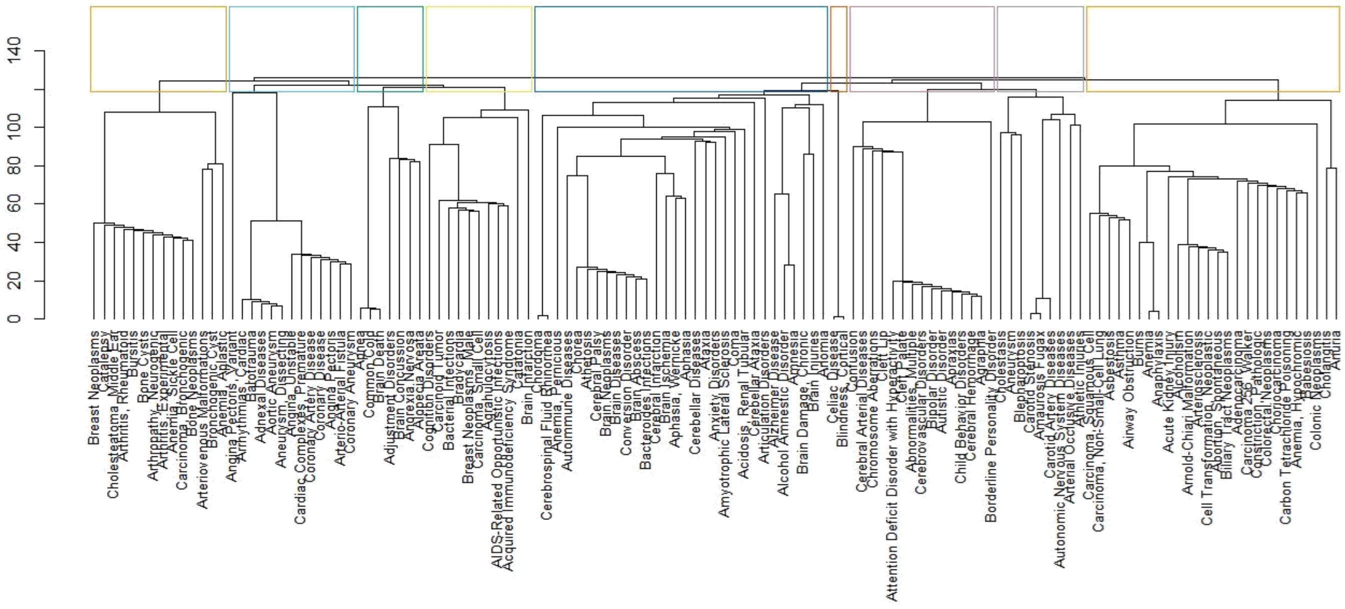
3.7 FastGreedy Algorithm

In this case the algorithm is agglomerative. At each step two groups merge. The merging is decided by optimizing modularity. This is a fast algorithm but has the disadvantage of being a greedy algorithm. Thus, is might not produce the best overall community partitioning.

This method gave us a modularity of ***0.682153***.



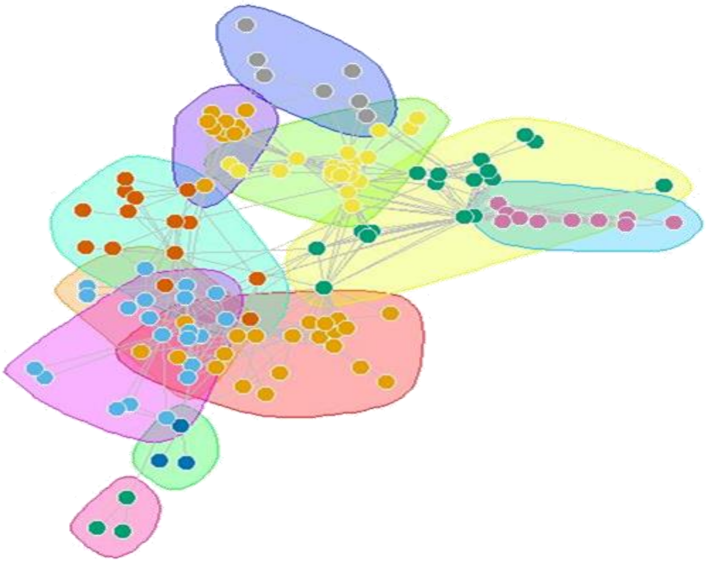
The dendogram for the same is as follows:



3.8 Edge Betweenness Community Detection

Vertex betweenness is an indicator of highly central nodes in networks. It is defined as the number of shortest paths between pairs of nodes that run through it. It is relevant to models where the network modulates transfer of goods between known start and end points, under the assumption that such transfer seeks the shortest available route.

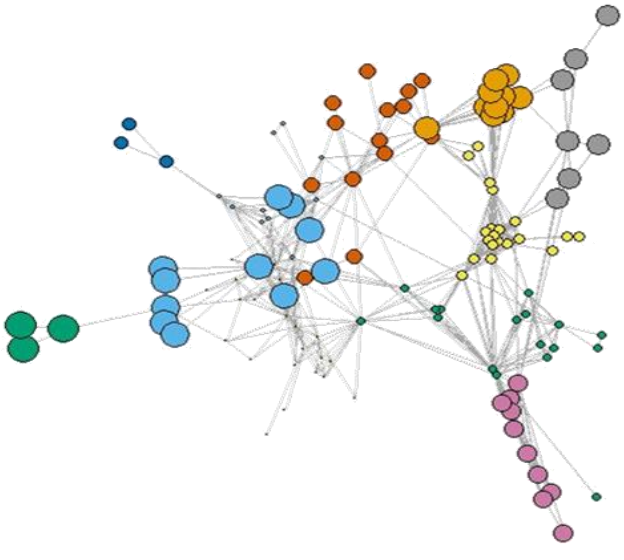
This method gave us a modularity of ***0.6322804***.



3.9 Girvan Newman Community Detection

The Girvan–Newman algorithm extends the definition of "edge-betweenness“ by assigning equal weight to each path such that the total weight of all of the paths is equal to unity. If a network contains loosely connected group, then all shortest paths between communities must go along one of these few edges. Thus, the edges connecting communities will have high edge betweenness. By removing these edges, the groups are separated from one another.

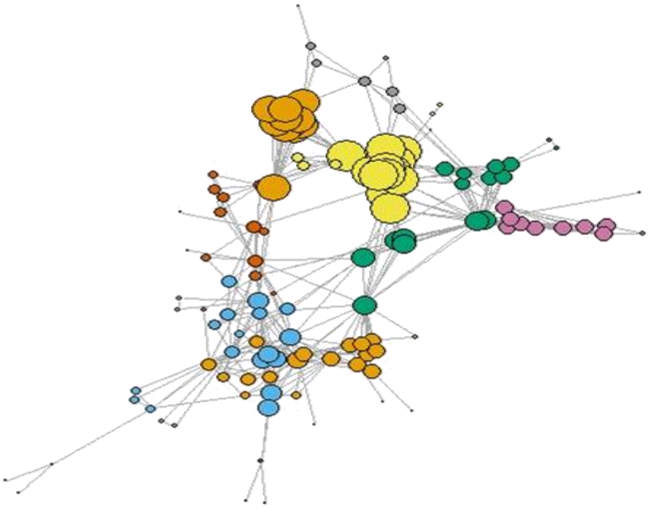
This method gave us a modularity of ***0.5729917***.



3.10 K-Core Decomposition Community Detection

K-core of a graph G is the largest induced subgraph of G in which every vertex has degree at least k. The coreness of a vertex v in G is the largest value of k such that there is a k-core of G containing v. In the k-core decomposition problem, the goal is to compute the coreness of each vertex in G.

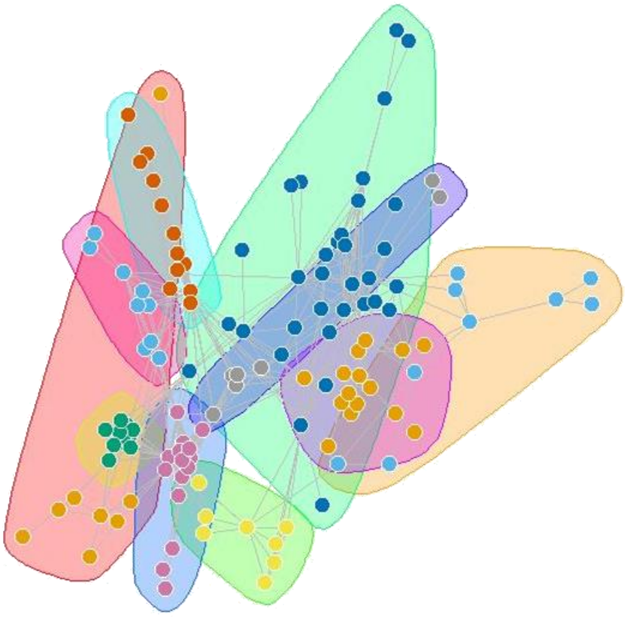
This method gave us a modularity of ***0.4544663***.



3.11 Spinglass Community Dtection

This algorithm uses as spin-glass model and simulated annealing to find the communities inside a network. The community structure of the network is interpreted as the spin configuration that minimizes the energy of the spin glass with the spin states being the community indices. Simulated Annealing is a probabilistic method to optimize the modularity function. We elucidate the properties of the ground state configuration to give a concise definition of communities as cohesive subgroups in networks that is adaptive to the specific class of network under study.

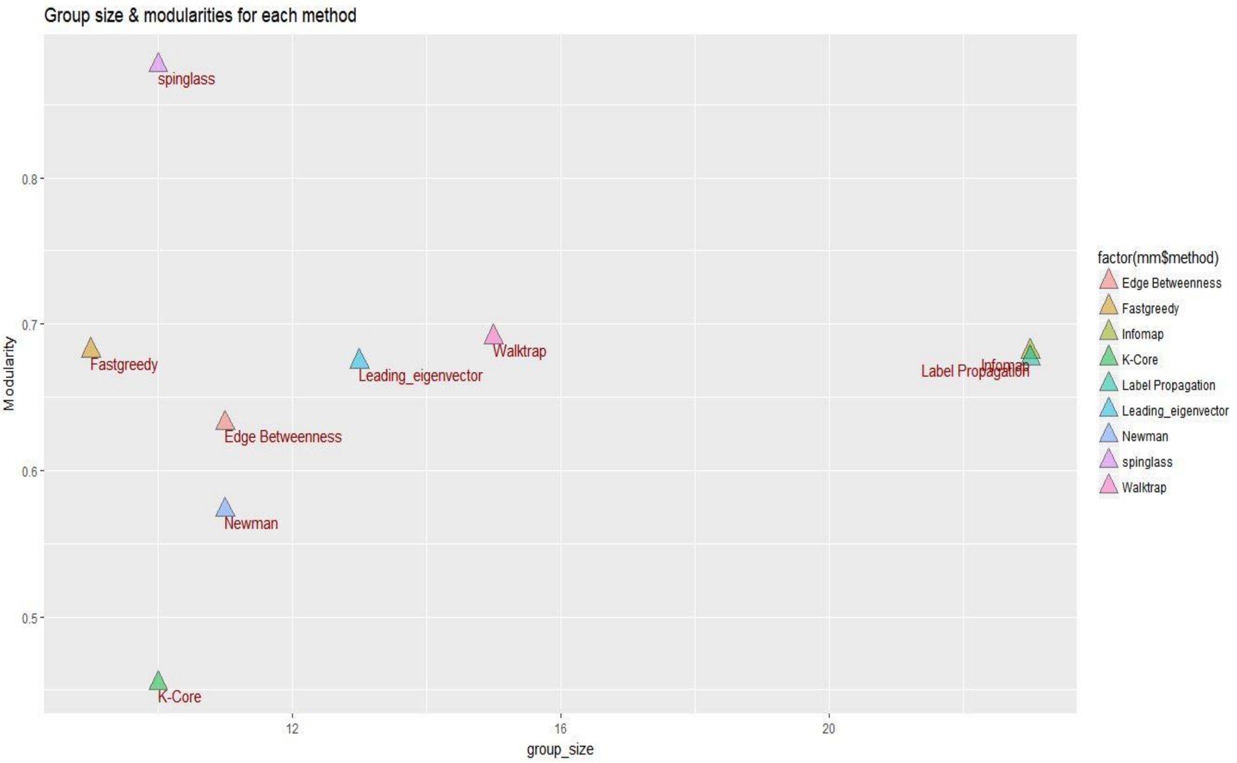
This method gave us a modularity of ***0.8768559***.



3.12 Comparison of Different Algorithms

Modularity is one measure of the structure of networks or graphs. It was designed to measure the strength of division of a network into modules. However, it has been shown that modularity suffers a resolution limit and, therefore, it is unable to detect small communities. Biological networks, including animal brains, exhibit a high degree of modularity.

Modularity is the fraction of the edges that fall within the given groups minus the expected fraction if edges were distributed at random. The value of the modularity lies in the range [−1/2,1). It is positive if the number of edges within groups exceeds the number expected on the basis of chance. For a given division of the network's vertices into some modules, modularity reflects the concentration of edges within modules compared with random distribution of links between all nodes regardless of modules. **High modularity for a partitioning reflects dense connections within communities and sparse connections across communities.**



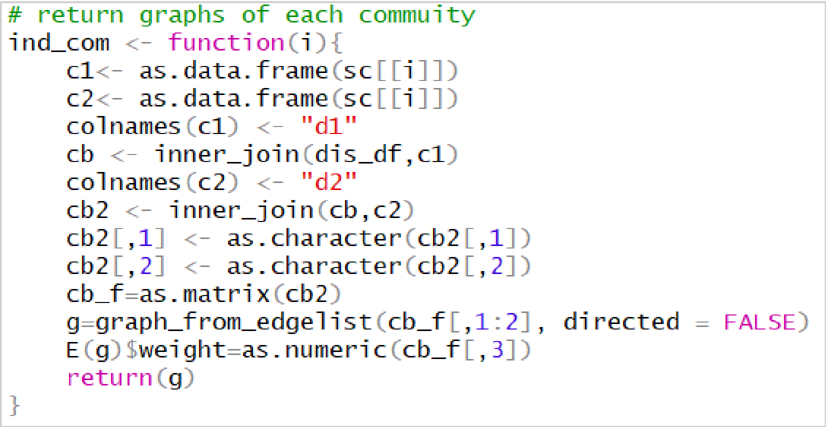
Since the Spin-glass Algorithm gave us the highest Modularity (0.87), we decided to go with its results to further Community Analysis and Key Player detection.

Although the Spin-glass Algorithm gave us the best results in this case, we cannot claim that this is the best algorithm for community detection. The best answer to this question is that the community detection algorithm depends largely on the domain that you are working with and the properties of the data being clustered. There could be some inherent properties of the algorithm that might help narrow down the choices.

4. Analysis on each community

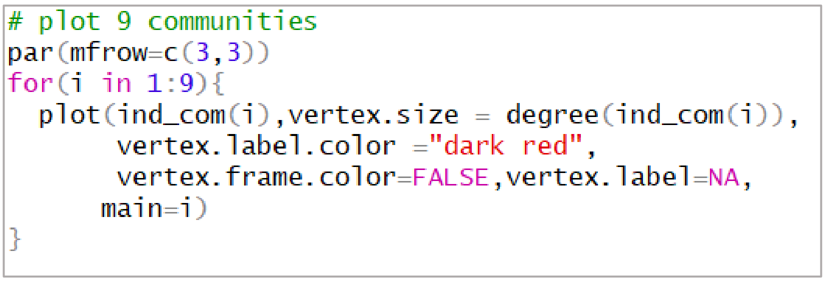
From all the community detection methods we have used, it was not difficult to find the one gave the best result was the Springlass algorithm, which reached the high modularity of 0.877 and also kept its size of number of communities of 9. So, in this part we would use the results we obtained from the Springlass algorithm to do analysis.

At first step, we made a function **inc_com(i)** which is to return the individual communities as individual graphs:



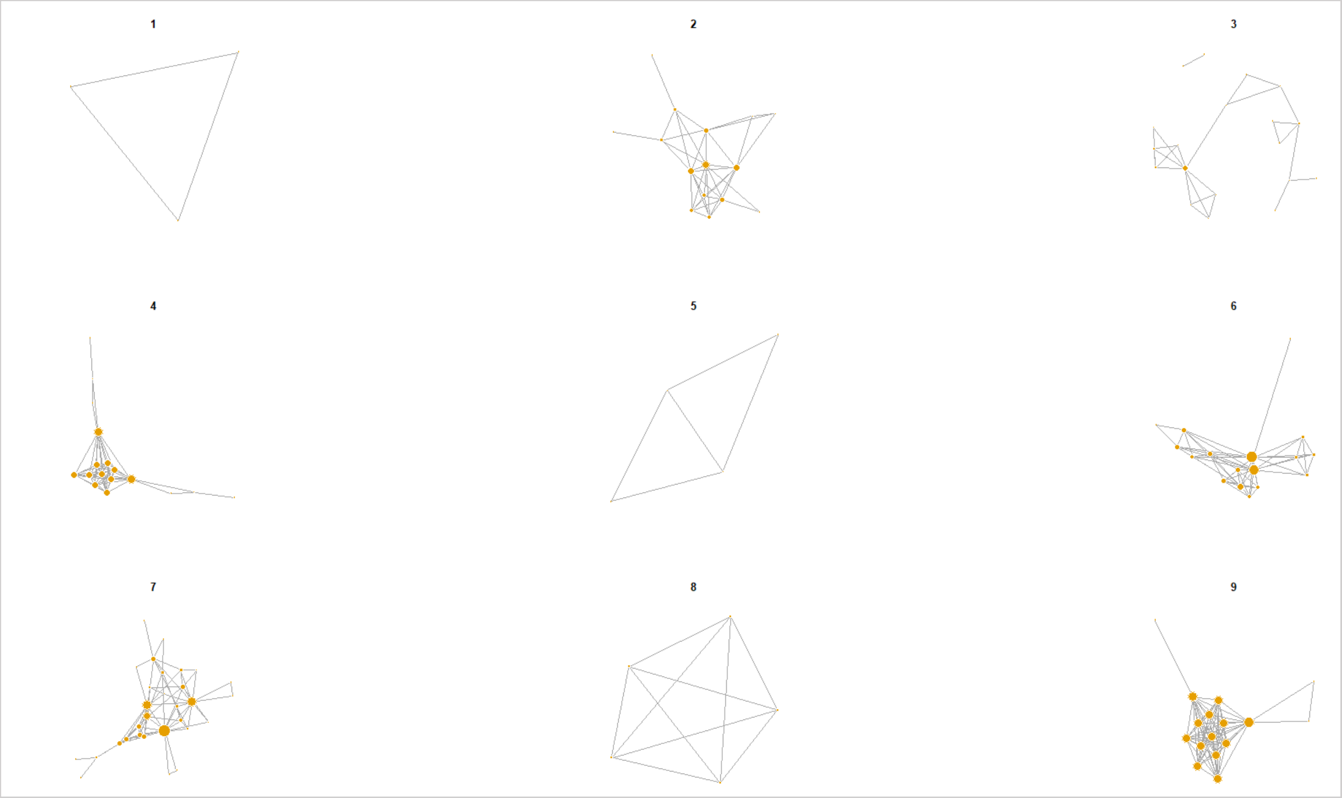
```
# return graphs of each commity
ind_com <- function(i){
    c1<- as.data.frame(sc[[i]])
    c2<- as.data.frame(sc[[i]])
    colnames(c1) <- "d1"
    cb <- inner_join(dis_df,c1)
    colnames(c2) <- "d2"
    cb2 <- inner_join(cb,c2)
    cb2[,1] <- as.character(cb2[,1])
    cb2[,2] <- as.character(cb2[,2])
    cb_f=as.matrix(cb2)
    g=graph_from_edgelist(cb_f[,1:2], directed = FALSE)
    E(g)$weight=as.numeric(cb_f[,3])
    return(g)
}
```

Then we can call individual graphs for i^{th} community by inputting **i** into **inc_com(i).**
We can also draw all the individual graphs for 8 communities respectively:

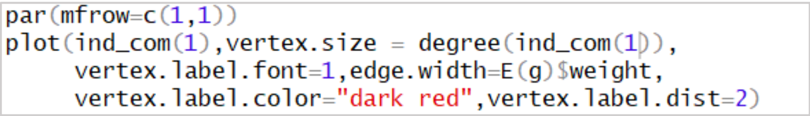


```
# plot 9 communities
par(mfrow=c(3,3))
for(i in 1:9){
  plot(ind_com(i),vertex.size = degree(ind_com(i)),
       vertex.label.color ="dark red",
       vertex.frame.color=FALSE,vertex.label=NA,
      main=i)
}
```

We can firstly briefly see the structure of the 9 communities formed, and then to analyze them individually.

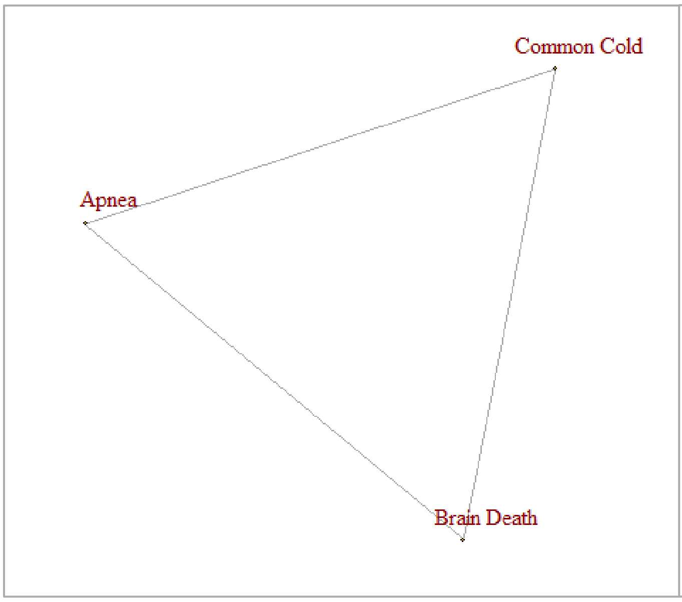


From the graph we can see that some communities are compact and concentrate (community 7, community 9), but some of them are much sparser (community 3). We may use below R code to see every community in detail:

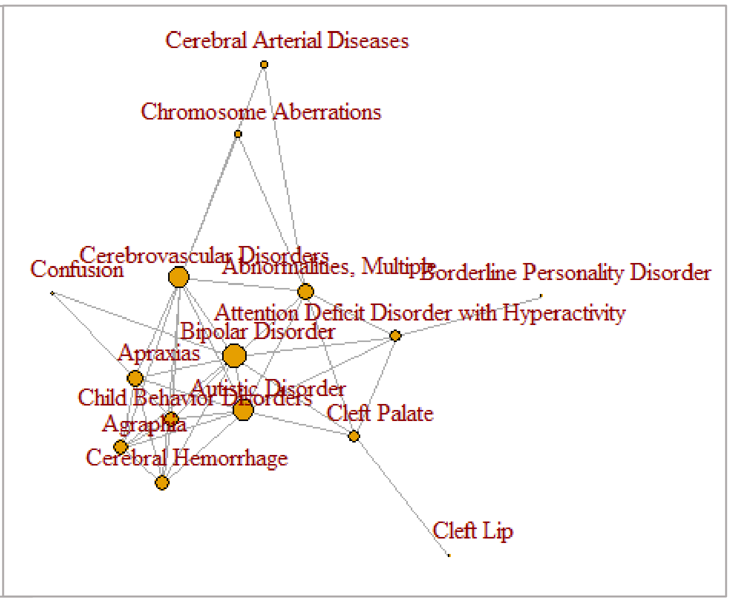


```
par(mfrow=c(1,1))
plot(ind_com(1),vertex.size = degree(ind_com(1)),
     vertex.label.font=1,edge.width=E(g)$weight,
     vertex.label.color="dark red",vertex.label.dist=2)
```

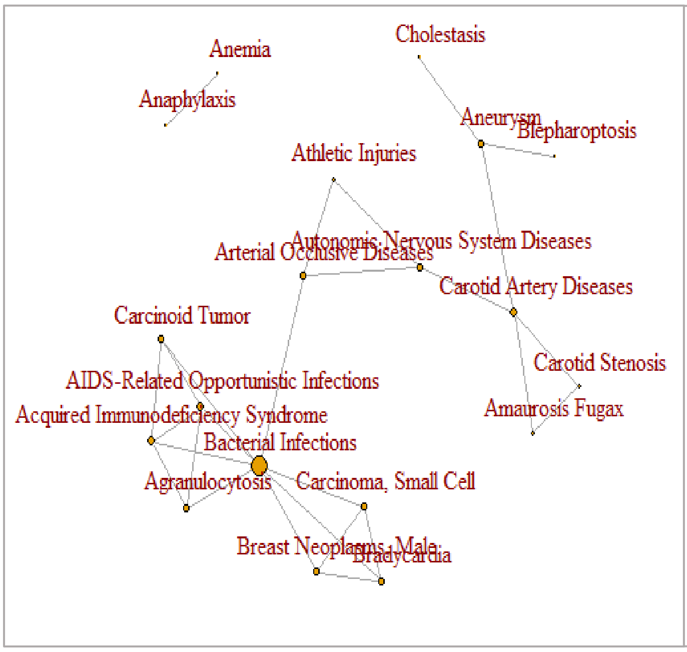
Community 1



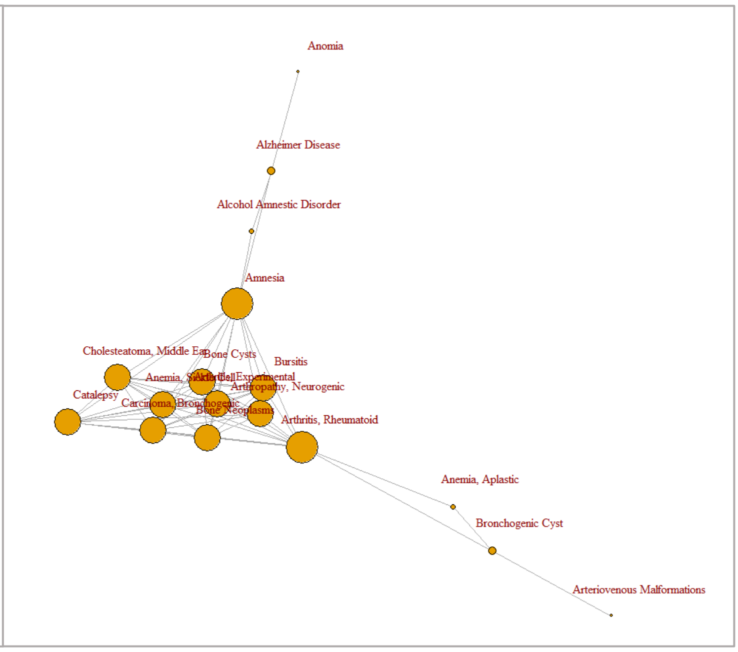
Community 2



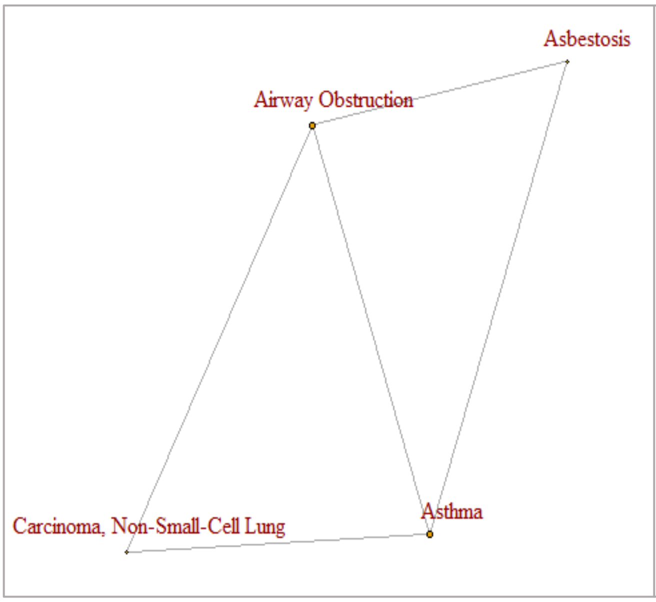
Community 3



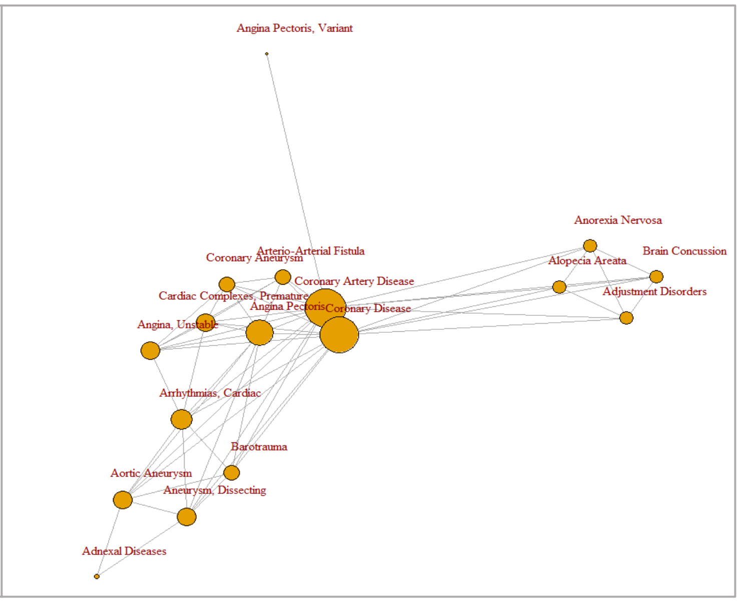
Community 4



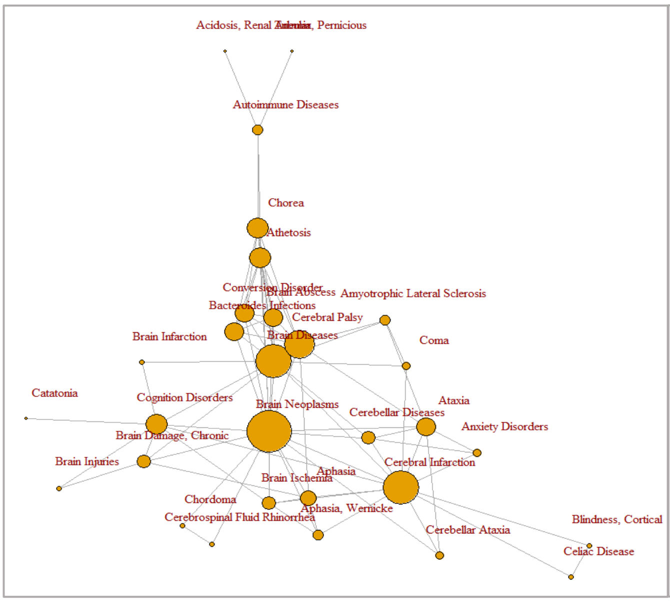
Community 5



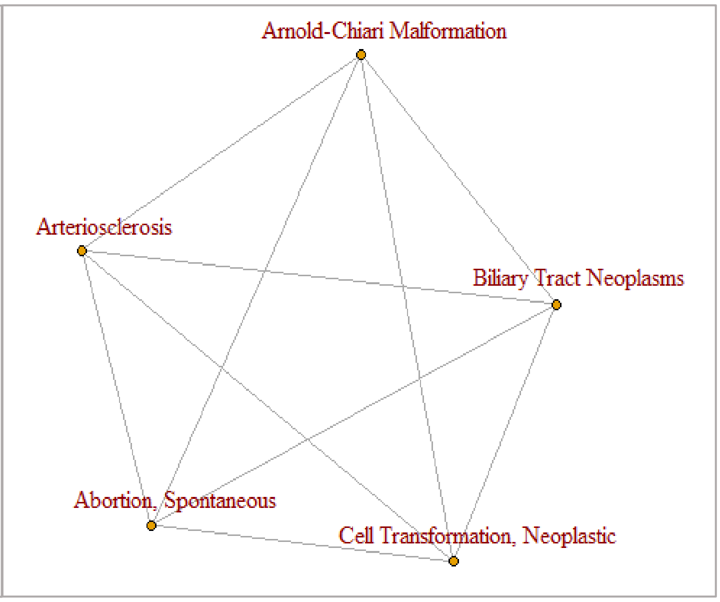
Community 6



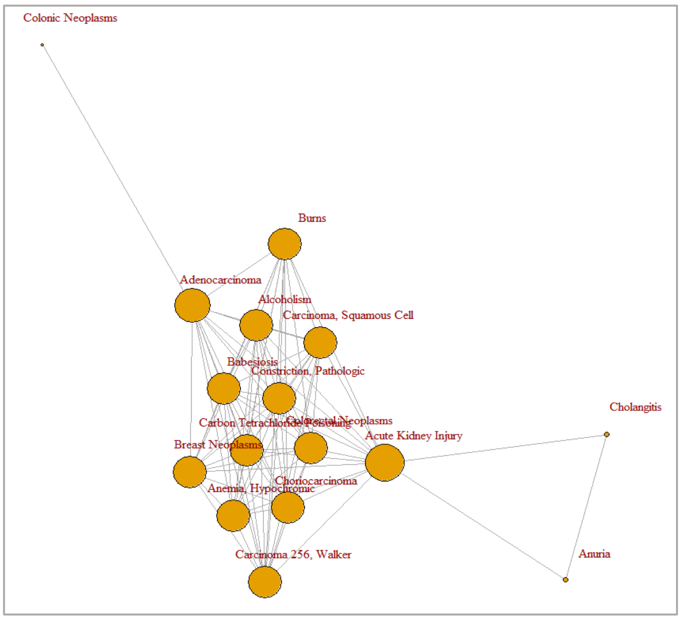
Community 7



Community 8



Community 9



From the above graphs and appendix I (diseased contained in each community) we can have a basic understanding of the 8 communities:

The disease type of community 1 shows **no certain pattern**;
community 2 represents **Mental disorder**;
community 3 represents **Immunological diseases**;
community 4 represents **Hydatoncus or inflammation**;
community 5 represents **Respiratory system related diseases**;
community 6 represents **heart disease**;
community 7 represents **Cerebral diseases**;
The disease type of community 8 shows **no certain pattern**;
community 9 represents **Tumor/cancer/damage to organs**;

5. Key player detection

In this part, we want to find the most effective diseases in three dimensions:

1. **For each community, which diseases' symptoms would be similar to most other diseases'?**
- **for each community**, find the ones who are effective **to the whole graph**
2. **For each community, which diseases' symptoms would be like to most other disease of the similar type?**
- **for each community**, find the ones who are effective **to their own communities**
3. **overall, which diseases' symptoms would be like to most other disease?**
- find the most effective ones in **the whole graph** (regardless of the community)

5.1 find the most effective diseases which would affect the whole graph (for each community)

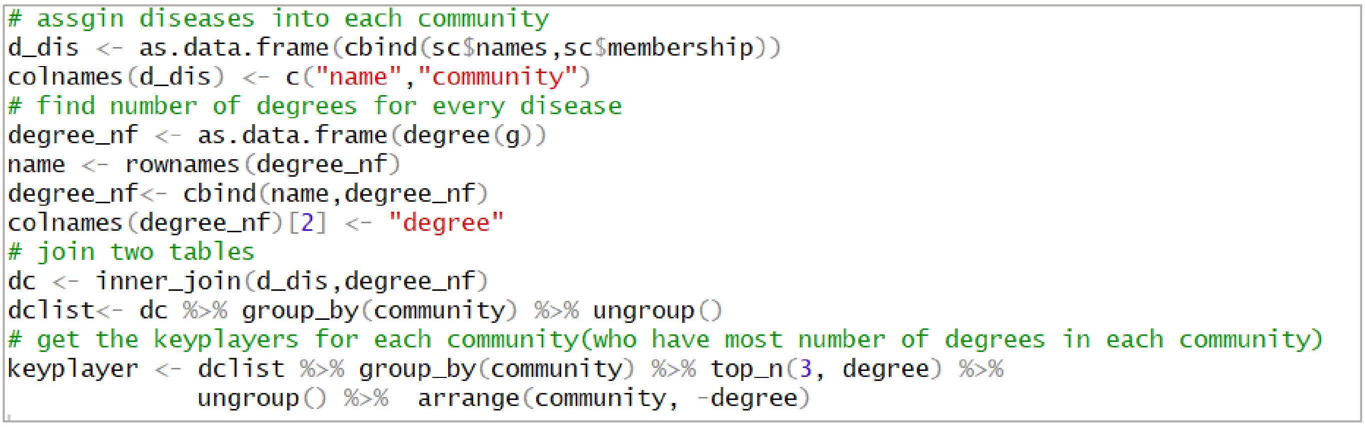
Based on the above analysis, we want to find the key players in graph. In this part we took each community as a unit and the key players would be the diseases which are more effective to the whole graphs than others in the same communities.

Here we used four measurements (degree, betweenness, closeness, weight) to find the key players.

5.1.1 using four measurements to detect top 3 key players in each community (Taking degree as an example)

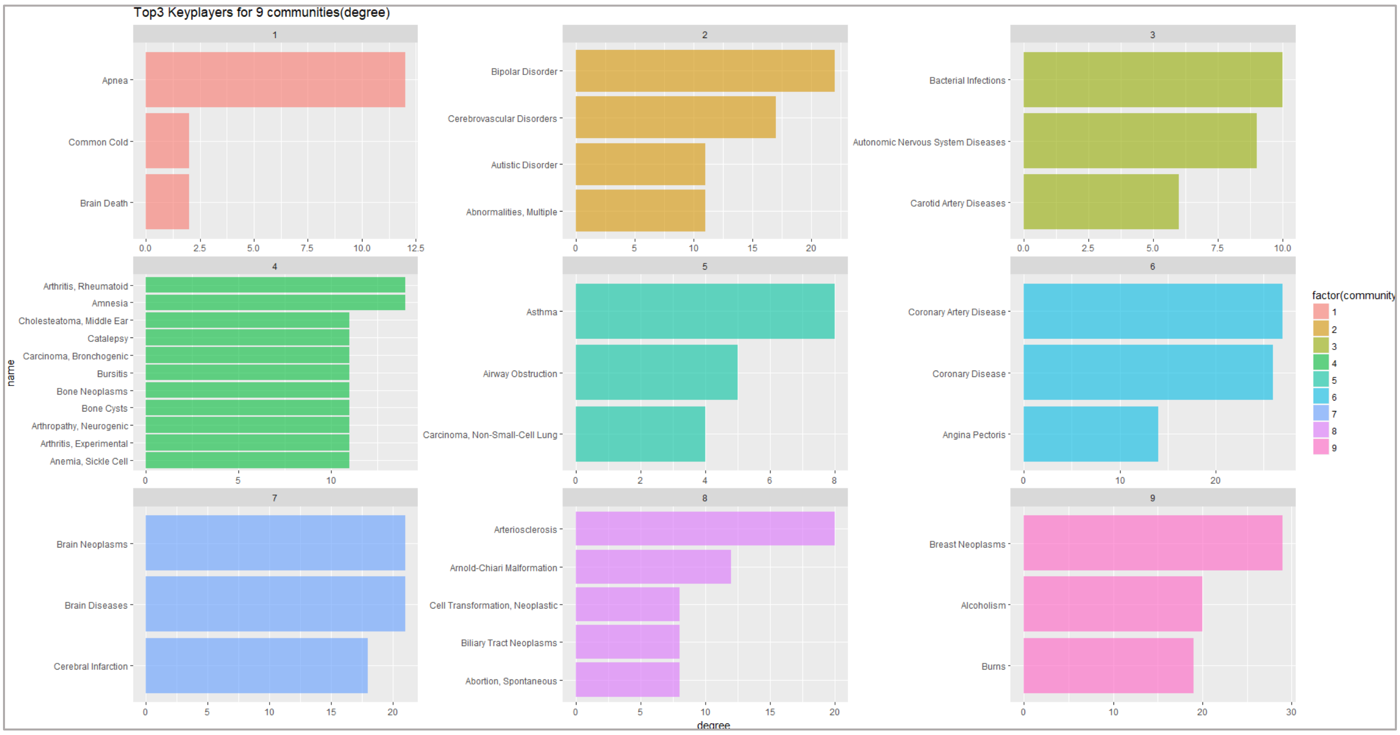
we aimed to find the diseases which have the high degrees in each community.

To do this, we first calculated the degree number for each node in each community, and ranked them from large degree to small degree, taking each community as a unit, then finally chose the diseases which have top 3 highest degree in each community.



```
# assgin diseases into each community
d_dis <- as.data.frame(cbind(sc$names,sc$membership))
colnames(d_dis) <- c("name","community")
# find number of degrees for every disease
degree_nf <- as.data.frame(degree(g))
name <- rownames(degree_nf)
degree_nf<- cbind(name,degree_nf)
colnames(degree_nf)[2] <- "degree"
# join two tables
dc <- inner_join(d_dis,degree_nf)
dclist<- dc %>% group_by(community) %>% ungroup()
# get the keyplayers for each community(who have most number of degrees in each community)
keyplayer <- dclist %>% group_by(community) %>% top_n(3, degree) %>%
             ungroup() %>%  arrange(community, -degree)
```

And then we can plot top 3 key players for each community by degree-based measurement:

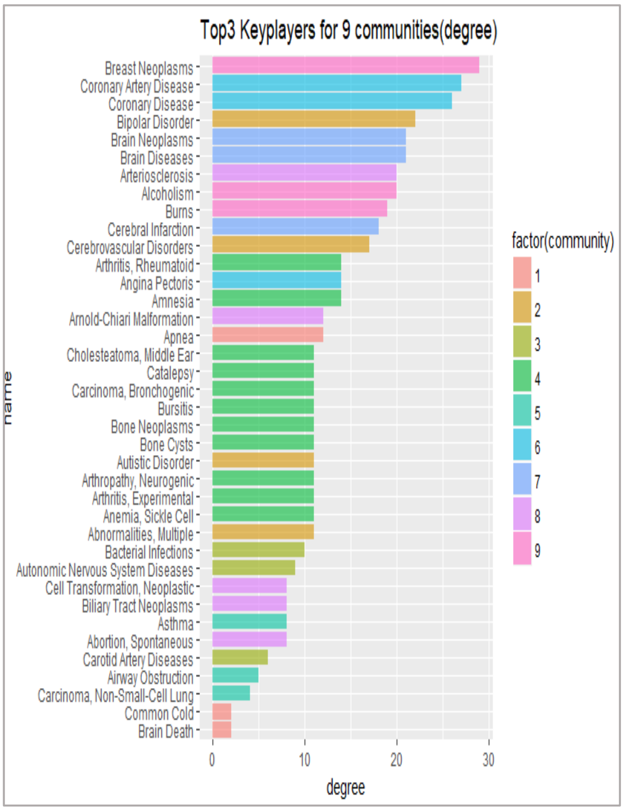


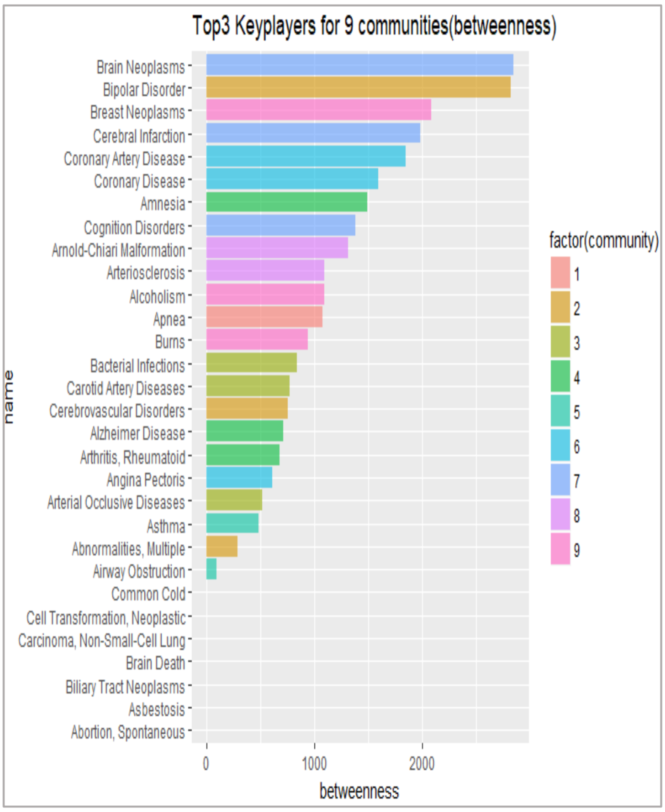
From the graph we can see, for community 2 (Mental disorder), Bipolar Disorder is the most effective one, while for community 3 (Immunological diseases), Bacterial Infections is the most effective one; for other communities we can also find their corresponding key players.

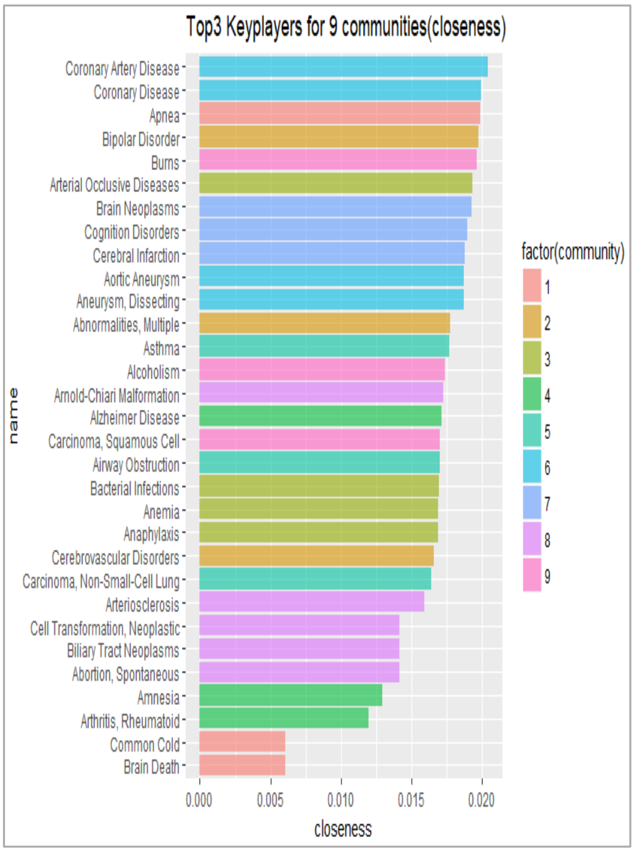
We can also use similar R codes, based on betweenness, closeness, weight, to detect the key players (see appendix II).

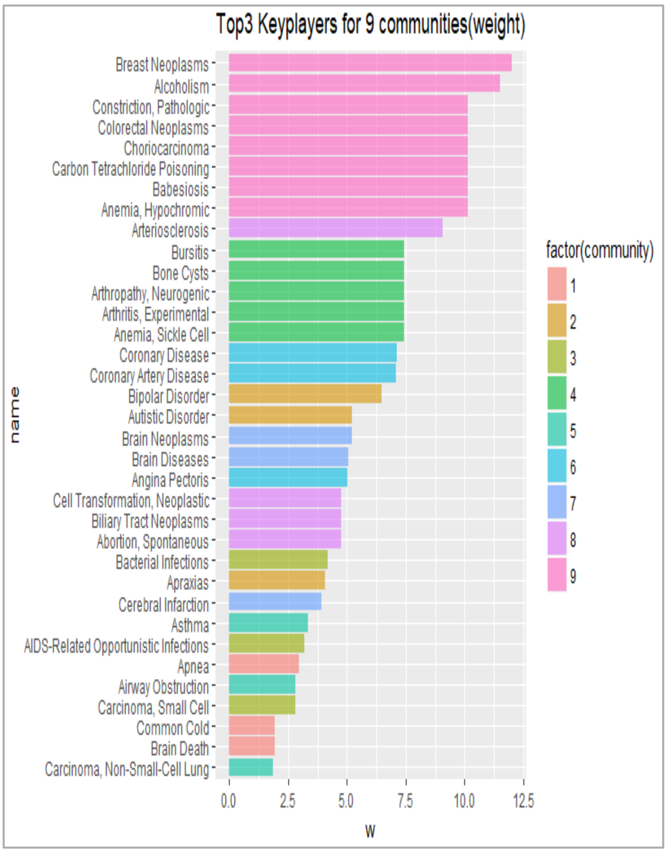
5.1.2 comparison of key players for each measurement

After finished key players for each community, we compared the key players we obtained from them:









In graph above we can find for certain community, the top N key players obtained by four methods, although have somewhat difference but are pretty similar in trend. Like in for community 2 (Mental disorder), Bipolar Disorder is the one of the top effective one. On the other side, we can also find the diseases which are most effective to the graph, like Coronary Artery Disease.

5.1.3 Top 1 key player for each community

We can also use similar code to obtain the top 1 key player for each community:

community	degree_based	betweenness_based	closeness_based	weight_based
1	Apnea	Apnea	Apnea	Apnea
2	Bipolar Disorder	Bipolar Disorder	Bipolar Disorder	Bipolar Disorder
3	Bacterial Infections	Bacterial Infections	Arterial Occlusive Diseases	Bacterial Infections
4	Amnesia; Arthritis, Rheumatoid	Amnesia	Alzheimer Disease	Anemia, Sickle Cell; Arthritis, Experimental;Arthropathy, Neurogenic; Bone Cysts; Bursitis
5	Asthma	Asthma	Asthma	Asthma
6	Coronary Artery Disease	Coronary Artery Disease	Coronary Artery Disease	Coronary Disease
7	Brain Diseases; Brain Neoplasms	Brain Neoplasms	Brain Neoplasms	Brain Neoplasms
8	Arteriosclerosis	Arnold-Chiari Malformation	Arnold-Chiari Malformation	Arteriosclerosis
9	Breast Neoplasms	Breast Neoplasms	Burns	Breast Neoplasms

From the table we can know that:

for community 1, Apnea maybe is the most effective one in community to connect other diseases in the whole graph;

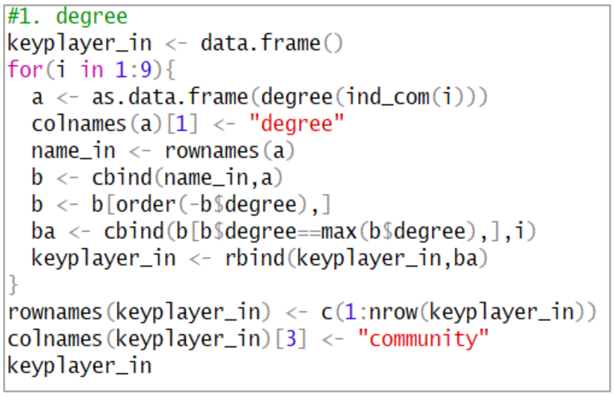
for community 2(Mental disorder), Bipolar Disorder is the most effective one in community to connect other diseases in the whole graph;

for community 3(Immunological diseases), Amnesia is the most effective one in community to connect other diseases in the whole graph;
for other communities, the top 1 key players are also listed on the table.

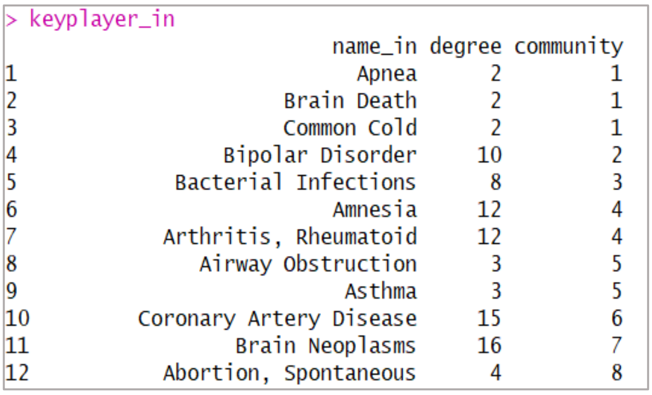
5.2 find nodes which are most effective in each community respectively

Because we have found the diseases which have affect the whole graph, using each community as a unit. Based on that, we also want to see the diseases which the most effective ones are to affect its own communities but not the whole number of the disease in the total graph.

To obtain this, we first split the graph into individual communities, and used the four measurements we have mentioned in 2.1 to apply individually to each community (following R code use degree-based method as an example).



```
#1. degree
keyplayer_in <- data.frame()
for(i in 1:9){
  a <- as.data.frame(degree(ind_com(i)))
  colnames(a)[1] <- "degree"
  name_in <- rownames(a)
  b <- cbind(name_in,a)
  b <- b[order(-b$degree),]
  ba <- cbind(b[b$degree==max(b$degree),],i)
  keyplayer_in <- rbind(keyplayer_in,ba)
}
rownames(keyplayer_in) <- c(1:nrow(keyplayer_in))
colnames(keyplayer_in)[3] <- "community"
keyplayer_in
```

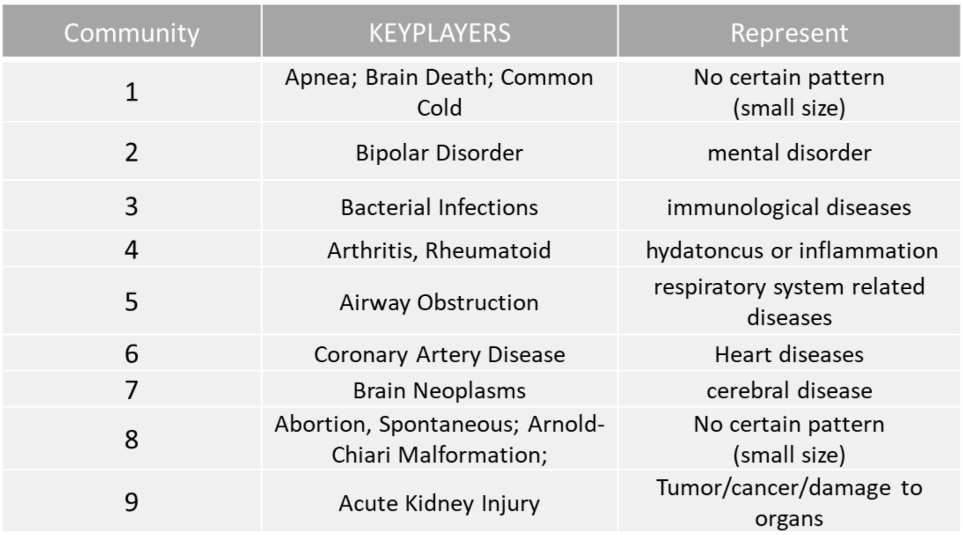


```
> keyplayer_in
                         name_in degree community
1                          Apnea      2         1
2                    Brain Death      2         1
3                    Common Cold      2         1
4               Bipolar Disorder     10         2
5            Bacterial Infections      8         3
6                       Amnesia     12         4
7           Arthritis, Rheumatoid     12         4
8             Airway Obstruction      3         5
9                         Asthma      3         5
10        Coronary Artery Disease     15         6
11                Brain Neoplasms     16         7
12           Abortion, Spontaneous      4         8
```

Use the function **ind_com()** which have been defined previously, we obtained the individual communities graph, then attained the key players in each community based on the four measurements mentioned previously, as listed below:

Communityc	degree_based	betweenness_based	closeness_based	weight_based
1	Apnea; Brain Death; Common Cold	Apnea; Brain Death; Common Cold	Apnea; Brain Death; Common Cold	Apnea; Brain Death; Common Cold
2	Bipolar Disorder	Bipolar Disorder	Bipolar Disorder	Autistic Disorder
3	Bacterial Infections	Bacterial Infections	Arterial Occlusive Diseases	Bacterial Infections
4	Amnesia; Arthritis, Rheumatoid	Amnesia; Arthritis, Rheumatoid	Amnesia	Anemia, Sickle Cell; Arthritis, Experimental; Arthropathy, Neurogenic; Bone Cysts; Bursitis
5	Airway Obstruction; Asthma	Airway Obstruction; Asthma	Asthma	Airway Obstruction
6	Coronary Artery Disease	Coronary Artery Disease	Aneurysm, Dissecting; Aortic Aneurysm	Coronary Artery Disease
7	Brain Neoplasms	Brain Neoplasms	Chorea	Brain Neoplasms
8	Abortion, Spontaneous; Arnold-Chiari Malformation; Arteriosclerosis; Biliary Tract Neoplasms; Cell Transformation, Neoplastic	Abortion, Spontaneous; Arnold-Chiari Malformation; Arteriosclerosis; Biliary Tract Neoplasms; Cell Transformation, Neoplastic	Arnold-Chiari Malformation	Abortion, Spontaneous; Biliary Tract Neoplasms; Cell Transformation, Neoplastic
9	Acute Kidney Injury	Acute Kidney Injury	Acute Kidney Injury	Anemia, Hypochromic; Babesiosis; Carbon Tetrachloride Poisoning; Choriocarcinoma; Colorectal Neoplasms; Constriction, Pathologic

After that we can do some summary and to find the certain key players in each community:



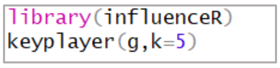
Community	KEYPLAYERS	Represent
1	Apnea; Brain Death; Common Cold	No certain pattern (small size)
2	Bipolar Disorder	mental disorder
3	Bacterial Infections	immunological diseases
4	Arthritis, Rheumatoid	hydatoncus or inflammation
5	Airway Obstruction	respiratory system related diseases
6	Coronary Artery Disease	Heart diseases
7	Brain Neoplasms	cerebral disease
8	Abortion, Spontaneous; Arnold-Chiari Malformation;	No certain pattern (small size)
9	Acute Kidney Injury	Tumor/cancer/damage to organs

From the table we can know:

For community 2 (Mental disorder), Bipolar Disorder is the most effective one in community to have strong connections with other diseases in the same community; that means, when person have the symptoms of Bipolar Disorder, he or she is more likely to get other diseases in community 1;
similarity we can obtain that:
For community 3 (Immunological diseases), Bacterial Infections is the most effective one in community to have strong connections with other diseases in the same community;
For community 4(hydatoncus or inflammation), Arthritis, Rheumatoid is the most effective one in community to have strong connections with other diseases in the same community;
For community 5(respiratory system related diseases), Airway Obstruction is the most effective one in community to have strong connections with other diseases in the same community;
For community 6(heart disease), Coronary Artery Disease is the most effective one in community to have strong connections with other diseases in the same community;
For community 7(cerebral disease), Brain Neoplasms is the most effective one in community to have strong connections with other diseases in the same community;
For community 9 (Tumor/cancer/damage to organs), Acute Kidney Injury is the most effective one in community to have strong connections with other diseases in the same community.
For community 1 and 8, because the sizes of the communities are small, and the diseases contained in these communities are not very correlated, we would not make conclusion on them.

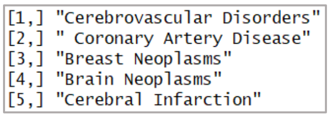
5.3 find nodes which are most effective in the whole graph

Finally, we use package "influenceR" to obtain the overall key players for the whole graph:



```
library(influenceR)
keyplayer(g,k=5)
```

Then we can obtain the overall top 5 key players for the overall graph:



```
[1,] "Cerebrovascular Disorders"
[2,] " Coronary Artery Disease"
[3,] "Breast Neoplasms"
[4,] "Brain Neoplasms"
[5,] "Cerebral Infarction"
```

6. Visualizations and Insights in Gephi

6.1 What is Gephi

Gephi is an interactive graph and network analysis and visualization tool that can help its users to study the properties of graphs and networks in detail, without having to use write any code. What's more, it also supports almost all types of graphical networks, including complex networks, hierarchical networks, and temporal networks. Gephi has a lot of ready-to-use features that allow users to create stunning and informative visualizations.

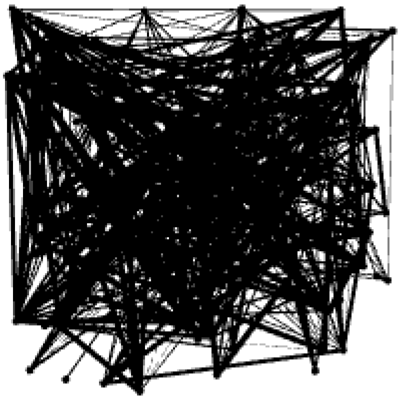
6.2 Limitations of Gephi

During using Gephi to visualize our human disease-symptom network, we found that the main shortcoming of this software is that we cannot modify parameters, which may influence the final result.

6.3 How we do visualizations of disease-symptom network in Gephi

6.3.1 Import the spreadsheet

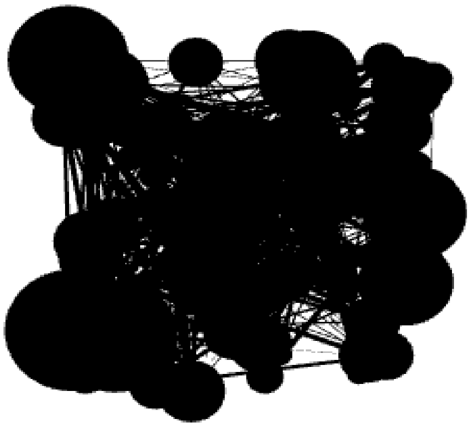
The steps to generate the original graph of disease-symptom network are as follows:
1) Click the **File** in the menu bar.
2) Choose **Import the Spreadsheet** in the drop-down menu.
3) Choose **Side Table** for import data, and **GB2312** for character set.
4) There is an enter report window and select **Undirected** type of the graph.
Then a random graph has been created.



6.3.2 Coloring and sizing nodes and edges in the graph

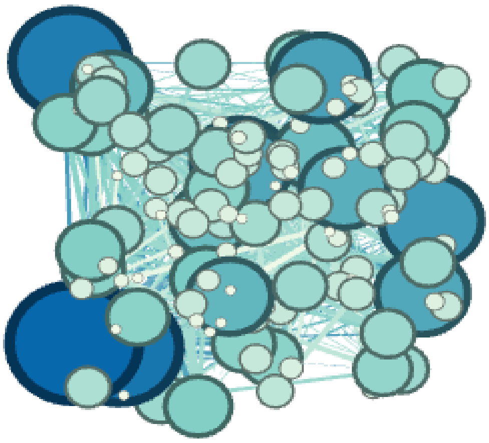
This original graph actually cannot satisfy our expectation. As a result, we manually alter some of the attributes of the nodes, such as the color and size. This section explains how to achieve this.

The following steps illustrate how to resize individual nodes in the graph:
1) For vertices, click on the snail **Shell-Shaped Button** in the toolbar placed on the left-hand side of the screen.
2) In the **Ranking** window, there is only one option left – **Degree** due to the undirected graph.
3) Input the number **15** for minimum size and **180** for maximum size.
4) Click the '**Application**' button.

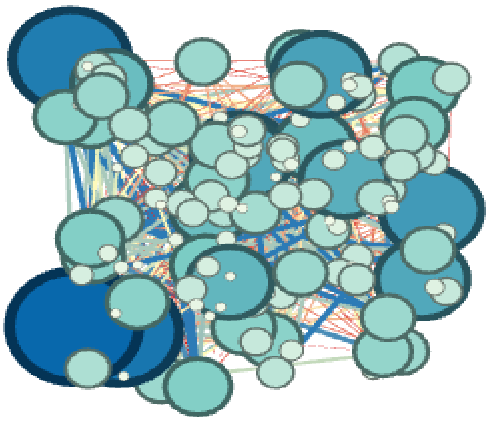


We also want to recolor individual nodes in the graph, and follow these steps:

1) For vertices, click on the **Palette-Shaped Button** in the toolbar placed on the left-hand side of the screen.
2) In the **Ranking** window, there is only one option left – **Degree**.
3) There is a **Color Box** and select the desired color that we would like to assign to nodes.
4) Click the '**Application**' button.



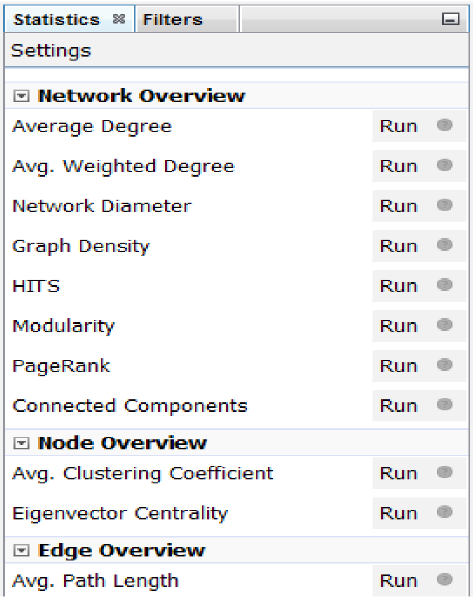
Steps of coloring edges are the same as nodes. The graph is below:



6.3.3 Running statistical metrics (take average degree and modularity as examples)

Gephi provides some ready-to-use ways to study the statistical properties of graphical networks. These statistical properties include the properties of the network as a whole, as well as individual properties of nodes and edges within the network.

To view different metrics available in Gephi for exploring a graph, we follow these steps:
1) In the Statistics panel situated on the right-hand side of the Gephi window, find the tab that reads Settings.
2) From the list of available metrics in the pop-up window, check the ones you would like to use:



Settings

Network Overview

Average Degree	Run
Avg. Weighted Degree	Run
Network Diameter	Run
Graph Density	Run
HITS	Run
Modularity	Run
PageRank	Run
Connected Components	Run

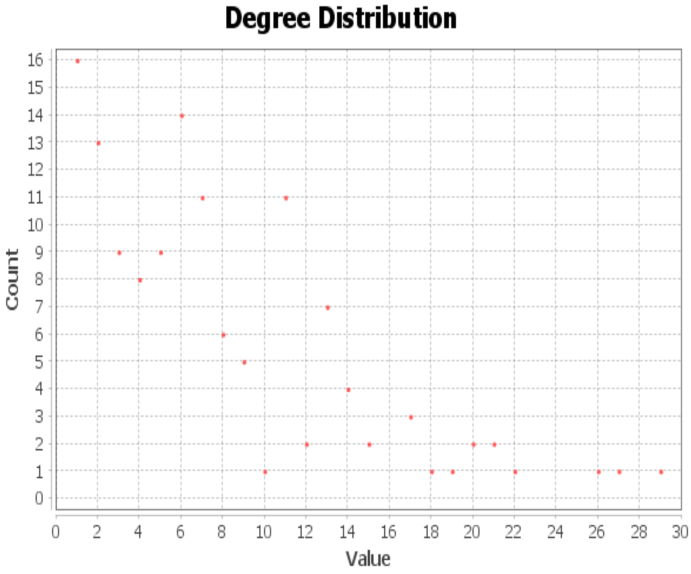
Node Overview

Avg. Clustering Coefficient	Run
Eigenvector Centrality	Run

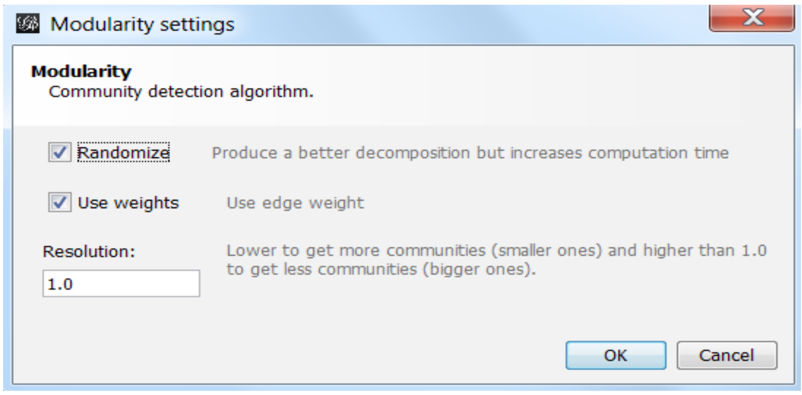
Edge Overview

Avg. Path Length	Run

3) Click on the **Run** button located beside **Average Degree**. This opens up a window containing the degree report for the disease-symptom network, as shown in the following screenshot.



4) Hit the **Run** button adjacent to **Modularity**.
5) In the Modularity settings window, enter a **Resolution** in the textbox depending on whether we want a small or large number of communities (we choose 1.2):


Modularity settings

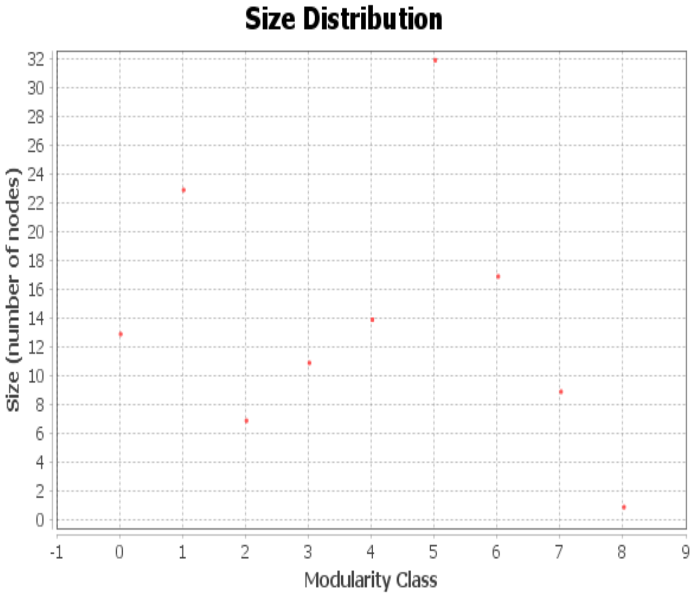
Modularity
Community detection algorithm.

☑ Randomize — Produce a better decomposition but increases computation time

☑ Use weights — Use edge weight

Resolution: 1.0 — Lower to get more communities (smaller ones) and higher than 1.0 to get less communities (bigger ones).

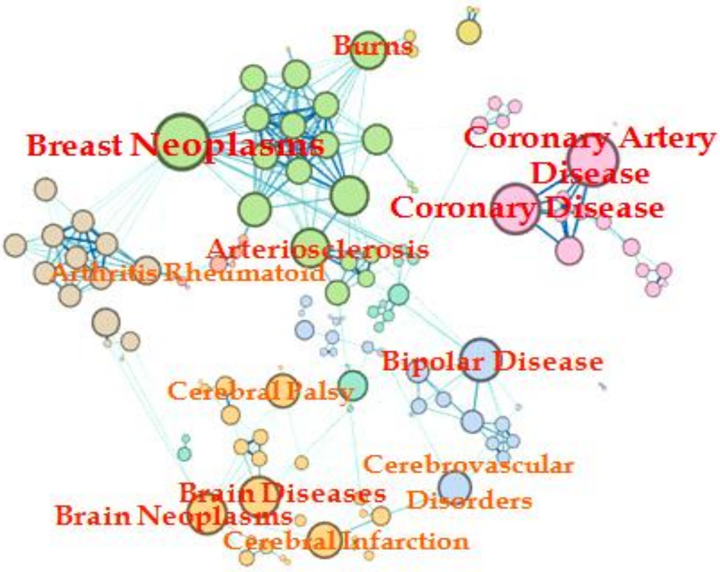
We also choose to randomize to get a better decomposition of the graph into communities, but it will increase the computation time.



The result is 0.705, which is an excellent result, which points to strong relationships within the same commodities but weaker relationship across different communities.

6.3.4 Using graph layout algorithms (take Force Atlas 2 and Fruchterman Reingold as examples)
While working with graph and network visualizations, one of the most important requirements is to have the ability to visualize the graph with its nodes placed according to some structure across the graphical space. A good tool provides the capability to the users to restructure the network in order to visualize it in a way in which the required parts of the graph are enhanced, similar nodes occupy the same subspace in the graphical space, and all the nodes are clearly distinguishable from each other.

Force Atlas 2 is an algorithm in the set of force-directed algorithms available in Gephi. It attempts to make a balance between the quality of the final layout and the speed of the computation algorithm.
1) As we have calculated the modularity of this network, first we come back to the **Partition** button and choose **Modularity class**. We can also change colors for different communities.
2) In the **Layout** panel, click on the drop-down menu that says – **Choose a layout**.
3) From the drop-down menu, select **Force Atlas 2**.
4) Check the **Prohibit Hubs**, **Linlog mode**, and **Avoid Overlap**.



The Fruchterman Reingold layout algorithm belongs to the class of force-directed algorithms. It is one of the standard algorithms in Gephi and is made use of quite often.

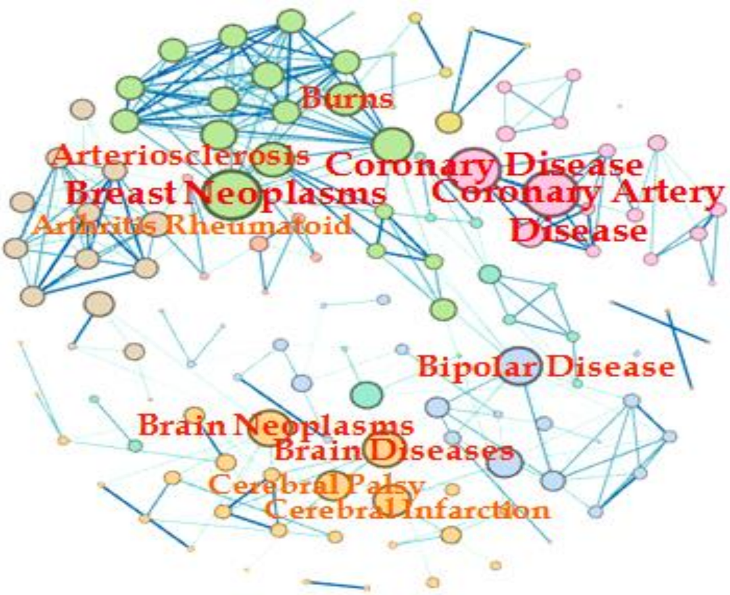
In the Fruchterman Reingold layout algorithm, the nodes are assumed to be entities made of steel and the edges are assumed to be springs. The attractive force between the nodes mimics the spring force, whereas the repulsive force between the nodes is analogous to the electrical force.

This algorithm does not take into consideration the edge weight to come up with an optimal layout.

1) In the **Layout** panel, click on the drop-down menu that says – **Choose a layout**.

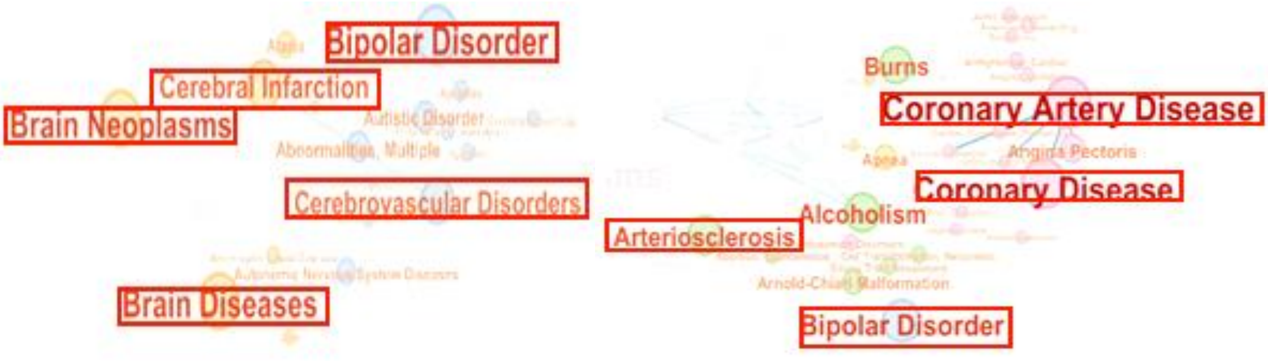
2) From the drop-down menu, select **Fruchterman Reingold**.

3) Change the number of zone to increase the size of the graph.



6.4 Some different insights behind disease-symptom network

In the first graph, some diseases are chosen by red boxes, such as Brain Neoplasms, Cerebral Infarction, which are correlated to brain and occur commonly in adults. In other word, some people who suffer from one kind of brain disease will also gain other brain diseases. One reason is that shared symptoms indicate shared genes between diseases. Diseases with more similar symptoms are more likely to have common gene associations. The second connection graph surrounds Arterial Blood Vessels. Apart from this, some diseases will cause other disease. For example, people who prolonged to suffer from Bipolar Disorder may have a higher probability of Coronary Disease and Arteriosclerosis along with the growth of the age.



Appendix I (Diseased Contained in Each Community):

Community 1:
"Apnea", "Brain Death", "Common Cold"

Community 2:
"Abnormalities, Multiple", "Attention Deficit Disorder with Hyperactivity", "Autistic Disorder", "Bipolar Disorder", "Cerebral Arterial Diseases", "Cerebrovascular Disorders", "Chromosome Aberrations", "Cleft Palate", "Agraphia", "Apraxias", "Cerebral Hemorrhage", "Child Behavior Disorders", "Confusion", "Articulation Disorders", "Borderline Personality Disorder", "Cleft Lip"

Community 3:
"Acquired Immunodeficiency Syndrome", "Agranulocytosis", "AIDS-Related Opportunistic Infections", "Bacterial Infections", "Carcinoid Tumor", "Carcinoma, Small Cell", "Amaurosis Fugax", "Carotid Artery Diseases", "Carotid Stenosis", "Autonomic Nervous System Diseases", "Anaphylaxis" "Anemia", "Aneurysm", "Blepharoptosis", "Cholestasis", "Arterial Occlusive Diseases", "Athletic Injuries", "Bradycardia", "Breast Neoplasms, Male"

Community 4:
"Alcohol Amnestic Disorder", "Alzheimer Disease", "Amnesia", "Anomia", "Anemia, Sickle Cell", "Arthritis, Experimental", "Arthritis, Rheumatoid", "Arthropathy, Neurogenic", "Bone Cysts", "Bone Neoplasms", "Bursitis", "Carcinoma, Bronchogenic", "Catalepsy", "Cholesteatoma, Middle Ear", "Anemia, Aplastic", "Bronchogenic Cyst", "Arteriovenous Malformations"

Community 5:
"Airway Obstruction", "Asbestosis", "Asthma", "Carcinoma, Non-Small-Cell Lung"

Community 6:
"Coronary Artery Disease", "Coronary Disease", "Adjustment Disorders", "Alopecia Areata","Anorexia Nervosa", "Brain Concussion", "Adnexal Diseases", "Aneurysm, Dissecting", "Aortic Aneurysm", "Angina Pectoris", "Arrhythmias, Cardiac", "Barotrauma", "Angina, Unstable", "Arterio-Arterial Fistula", "Cardiac Complexes, Premature", "Coronary Aneurysm", "Angina Pectoris, Variant"

Community 7:
"Athetosis", "Brain Diseases", "Cerebral Palsy", "Chorea", "Acidosis, Renal Tubular", "Autoimmune Diseases", "Cognition Disorders", "Brain Neoplasms", "Aphasia", "Brain Damage, Chronic", "Brain Injuries", "Cerebral Infarction", "Amyotrophic Lateral Sclerosis", "Ataxia", "Coma", "Anemia, Pernicious", "Anxiety Disorders", "Cerebellar Diseases", "Aphasia, Wernicke", "Brain Ischemia", "Cerebellar Ataxia", "Bacteroides Infections", "Brain Abscess", "Conversion Disorder", "Blindness, Cortical", "Celiac Disease", "Brain Infarction", "Cerebrospinal Fluid Rhinorrhea", "Chordoma", "Catatonia"

Community 8:
"Abortion, Spontaneous", "Arnold-Chiari Malformation", "Arteriosclerosis", "Biliary Tract Neoplasms", "Cell Transformation, Neoplastic"

Community 9:
"Alcoholism", "Acute Kidney Injury", "Adenocarcinoma", "Anemia, Hypochromic", "Anuria", "Babesiosis", "Breast Neoplasms", "Burns", "Carbon TetrachloridePoisoning", "Carcinoma 256, Walker", "Carcinoma, Squamous Cell", "Cholangitis" "Choriocarcinoma", "Colorectal Neoplasms", "Constriction, Pathologic",

"Colonic Neoplasms"

Translate version:

Community 1:
“呼吸暂停”，“脑死亡”，“感冒”

Community 2:
“异常多发”“多动症注意力缺陷症”“自闭症”“双相情感障碍”“脑动脉疾病”“脑血管障碍”“染色体畸变”“腭裂”“失写症” (精神性)失用(症) “，”脑溢血“，”儿童行为障碍“，”混乱“，”发音障碍“，”边缘性人格障碍“，”唇裂“

Community 3:
“获得性免疫缺陷综合征”，“粒细胞缺乏症”，“与艾滋病相关的机会感染”，“细菌感染”，“类癌瘤”，“癌细胞”，“一时性黑蒙”“颈动脉疾病”“颈动脉狭窄” ，“自主神经系统疾病”，“过敏症”，“贫血”，“动脉瘤”，“眼睑下垂”，“胆汁淤积”，“动脉闭塞性疾病”，“运动损伤”，“心动过缓”

Community 4:
“酒精记忆障碍”，“阿尔茨海默病”，“失忆症”，“失语症”，“贫血症，镰状细胞”，“关节炎，实验性”，“关节炎，类风湿性关节炎，神经原性”骨肿瘤“，”粘液囊炎“，”癌症，支气管炎“，”僵住“，”胆脂瘤，中耳“，”贫血，再生障碍性疾病“，”支气管囊肿“”动静脉畸形“

Community 5:
“气道阻塞”，“石棉肺”，“哮喘”，“癌，非小细胞肺”

Community 6:
“冠状动脉疾病”，“冠状动脉疾病”“调节障碍”“脱发症”“神经性厌食症”“脑震荡”“附件疾病”“动脉瘤解剖”“主动脉瘤”“心绞痛“心律失常”“气压伤”“心绞痛不稳定”“动脉动脉瘘”“心脏综合征早产”“冠状动脉瘤”“心绞痛，变异”

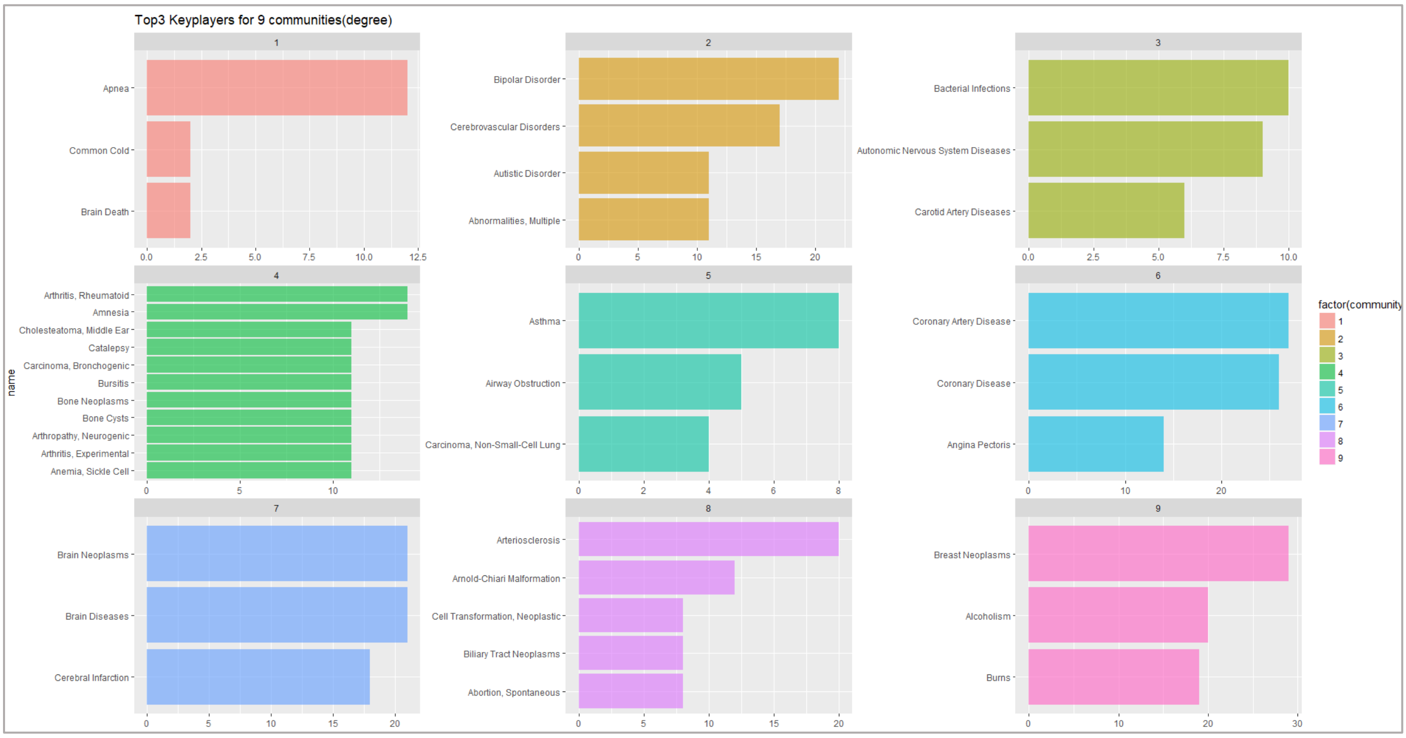
Community 7:
“麻痹病”，“脑疾病”，“脑性麻痹”，“舞蹈病”，“酸中毒，肾管”，“自身免疫性疾病”，“认知障碍”，“脑肿瘤”，“失语症” ，“脑损伤”，“脑梗塞”，“肌萎缩性侧索硬化症”，“共济失调”，“昏迷”，“贫血，有害”，“焦虑症”，“小脑疾病”“ 感觉性失语症” “小脑性共济失调”“拟杆菌感染”“脑脓肿”“转化障碍”“盲目皮质”“乳糜泻”“脑梗塞”“脑脊液鼻漏”“脊索瘤”“紧张症 “

Community 8:
“堕胎，自发”，“阿-基氏脑畸形”，“动脉硬化”，“胆道肿瘤”，“细胞转化，肿瘤”

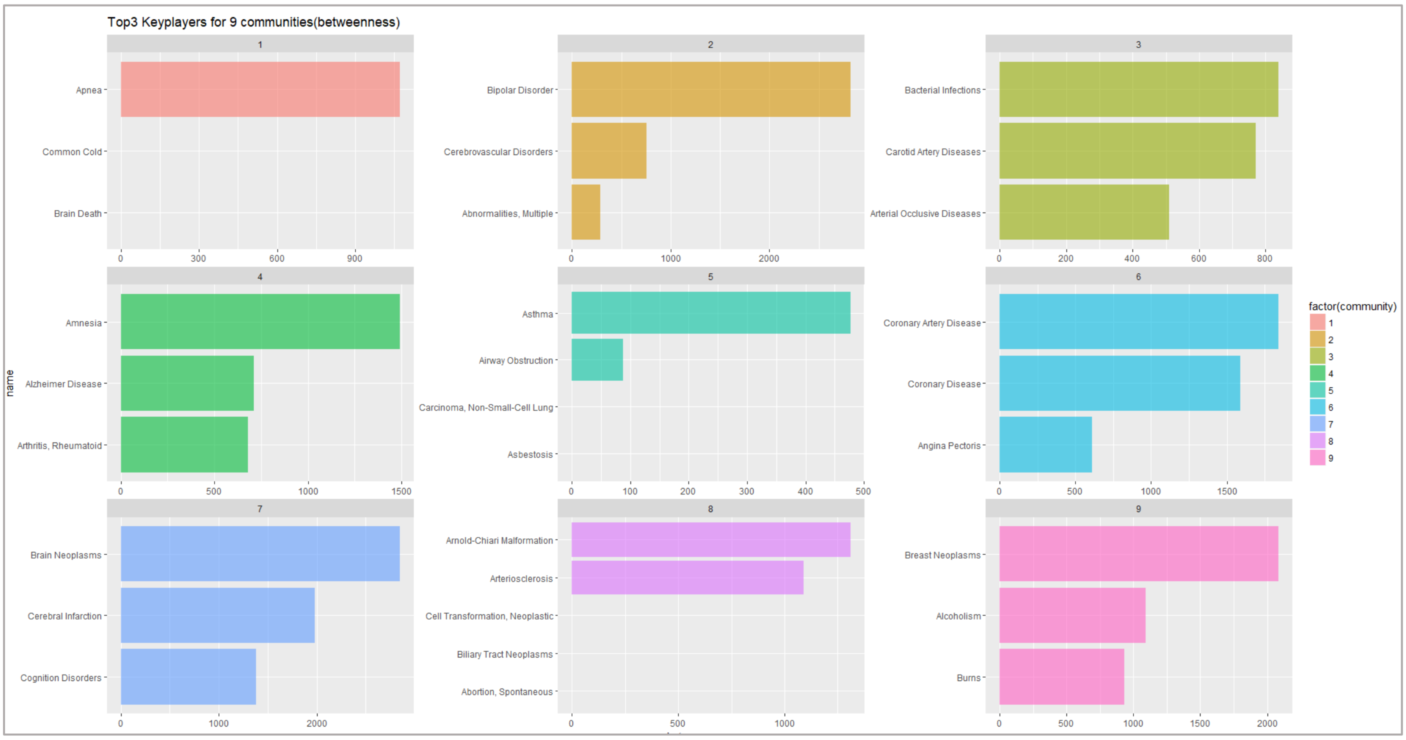
Community 9:
“酒精中毒”“急性肾损伤”“腺癌”“贫血，色素减退”“无尿”“巴贝斯虫病”“乳腺肿瘤”“烧伤”“四氯化碳中毒”“癌症 256，沃克”癌，鳞状细胞“，”胆管炎“，”绒毛膜癌“，”结肠直肠肿瘤“，”缩窄，病理“，”结肠肿瘤“

Appendix II (Key Players Detection using Four Measurements):

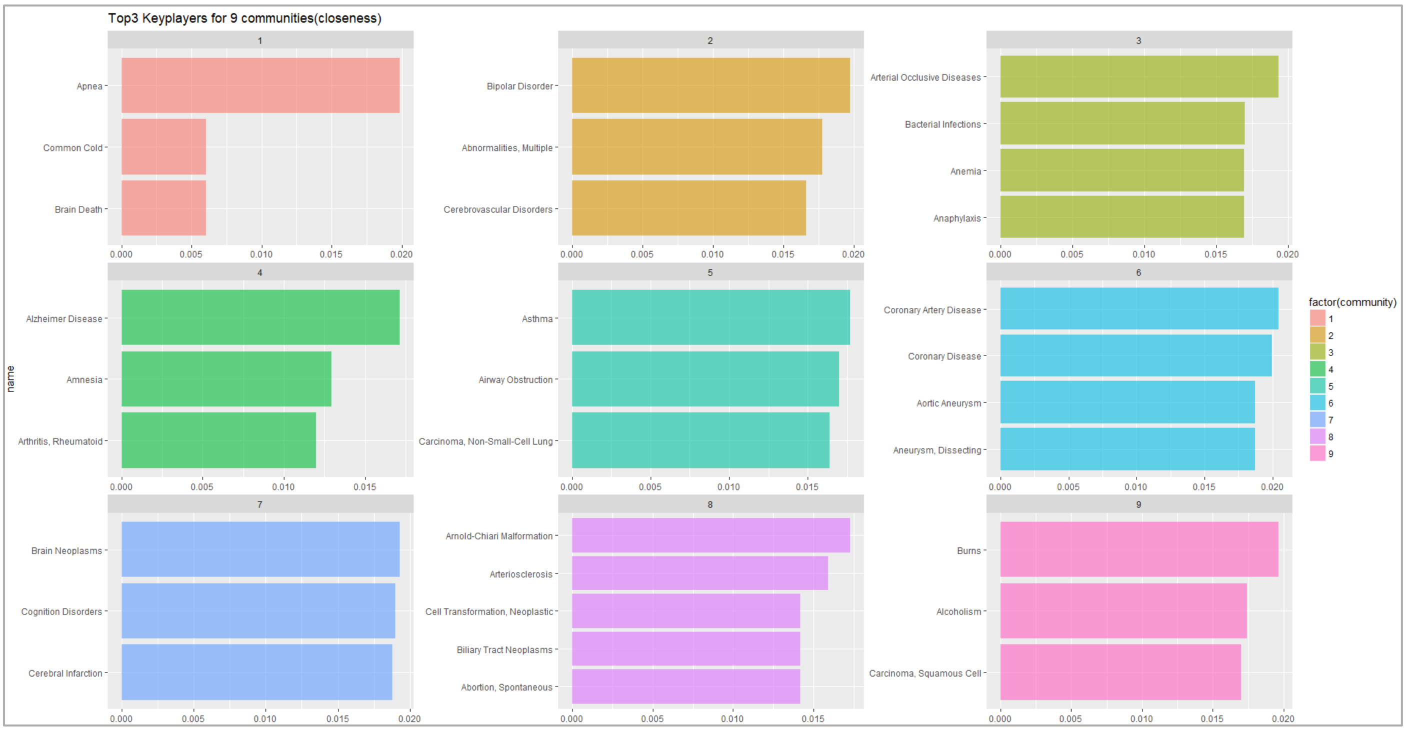
1. degree



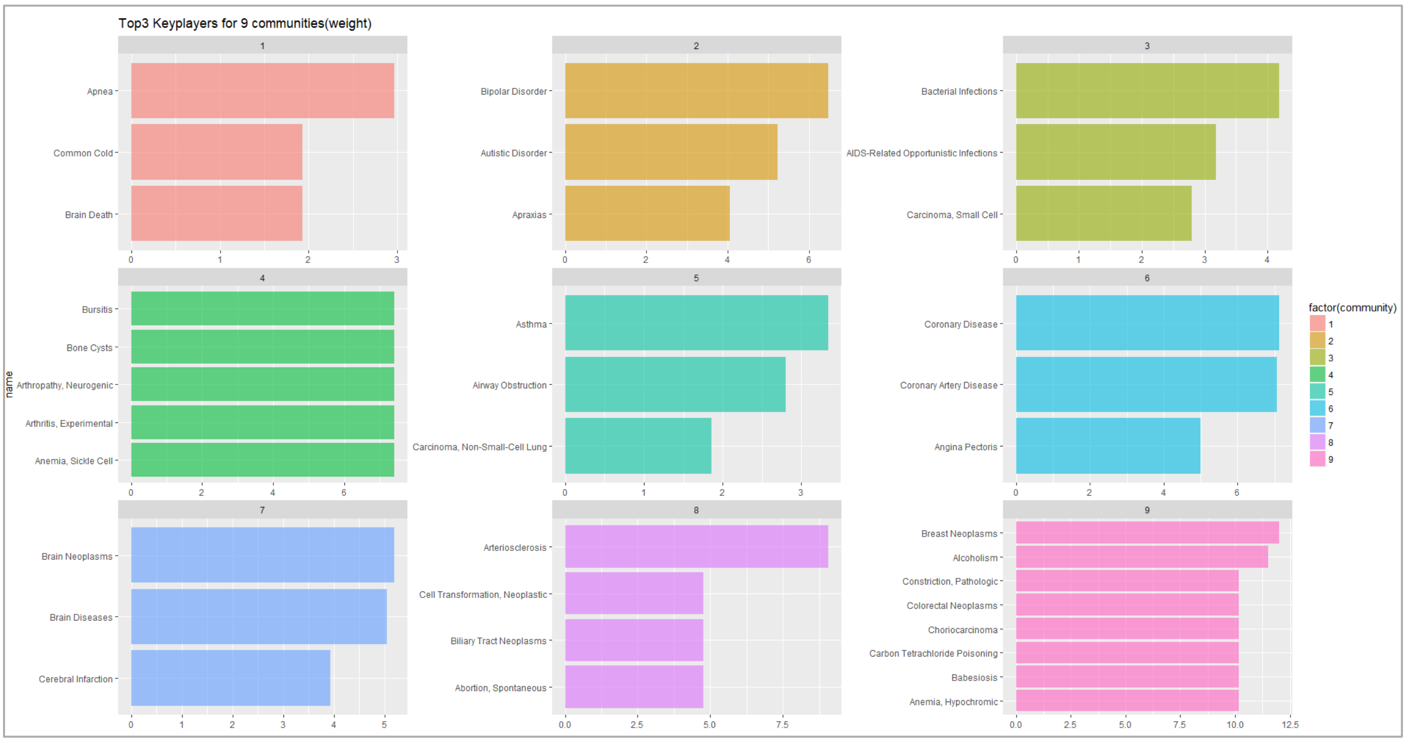
2. betweenness



3. closeness



4. weight


Top3 Keyplayers for 9 communities(weight)
Apnea
Common Cold
Brain Death
Bipolar Disorder
Autistic Disorder
Apraxias
Bacterial Infections
AIDS-Related Opportunistic Infections
Carcinoma, Small Cell
Bursitis
Bone Cysts
Arthropathy, Neurogenic
Arthritis, Experimental
Anemia, Sickle Cell
Asthma
Airway Obstruction
Carcinoma, Non-Small-Cell Lung
Coronary Disease
Coronary Artery Disease
Angina Pectoris
Brain Neoplasms
Brain Diseases
Cerebral Infarction
Arteriosclerosis
Cell Transformation, Neoplastic
Biliary Tract Neoplasms
Abortion, Spontaneous
Breast Neoplasms
Alcoholism
Constriction, Pathologic
Colorectal Neoplasms
Choriocarcinoma
Carbon Tetrachloride Poisoning
Babesiosis
Anemia, Hypochromic
name
factor(community)