CSN Lab Session 5

Finding and assessing community structure

Daniel Benedí & Jan Herlyn

Complex Social Networks
UPC
January, 2024

Contents

1 Introduction

Clustering is the process of organizing a set of objects into groups in which the objects within each group are more similar to each other than to those in other groups. In the study of complex networks, it is claimed that a network has a community structure if the nodes of the network can be easily divided into groups with a high level of internal connections. See an example of a network with community structure in Figure 1.

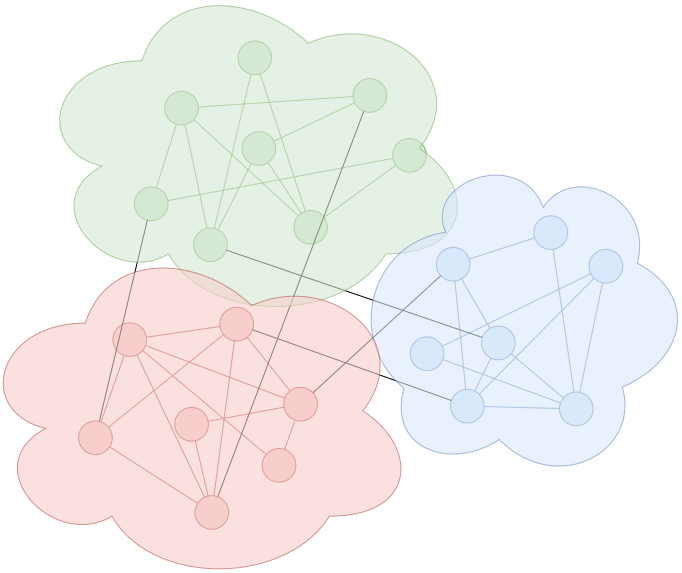

Figure 1: A schematic representation of a network with community structure. In this network there are three communities of densely connected vertices with a much lower density of connections between them. Extracted and adapted from [2].

Discovering a community structure in a real network is a frequent occurrence. It is essential to identify any existing underlying community structure in a network for a variety of reasons. For example, in a protein interaction network, communities are associated with proteins that have similar functions within a biological cell [12]. Similarly, the presence of communities can have an impact on a wide range of behaviors, including the propagation of rumors and the transmission of epidemics in a network [7, 8].

We have observed that the identification of communities is of great importance in various fields. The report will examine different community detection algorithms in terms of a variety of metrics. We will assess the performance of our methods by utilizing a set of 4 networks, ranging from real-world networks to artificial ones, some of which have known ground-truth communities and some with unknown communities.

This document is divided into four different sections. Next, our results are presented in Section 2. Discussion and conclusions are covered in section 3, and finally, some aspects of the methodology are presented in section 4.

2 Results

Tables 7, 8, 9 and 10 show the significance metrics computed for the different networks using `evaluate_significance` provided by the `clusterAnalytics` R package. Tables 1 and 2 highlight which clustering algorithm performs best according to the different metrics. As can be seen, the best algorithm is highly dependent on the network itself.

Surprisingly, the known canonical clustering does not seem to outperform the best clustering algorithm in terms of the metrics considered for Barabási-Albert blocks.

Table 3 shows the global Jaccard indices for each of the networks and algorithms. In case a canonical clustering was known, it was used as the baseline. For the Enron network, we compared the algorithms to clustering

produced by the Label Propagation algorithm. In the case of DBLP we used Louvain as a benchmark. Note that the clustering algorithms use randomness in their decisions, so two computations of the same algorithm produced two different clusterings, which in turn lead to a Jaccard index lower than 1 for Louvain.

Metric	**Karate**		**Barabasi Albert Blocks**	
	best	second best	best	second best
↑ Internal Density	Edge Betweenness	Walktrap	Walktrap	Spin Glass
↑ Edges Inside	ground truth	Label Propagation + FG	Label Propagation	Edge Betweenness
↑ Average Degree	ground truth	Label Propagation + FG	Label Propagation	Louvain + SG
↓ Expansion	ground truth	Label Propagation + FG	Label Propagation	Louvain + SG
↓ Cut Ratio	ground truth	Label Propagation + FG	Louvain + SG + LP	
↓ Conductance	ground truth	Label Propagation + FG	Label Propagation	Lovain
↓ Normalized Cut	ground truth	Label Propagation + FG	Label Propagation	Lovain + SG
↓ Maximum ODF	ground truth	Label Propagation + FG	Label Propagation	Spin Glass
↓ Average ODF	ground truth	Label Propagation + FG	Label Propagation	Spin-Glass

Table 1: Best values for communities with ground truth

Metric	**Enron**	DBLP
↑ Internal Density	Edge Betweenness	Label Propagation
↑ Edges Inside	Label Propagation	Spin-Glass
↑ Average Degree	Label Propagation	Louvain
↓ Expansion	Label Propagation	Louvain
↓ Cut Ratio	Label Propagation	Louvain
↓ Conductance	Label Propagation	Louvain
↓ Normalized Cut	Label Propagation	Louvain
↓ Maximum ODF	Label Propagation	Louvain
↓ Average ODF	Label Propagation	Louvain

Table 2: Best values for communities without ground truth

	Louvain	Label Propagation	Walktrap	Edge Betweenness	Fast Greedy	Spin-Glass
karate	0.7977941	0.7977941	0.5853758	0.4040033	0.7977941	0.6147876
Barabasi-Albert	0.8470966	0.2700000	0.7433218	0.8605959	0.8637888	0.8798209
ENRON	0.1534839	1.0000000	0.2247313	0.1095882	0.4229562	0.1800507
DBLP	0.7845611	0.2427005	0.2691897	0.5206746	0.6423933	0.3293205

Table 3: Global jaccard indices for each data set and clustering algorithm

3 Discussion

As seen in section 2, different algorithms obtained the best results according to the considered metrics described in table 5.

For example in the karate network, Label propagation and Fast Greedy resulted in the best clustering according to both, the metrics as well as the closeness to the canonical clustering. On the other hand, for our DBLP generated network, Louvain generally resulted in the best metrics. Especially interesting is that the best metrics were not obtained by the canonical clustering.

Blindly applying the metrics however hides some important characteristics of the cluster generated. The metrics Edges Inside, Average Degree, Expansion Conductance, Normalized Cut, Max ODF and Average ODF are biased towards clusterings with a fewer number of clusters. This is especially noteworthy in the Barabasi Albert networks where label propagation generated a single large cluster (check the plots in Appendix B).

Network	Vertices	Edges	Mean.Degree	Density
Karate	34	78	4.5882	0.139
Barabasi-Albert	200	800	8	0.0402
ENRON	182	2097	23.044	0.1273
DBLP	2130	4587	4.307	0.002

Table 4: Summary of the properties of the networks used.

Metric	**computation method** (N)
$\uparrow$Internal Density	$\frac{m_s}{n_s(n_s-1)/2}$
$\uparrow$Edges Inside	m_s
$\uparrow$Average Degree	$\frac{2m_s}{n_s}$
$\downarrow$Expansion	$\frac{c_s}{n_s}$
$\downarrow$Cut Ratio	$\frac{c_s}{n_s(n-n_s)}$
$\downarrow$Conductance	$\frac{c_s}{2m_s+c_s}$
$\downarrow$Normalized Cut	$\frac{c_s}{2m_s+c_s} + \frac{c_s}{2(m-m_s)+c_s}$
$\downarrow$Maximum ODF	$\max_u \in S \frac{\lvert\{(u,v)\in E : v\notin S\}\rvert}{d(u)}$
$\downarrow$Average ODF	$\frac{1}{n_s}\sum_u \in S \frac{\lvert\{(u,v)\in E : v\notin S\}\rvert}{d(u)}$

Table 5: Metrics for evaluating the quality of a clustering

Thus judging the quality of a clustering solely on these metrics might not be sufficient. Depending on the use case, it might be desirable to generate clusterings that are close to the canonical clustering. The clusterings favored by the metrics rarely coincide with the actual canonical clustering and therefore don't seem to capture the property of real world clusters well.

In the networks we analyzed, we can't see a general trend on which algorithms are best at generating clusterings favoring certain metrics. This may however only reflect on the networks we analyzed and not on the clustering algorithms themselves. In the case of DBLP, our sampling method is a random walk which may impact the optimal clustering method.

Often, the best metrics obtained were the same for multiple algorithms. This might likewise be related to the networks we chose.

Another thing to consider is that all of the networks considered were either fully connected or filtered to be fully connected to apply all algorithms. Potentially partially disconnected networks might lead to different results. Likewise we only considered unweighted networks. Adding weights to the edges might similarly impact the results obtained.

Lastly, the Jaccard similarity gives a measure of the similarity of two different clusterings.

For ENRON and DBLP, we applied Label Propagation and Louvin respectively as the baseline. In the table 3 it can be seen that the Jaccard similarity produced by the Louvain algorithm is significantly lower than 1 even for clusterings generated by the same method. Meaning, that the clusterings produced are quite different despite using the same algorithm. This shows that the clustering algorithms, Louvain, is quite unstable in it's output.

As we have seen in the previous labs, in many cases, the best algorithm to apply to a network heavily depends on what you want to discover or achieve. The data we have produced for this lab shows that this is the case as well when it comes to clustering algorithms and understanding your network structure and what properties your clustering should exhibit is essential for selecting the best clustering algorithm.

4 Methods

4.1 Description of the networks

We performed our analysis over four networks:

- **Zachary's karate club** is a social network of a university karate club described in the paper [11] by Wayne W. Zachary. It consists of a network of 34 vertices that represent its members with a link between them if they interacted outside. This network has a ground truth given by a conflict that split the club.
- **Random scale-free network**. We generate an artificial network using the Barbási-Albert model [1] that allows us to generate random scale-free networks using a preferential attachment which allows us to know the ground truth. Our network had 200 vertices with 800 edges, with 4 clusters with equal probability and an affinity of 1 and 0.1.
- **ENRON** [4] is an email communication network that covers all email communication within a dataset of around half million emails. These data were originally made public by the Federal Energy Regulatory Commission during its investigation. Network nodes are email addresses, and if an address i sent at least one email to j, the graph contains an undirected edge between i and j. This network is a multigraph, which means that there are parallel edges; we simply transformed it into a simple graph by removing the parallel edges. The final network has 184 vertices and 2216 edges. This network does not have ground truth communities. Moreover, the network is not connected, so we opted to take the largest component by removing isolated nodes.
- **DBLP** provides a comprehensive list of research papers in computer science. In this network [10, 5], each vertex represents one author, and two authors are connected if they publish at least one paper together. The publication venue defines an individual ground truth community. The network has 317080 nodes with 1049866 edges and 5000 communities. This network was too large, so we opt for taking a random induced subgraph (explained in Section 4.2).

In table 6, we show a small summary of the networks properties that we used.

Network	Vertices	Edges	Mean.Degree	Density
Karate	34	78	4.5882	0.139
Barabasi-Albert	200	800	8	0.0402
ENRON	182	2097	23.044	0.1273
DBLP	2130	4587	4.307	0.002

Table 6: Summary of the properties of the networks used.

4.2 Sampling by Random Walk

Firstly, we proposed that the random sampling of the DBLP network be performed by uniformly sampling the vertices and taking the induced subgraph. We observed that the mean degree decreased from around 6 to almost 2. This behavior was explained by [9] which led to biased degrees. Instead, we proposed to use a random walk, although it is known that it also produces biased subgraphs [3, 6]. They propose alternative methods for samplings that preserve properties, but we believe that implementing them was out of the scope of this report.

A Graph Metrics

	Louvain	Label Propagation	Walktrap	Edge Betweenness	Fast Greedy	Spin-Glass	ground truth
size	9.5882353	13.8235294	10.0000000	7.1176471	13.8235294	9.5882353	17.0588235
internal density	1.2949198	1.0659170	1.3225490	1.5253268	1.0659170	1.2949198	0.7688581
edges inside	50.3529412	83.3529412	51.8235294	30.7647059	83.3529412	50.3529412	104.8235294
av degree	5.0588235	5.8235294	5.0588235	3.9117647	5.8235294	5.0588235	6.1470588
FOMD	0.2647059	0.3529412	0.2647059	0.1470588	0.3529412	0.2647059	0.4117647
expansion	3.4705882	1.9411765	3.4705882	5.7647059	1.9411765	3.4705882	1.2941176
cut ratio	0.1431189	0.0937969	0.1454063	0.2140606	0.0937969	0.1431189	0.0763889
conductance	0.2569623	0.1435770	0.2548448	0.4194006	0.1435770	0.2569623	0.0951872
norm cut	0.3793707	0.2434733	0.3813161	0.5841706	0.2434733	0.3793707	0.1909091
max ODF	0.4357699	0.3907872	0.5117297	0.5870146	0.3907872	0.4357699	0.3891161
average ODF	0.1833604	0.1141638	0.1849360	0.3130916	0.1141638	0.1833604	0.0749885
flake ODF	0.0588235	0.0000000	0.0882353	0.2352941	0.0000000	0.0588235	0.0000000
density ratio	0.8775114	0.9042410	0.8684636	0.8252310	0.9042410	0.8775114	0.9001770
modularity	0.4197896	0.3990796	0.4111604	0.3618508	0.3990796	0.4197896	0.3714661
clustering coef	0.6046786	0.5372639	0.6138152	0.4505199	0.6170631	0.5817454	0.5340685
graph_order	34.0000000	34.0000000	34.0000000	34.0000000	34.0000000	34.0000000	34.0000000
n_clusters	4.0000000	3.0000000	4.0000000	6.0000000	3.0000000	4.0000000	2.0000000
mean_cluster_size	8.5000000	11.3333333	8.5000000	5.6666667	11.3333333	8.5000000	17.0000000
coverage	0.7445887	0.8571429	0.7445887	0.5757576	0.8571429	0.7445887	0.9047619
global density ratio	0.7586439	0.7881438	0.7427326	0.6646320	0.7881438	0.7586439	0.8004386
VIdist_to_GT	0.9078217	0.4216651	0.8729384	1.6382472	0.4216651	0.9078217	0.0000000

Table 7: Significance metrics of karate network

	Louvain	Label Propagation	Walktrap	Edge Betweenness	Fast Greedy	Spin-Glass	ground truth
size	23.9200000	200.0000000	23.0900000	40.6600000	28.9200000	26.0300000	50.4100000
internal density	0.1464999	0.0402010	0.2252489	0.4113405	0.1363536	0.1436153	0.0417554
edges inside	38.3050000	800.0000000	65.9400000	146.7850000	51.8650000	46.2400000	48.0800000
av degree	1.5800000	4.0000000	1.8150000	2.0150000	1.7500000	1.7550000	0.9850000
FOMD	0.1350000	0.4950000	0.1700000	0.1750000	0.1700000	0.1550000	0.1100000
expansion	4.8400000	0.0000000	4.3700000	3.9700000	4.5000000	4.4900000	6.0300000
cut ratio	0.0274631		0.0257214	0.0253710	0.0262705	0.0258238	0.0401307
conductance	0.6024022	0.0000000	0.5879614	0.5689075	0.5629768	0.5605864	0.8249440
norm cut	0.7725800		0.7702422	0.8654006	0.7399986	0.7213807	1.3281677
max ODF	0.7851892	0.0000000	0.7565357	0.7046858	0.7479673	0.7403859	1.0000000
average ODF	0.5364645	0.0000000	0.5230527	0.5122148	0.5103769	0.5087494	0.7930540
flake ODF	0.4350000	0.0000000	0.5600000	0.5200000	0.3950000	0.3750000	0.8400000
density ratio	0.8037900		0.8504611	0.8304326	0.7940070	0.8164088	-6.1196130
modularity	0.2745742	0.0000000	0.2145234	0.1601469	0.2912414	0.3043156	-0.0525914
clustering coef	0.1471581	0.1051262	0.1297047	0.1476865	0.1166031	0.1406470	0.0817483
graph_order	200.0000000	200.0000000	200.0000000	200.0000000	200.0000000	200.0000000	200.0000000
n_clusters	9.0000000	1.0000000	18.0000000	46.0000000	8.0000000	8.0000000	4.0000000
mean_cluster_size	22.2222222	200.0000000	11.1111111	4.3478261	25.0000000	25.0000000	50.0000000
coverage	0.3950000	1.0000000	0.4537500	0.5037500	0.4375000	0.4387500	0.2462500
global density ratio	0.6012572		0.6993590	0.5096080	0.5803467	0.6319086	-1.0220569
VIdist_to_GT	4.9283194	1.9939661	5.2662015	4.8533377	4.7366883	4.7641969	0.0000000

Table 8: Significance metrics of Barabasi Albert Blocks generated network

	Louvain	Label Propagation	Walktrap	Edge Betweenness	Fast Greedy	Spin-Glass
size	28.0659341	163.0989011	40.9010989	19.9450549	76.9780220	27.9010989
internal density	0.4629362	0.1681769	0.4414078	0.5762005	0.2575144	0.4714773
edges inside	172.1483516	1918.2747253	318.6263736	223.0824176	655.1538462	178.0659341
av degree	5.9615385	11.3131868	6.6428571	4.5989011	8.1868132	6.0879121
FOMD	0.0934066	0.4725275	0.2197802	0.1978022	0.3076923	0.0989011
expansion	11.1208791	0.4175824	9.7582418	13.8461538	6.6703297	10.8681319
cut ratio	0.0727568	0.0220930	0.0702472	0.0865590	0.0637434	0.0714230
conductance	0.4729326	0.0296510	0.4417027	0.6962676	0.2960506	0.4688008
norm cut	0.6223755	0.3892819	0.6293100	0.9276610	0.5312898	0.6164363
max ODF	0.7251472	0.4774987	0.7225894	0.8138148	0.5658964	0.7056839
average ODF	0.4074784	0.0303193	0.4090812	0.6785295	0.2565294	0.4019890
flake ODF	0.3241758	0.0000000	0.3021978	0.6098901	0.0604396	0.3186813
density ratio	0.8320747	0.8468334	0.8009065	0.8345215	0.7078507	0.8367735
modularity	0.3458183	0.0284290	0.2950982	0.1388929	0.2646726	0.3509633
clustering coef	0.6313311	0.3944489	0.5985013	0.6609014	0.6758753	0.6535819
graph_order	182.0000000	182.0000000	182.0000000	182.0000000	182.0000000	182.0000000
n_clusters	7.0000000	2.0000000	12.0000000	100.0000000	4.0000000	8.0000000
mean_cluster_size	26.0000000	91.0000000	15.1666667	1.8200000	45.5000000	22.7500000
coverage	0.5174058	0.9818789	0.5765379	0.3991416	0.7105389	0.5283739
global density ratio	0.6720036	0.6834442	0.5845891	0.6480276	0.4105592	0.6883573

Table 9: Significance metrics of enron network

	Louvain	Label Propagation	Walktrap	Edge Betweenness	Fast Greedy	Spin-Glass
size	51.2733397	11.2136670	19.7401347	55.9027911	59.6843118	91.6506256
internal density	0.1208516	0.4790165	0.4111536	0.0999436	0.1217949	0.0582293
edges inside	108.4894129	25.6251203	51.5606352	122.7045236	127.3118383	191.7516843
av degree	2.2267353	2.0019249	2.0721848	2.2204042	2.2240664	2.1953802
FOMD	0.3455245	0.2781521	0.2978826	0.3460058	0.3445621	0.3383061
expansion	0.2142541	0.6698749	0.5293551	0.2329163	0.2155919	0.2829644
cut ratio	0.0001090	0.0003240	0.0002582	0.0001156	0.0001075	0.0001427
conductance	0.0505868	0.1622873	0.1299318	0.0528932	0.0506092	0.0628818
norm cut	0.0528051	0.1631309	0.1316489	0.0545452	0.0529275	0.0659480
max ODF	0.4505721	0.4731290	0.4662499	0.4965947	0.4635304	0.5389403
average ODF	0.0439747	0.1336347	0.1098625	0.0506543	0.0448423	0.0616079
flake ODF	0.0024062	0.0288739	0.0178056	0.0043311	0.0033686	0.0038499
density ratio	0.9984335	0.9991332	0.9985666	0.9982526	0.9981309	0.9966883
modularity	0.9271365	0.8480436	0.8695307	0.9214453	0.9244761	0.8970664
clustering coef	0.5697728	0.6201277	0.5800038	0.5773218	0.5652247	0.6546957
graph_order	2078.0000000	2078.0000000	2078.0000000	2078.0000000	2078.0000000	2078.0000000
n_clusters	51.0000000	250.0000000	223.0000000	46.0000000	51.0000000	25.0000000
mean_cluster_size	40.7450980	8.3120000	9.3183857	45.1739130	40.7450980	83.1200000
coverage	0.9520181	0.8566722	0.8867381	0.9501647	0.9538715	0.9394563
global density ratio	0.9974996	0.9983464	0.9976741	0.9971519	0.9971878	0.9941178

Table 10: Significance metrics of DBLP network

B Networks with clusters

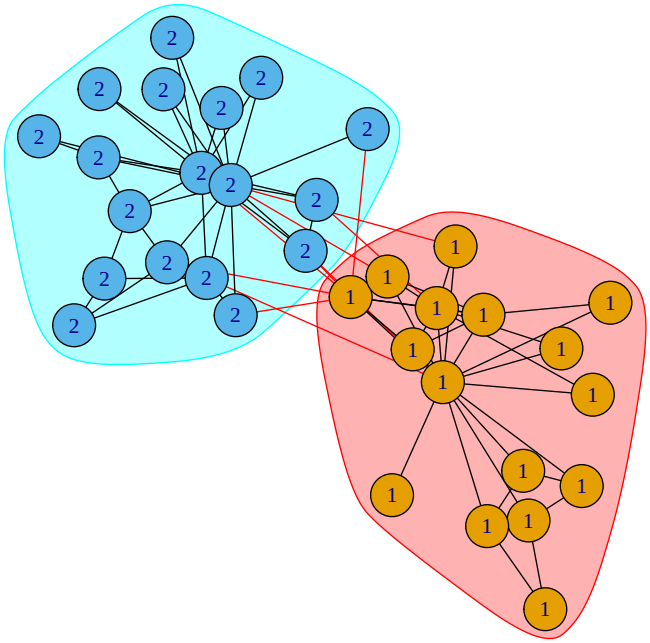

Figure 2: Network karatewith label using the cannonical and colors the clustering produced by baseline

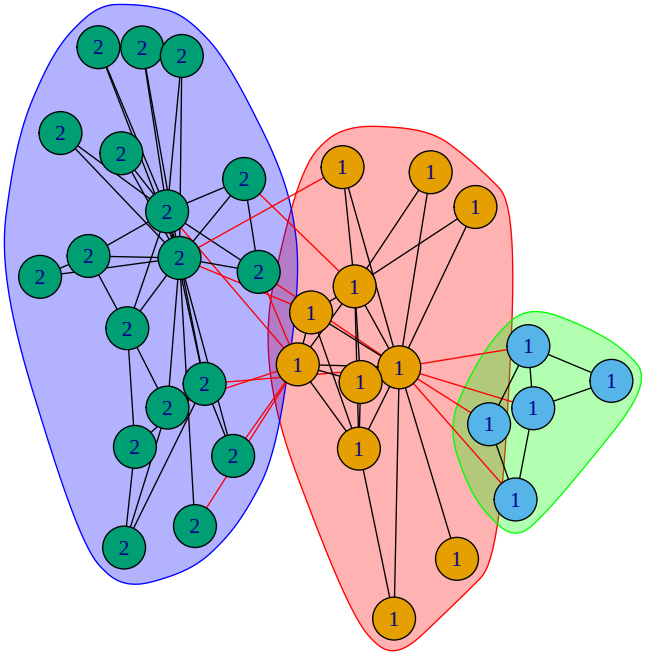

Figure 3: Network karatewith label using the cannonical and colors the clustering produced by Louvain

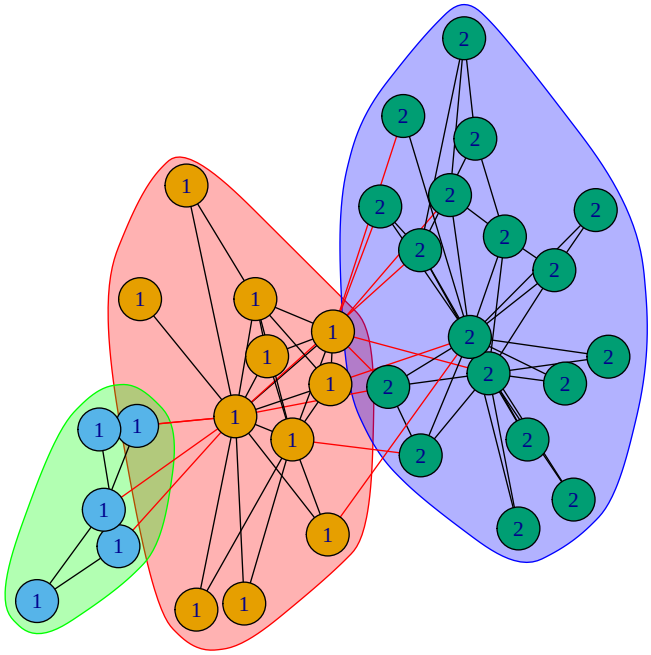

Figure 4: Network karatewith label using the cannonical and colors the clustering produced by Label Propagation

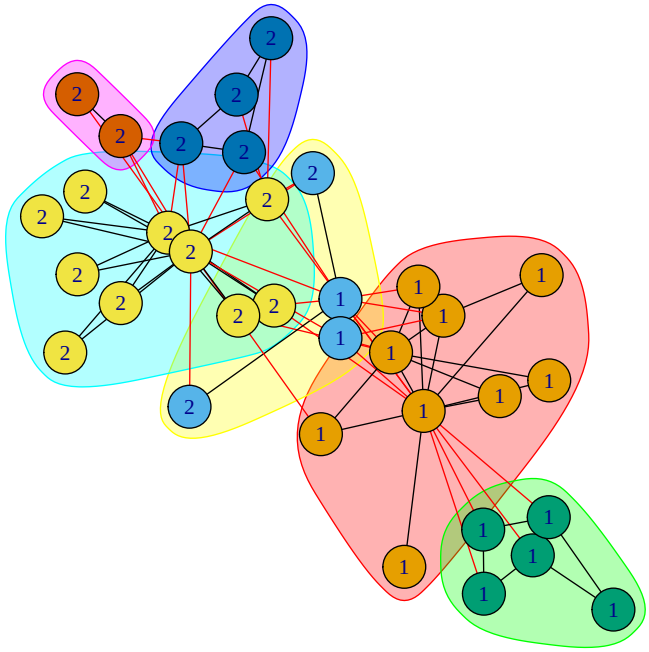

Figure 5: Network karatewith label using the cannonical and colors the clustering produced by Edge Betweenness

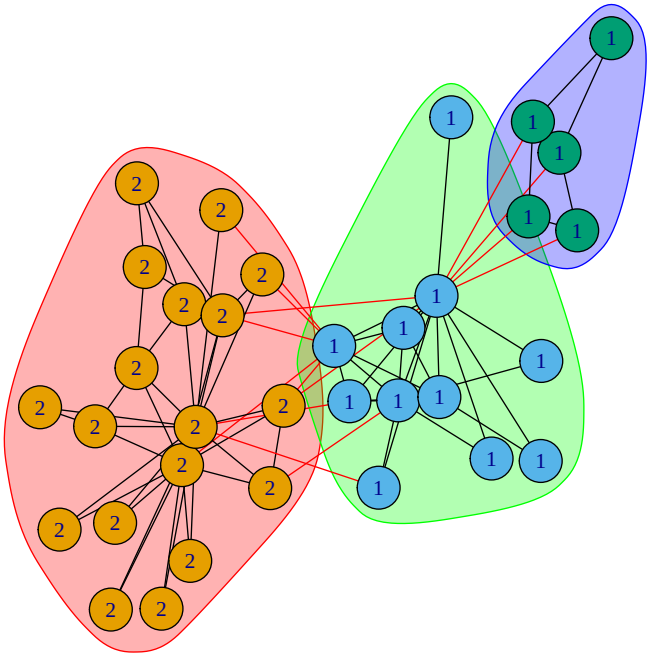

Figure 6: Network karatewith label using the cannonical and colors the clustering produced by Fast Greedy

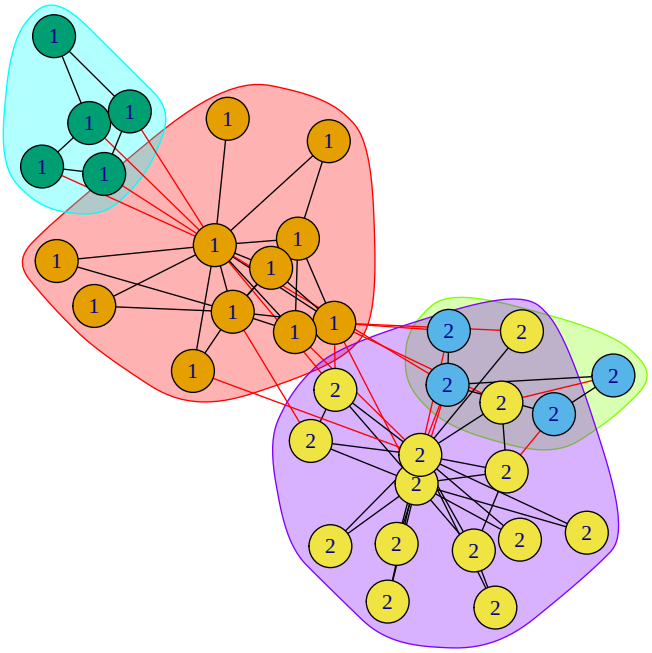

Figure 7: Network karatewith label using the cannonical and colors the clustering produced by Spin-Glass

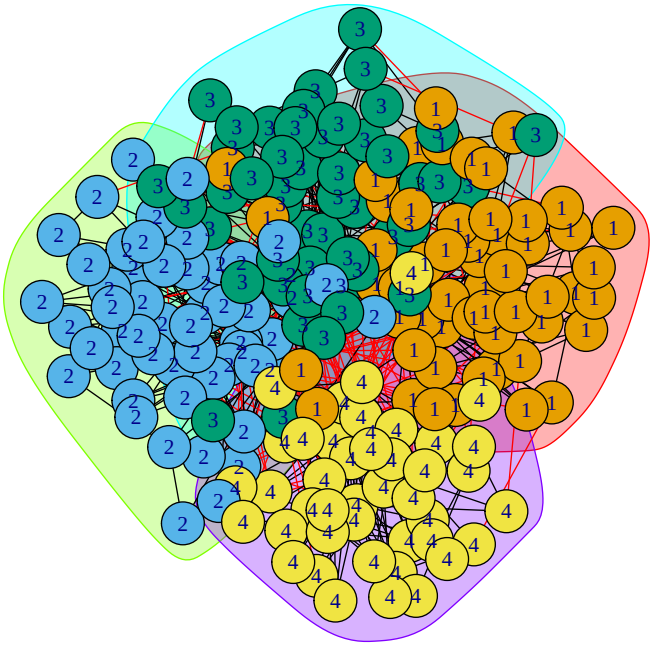

Figure 8: Network BAwith label using the cannonical and colors the clustering produced by baseline

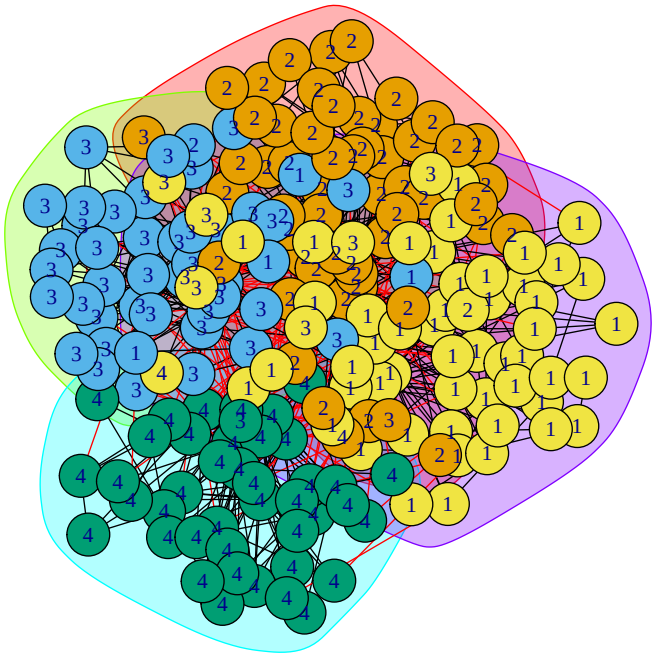

Figure 9: Network BAwith label using the cannonical and colors the clustering produced by Louvain

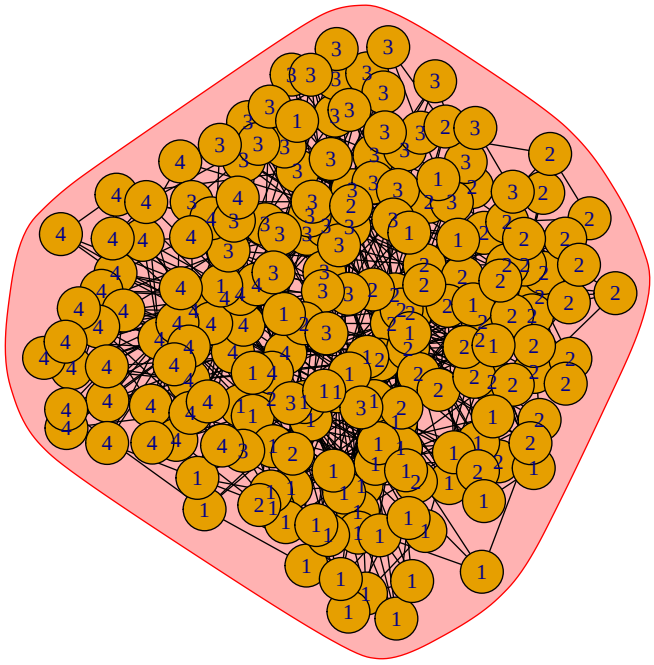

Figure 10: Network BAwith label using the cannonical and colors the clustering produced by Label Propagation

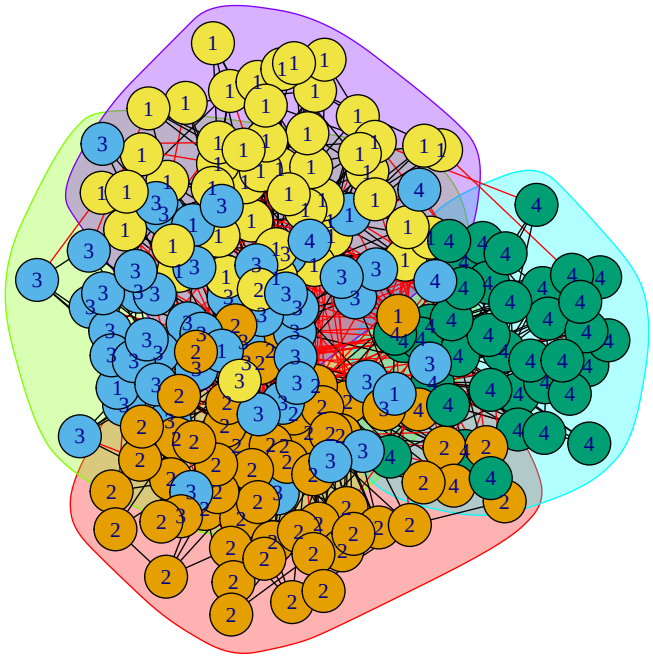

Figure 11: Network BAwith label using the cannonical and colors the clustering produced by Edge Betweenness

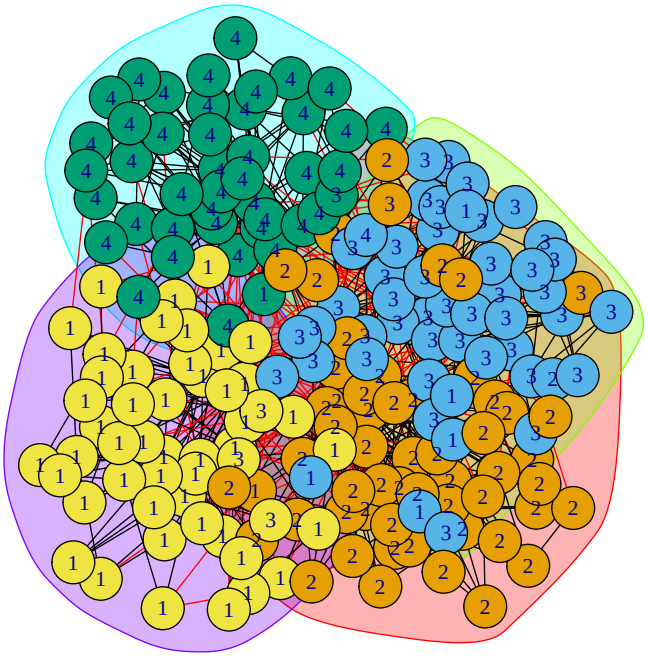

Figure 12: Network BAwith label using the cannonical and colors the clustering produced by Fast Greedy

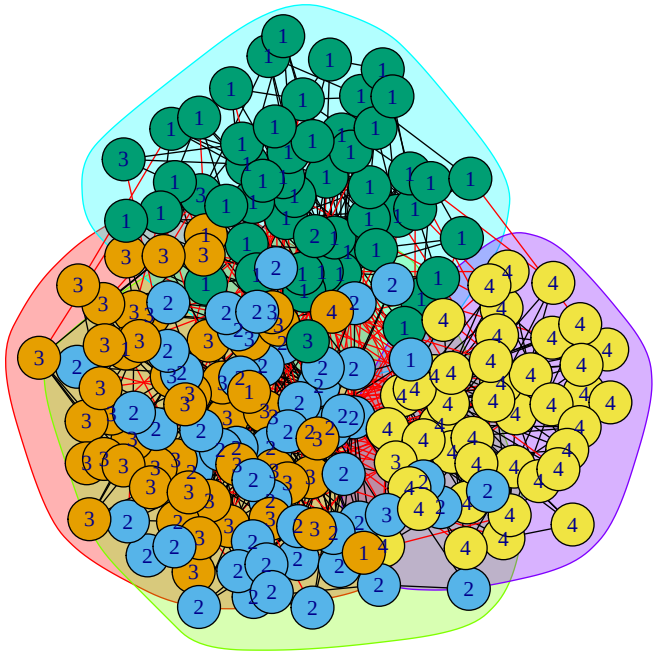

Figure 13: Network BAwith label using the cannonical and colors the clustering produced by Spin-Glass

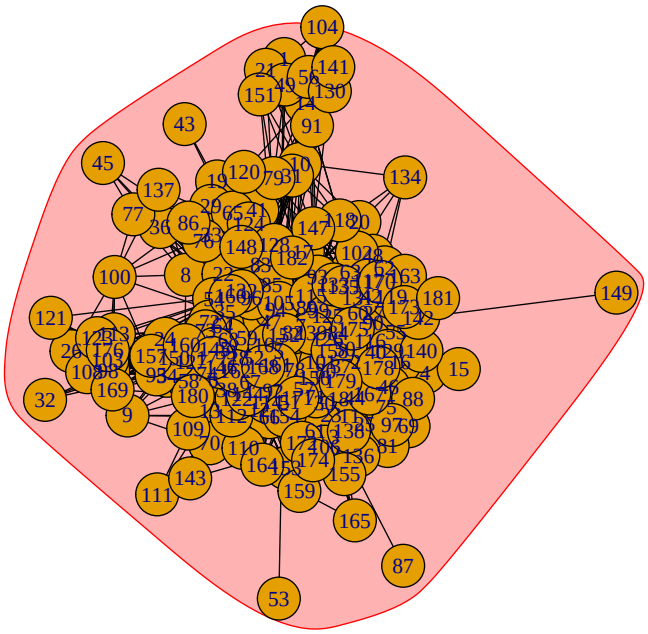

Figure 14: Network enronwith label using the cannonical and colors the clustering produced by baseline

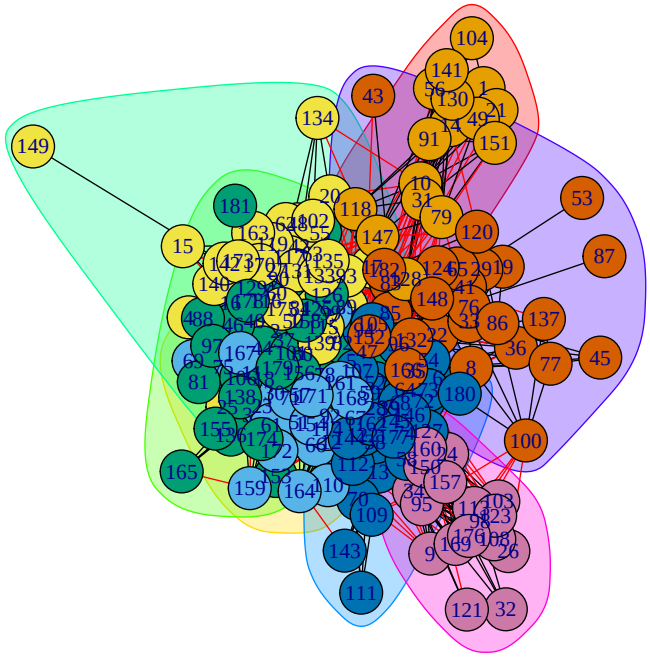

Figure 15: Network enronwith label using the cannonical and colors the clustering produced by Louvain

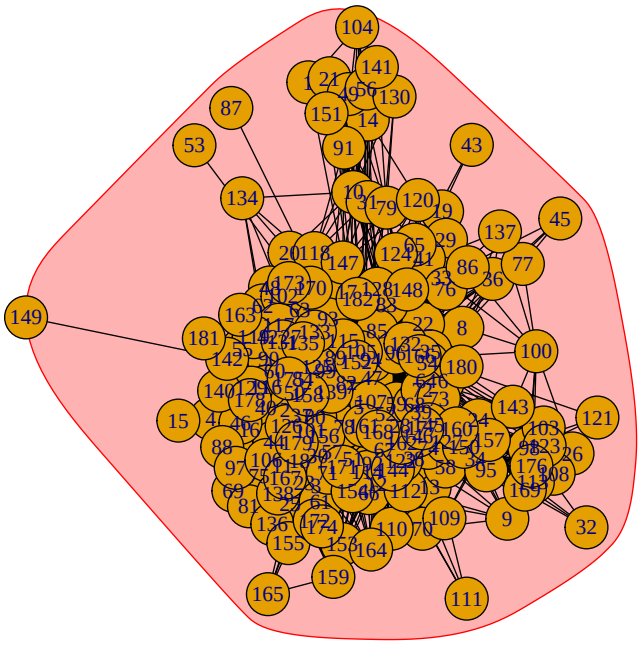

Figure 16: Network enronwith label using the cannonical and colors the clustering produced by Label Propagation

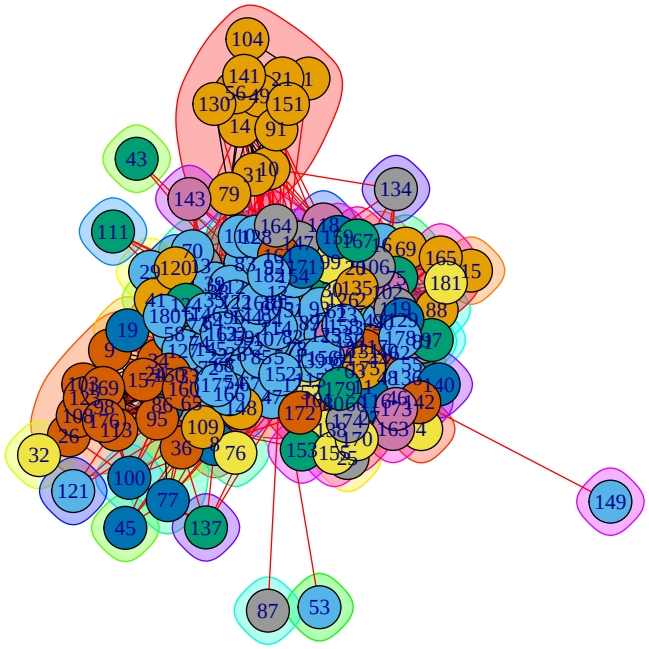

Figure 17: Network enronwith label using the cannonical and colors the clustering produced by Edge Betweenness

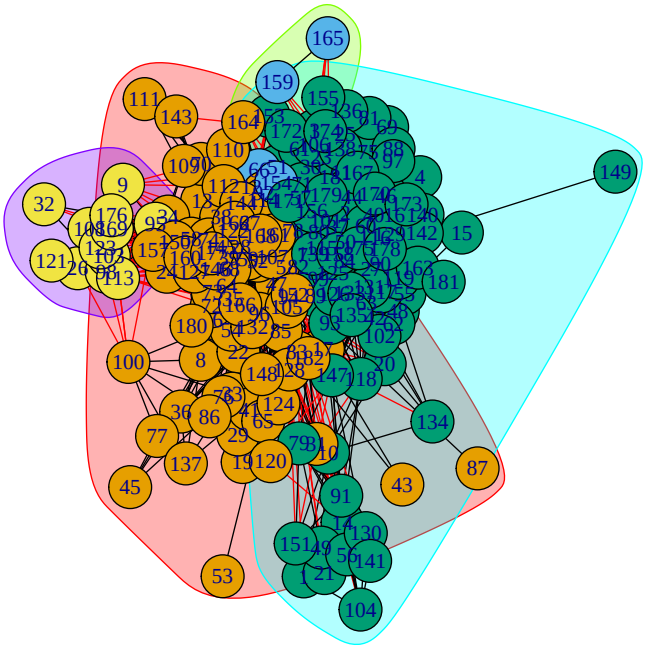

Figure 18: Network enronwith label using the cannonical and colors the clustering produced by Fast Greedy

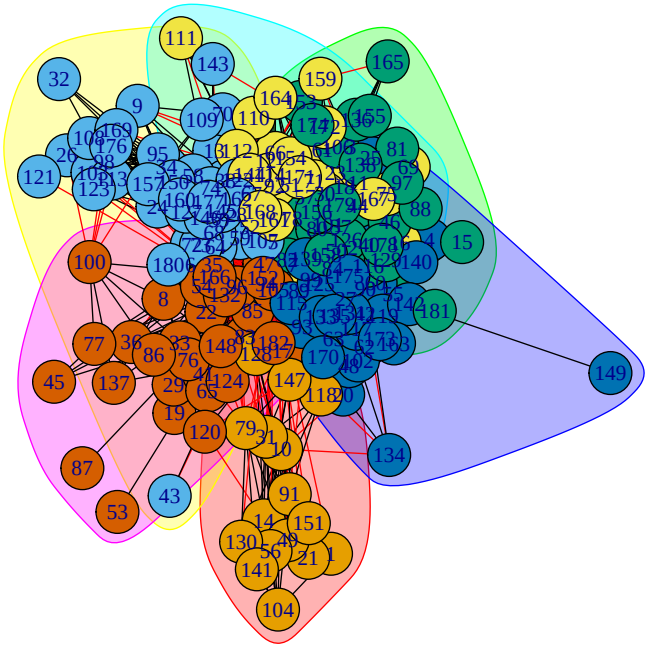

Figure 19: Network enronwith label using the cannonical and colors the clustering produced by Spin-Glass

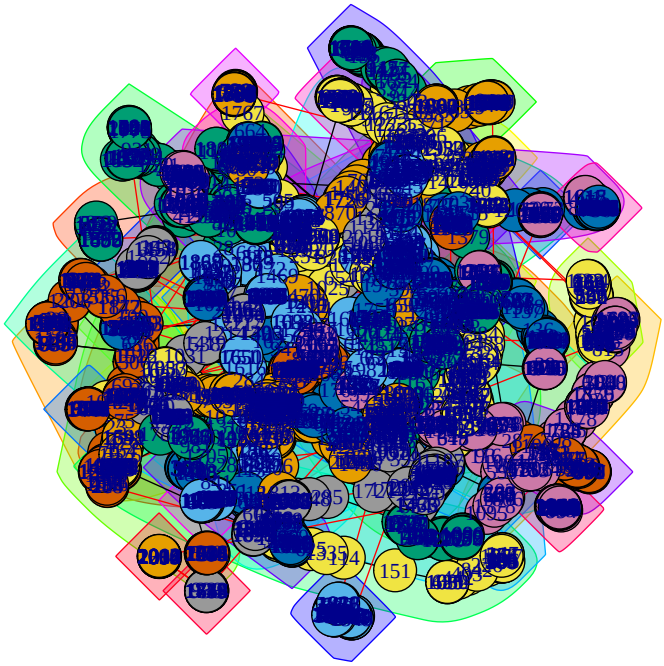

Figure 20: Network dblpwith label using the cannonical and colors the clustering produced by baseline

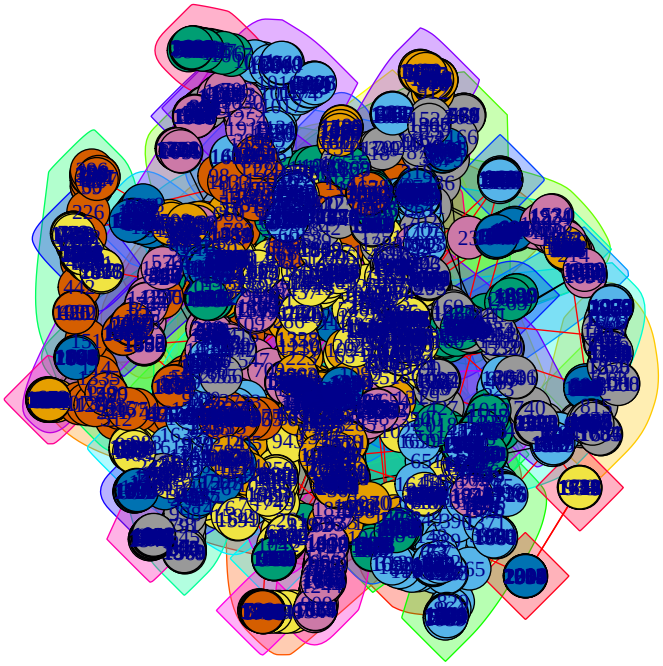

Figure 21: Network dblpwith label using the cannonical and colors the clustering produced by Louvain

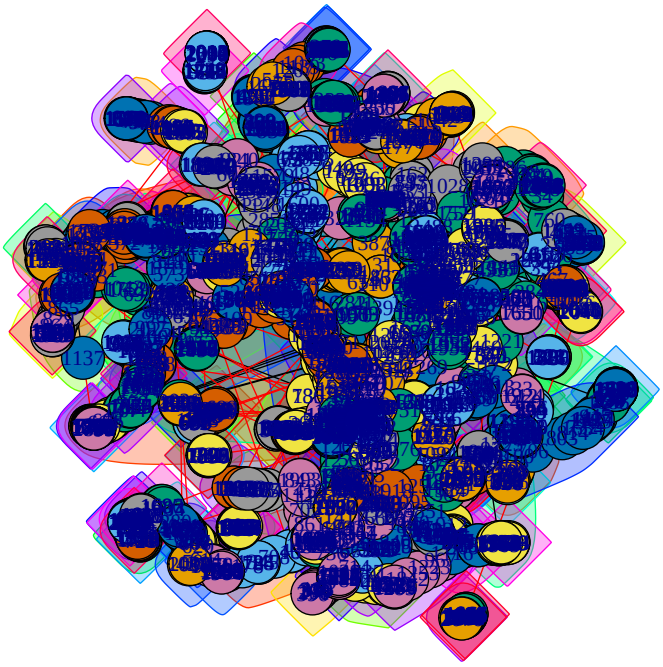

Figure 22: Network dblpwith label using the cannonical and colors the clustering produced by Label Propagation

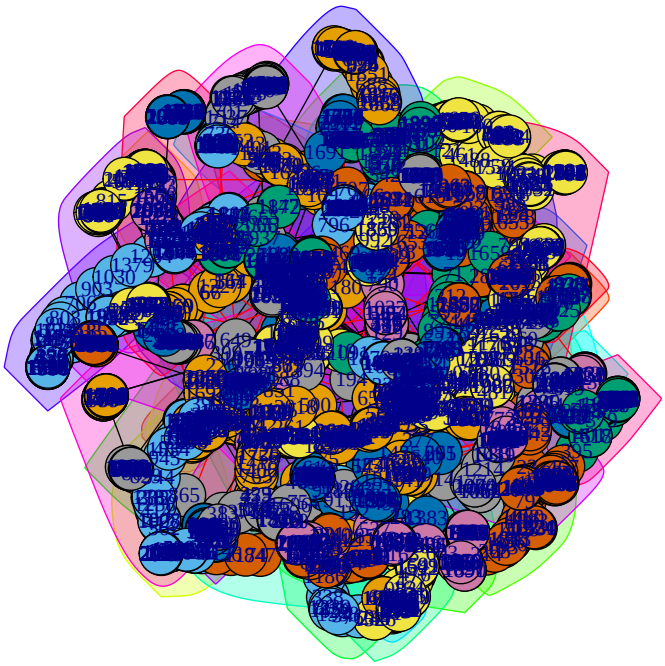

Figure 23: Network dblpwith label using the cannonical and colors the clustering produced by Edge Betweenness

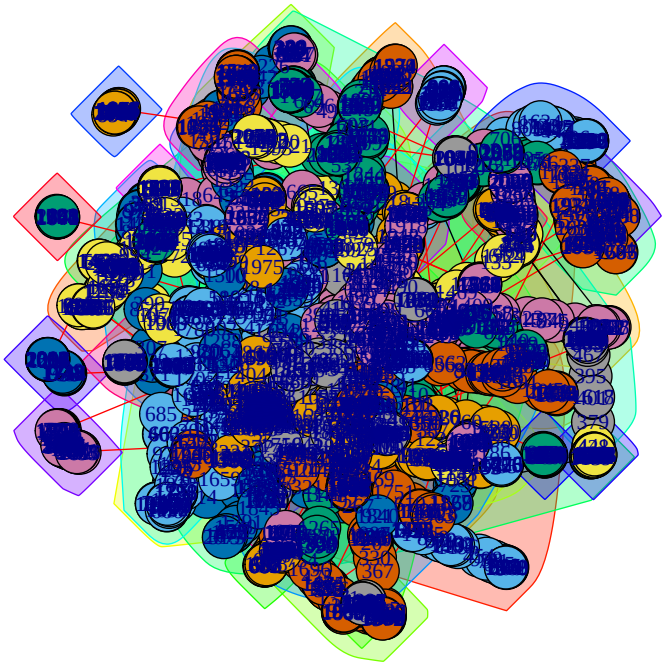

Figure 24: Network dblpwith label using the cannonical and colors the clustering produced by Fast Greedy

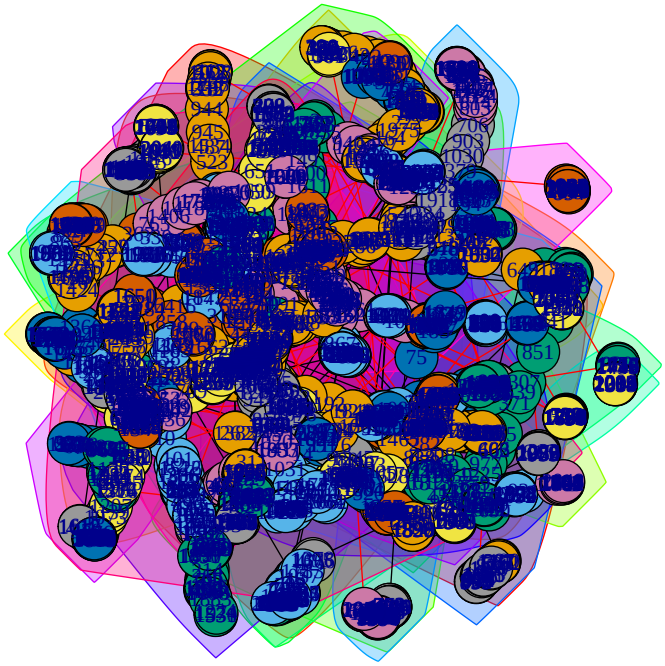

Figure 25: Network dblpwith label using the cannonical and colors the clustering produced by Spin-Glass