Nikhil Paonikar

Dr. Olfa Nasraoui

CECS 621-50 Web Mining

July 15, 2018

SOCIAL NETWORK ANALYSIS/ LINK-MINING

REPORT

DATASET

Bitcoin transactions

Bitcoin transactions are designed to be anonymous. However, multiple factors can dictate whether the owner of a Bitcoin wallet is protecting their identity. Co-relationships and interesting trails of information can be discovered by performing secondary searches for Bitcoin addresses. This dataset is targeted at a particular Bitcoin address to visualize the transactions flowing in and out of it. Then secondary dark web searches using Webhose.io were performed to see if any hidden services where the bitcoin wallets were mentioned can be identified.

Dataset-Properties:

Property	Attribute
Graph-type	Directed
Number of nodes	2068
Number of edges	2175
Dynamic graph	No
Dynamic attributes	No
Multi-graph	No

Link to the source page.

Retrieved dataset.

METRICS

1. Graph-distance

It is the average graph-distance between all pairs of nodes. Connected nodes have a graph distance of 1. The diameter is the longest graph distance between any two nodes in the network; i.e., how far apart are the two most distant nodes.[1] Source code.

2. PageRank distribution

It is an iterative algorithm which measures the importance of each node within a network. The metric assigns each node a probability that is the probability of being at that page after many clicks. The page-rank values are the values in the eigenvector that has the highest corresponding eigenvalue of a normalized adjacency matrix A'. The standard adjacency matrix is normalized so that the columns of the matrix sum to 1.[2]
Source code.

3. Clustering-coefficient

The clustering coefficient (Watts-Strogatz), when applied to a single node, is a measure of how complete the neighborhood of a node is. When applied to an entire network, it is the average clustering coefficient over all of the nodes in the network.[3] Source code.

4. Graph density

It measures the closeness of a network to the state where it is complete. A complete graph has all possible edges and density equal to 1.[4]
Source code.

---

[1] http://www-personal.umich.edu/~mejn/courses/2004/cscs535/review.pdf

[2] http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf

[3] http://worrydream.com/refs/Watts-CollectiveDynamicsOfSmallWorldNetworks.pdf

[4] https://xlinux.nist.gov/dads/HTML/sparsegraph.html

ALGORITHMS

1. Force Atlas

The Force Atlas layout algorithm is a spatial layout algorithm. It's used for real-world networks like web networks. This algorithm is a type of force-directed algorithms. It emphasizes the quality in lieu of speed; i.e., the Force Atlas layout algorithm gives more weight to the quality of the layout than the speed with which it has been computed. This is especially true in the case of large networks. However, as far as small networks are concerned, Force Atlas works perfectly.

Pseudocode[5]

```
set initial node velocities to zero
set initial node positions to random non-overlapping values
loop
    set total_KE = 0            // running sum of total kinetic energy (KE) over all particles
    for this_node               // loop over every node
        set net-force = 0       // running sum of total force on this particular node

        for each other_node
            net-force = net-force + Coulomb_repulsion( this_node, other_node )
        next node

        for each spring connected to this_node
            net-force = net-force + Hooke_attraction( this_node, spring )
        next spring

        // update velocity and position
        this_node.velocity = (this_node.velocity + timestep * net-force) * damping
        this_node.position = this_node.position + timestep * this_node.velocity
        total_KE = total_KE + this_node.mass * (this_node.velocity)^2
    next node
until total_KE is less than some small number  // the simulation has converged
```

5 https://www.ideals.illinois.edu/bitstream/handle/2142/42195/Sudipta_Dutta.pdf

2. Noverlap

Noverlap algorithm is used after any layout to prevent node overlapping while keeping the shape of the graph. It is optimized for big graphs.

Pseudocode[6]

1. The graph is divided into a grid of squares.
2. Nodes are mapped onto these squares.
    - A node could be a subset of multiple squares, especially if it's big.
3. A list of "proximity relations", which approximate if two nodes are in the same area is obtained.
    - Two nodes have a *proximity relation* if they are a subset of the same square.
4. Each of these relations is tested to eliminate nodes that actually do not overlap.
5. A repulsive force is applied between the nodes that overlap.

---

6 https://gephi.org/gephi-toolkit/0.9.1/apidocs/org/gephi/layout/plugin/noverlap/NoverlapLayout.html

3. Yifan Hu

The Yifan Hu Multilevel is a layout algorithm. It reduces the algorithmic complexity of visualizing a layout by incorporating the best features of force-directed algorithms and another multilevel algorithm. Here, only pairs of adjacent nodes are taken into consideration, which leads to reduced complexity in the Yifan Hu layout algorithm and hence the new layout is computed much faster. Its compatibility with large networks is very high. This algorithm is generally used to restructure graphs.

Pseudocode[7]

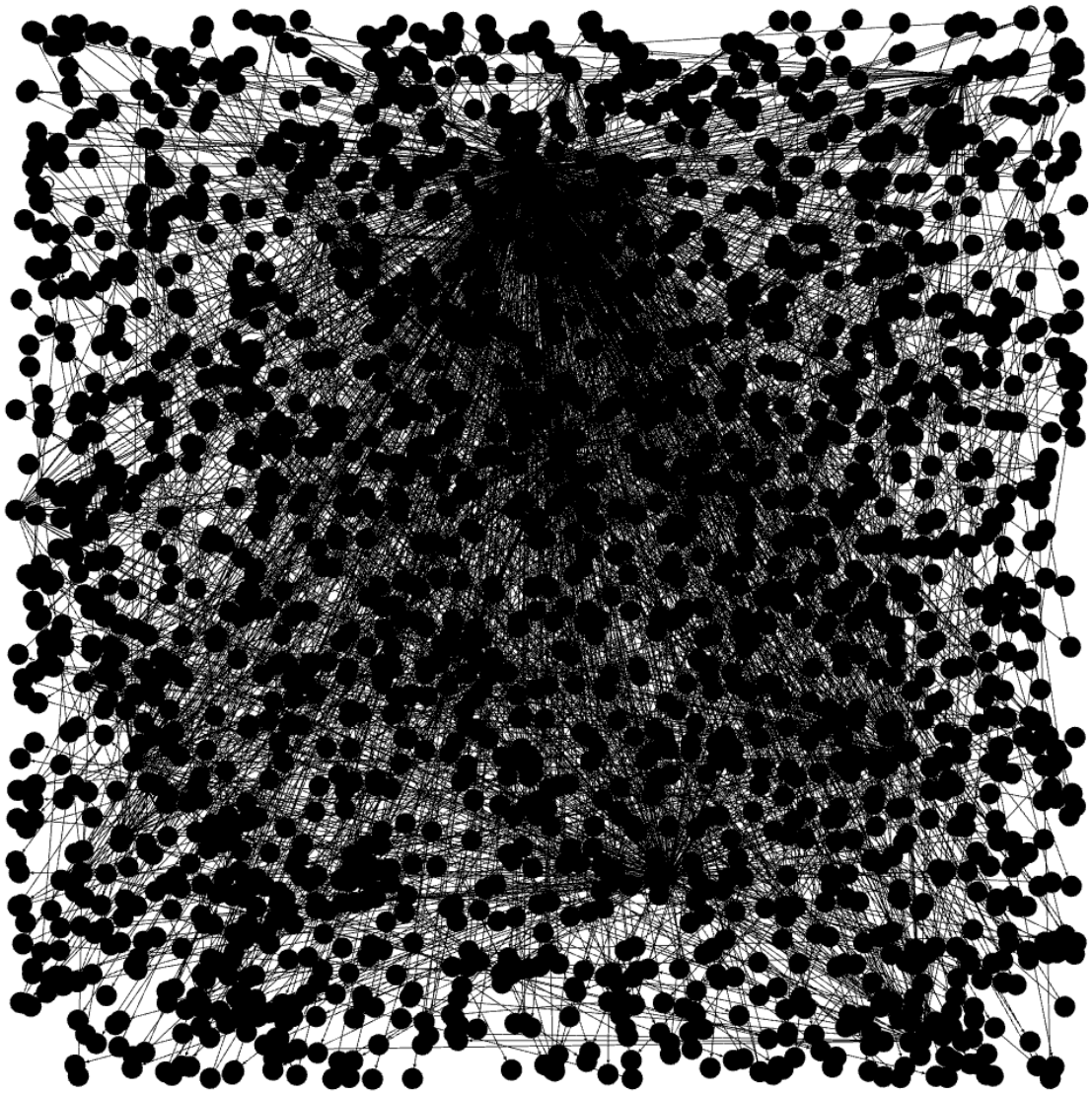
[7] http://www.mathematica-journal.com/issue/v10i1/contents/graph_draw/graph_draw.pdf

- ForceDirectedAlgorithm(G, x, tol) {
  - converged = FALSE;
  - step = initial step length;
  - Energy = Infinity
  - while (converged equals FALSE) {
    - $x^0 = x$;
    - Energy^0 = Energy; Energy = 0;
    - for $i \in V$ {
      - $f = 0$;
      - for $(j \leftrightarrow i)$ $f := f + \frac{f_r(i,j)}{\| x_j - x_i \|}(x_j - x_i)$;
      - for $(j \neq i,\ j \in V)$ $f := f + \frac{f_r(i,j)}{\| x_j - x_i \|}(x_j - x_i)$;
      - $x_{i:} = x_i + \text{step} * (f / \| f \|)$;
      - Energy := Energy + $\| f \|^2$;
    - }
    - step := update_steplength (step, Energy, Energy^0);
    - if ($| \ |x - x^0| \ |$ < K tol) converged = TRUE;
  - }
  - return x;
- }

- Coarsest graph layout
  - if (n^{i+1} < MinSize or $n^{i+1} / n^i > \rho$) {
    - $x^{i:}$ = random initial layout
    - x^i = ForceDirectedAlgorithm(G^i , x^i, tol)
    - return x^i
  - }
- The coarsening phase:
  - set up the $n^i \times n^{i+1}$ prolongation matrix P^i
  - $G^{i+1} = P^{i\,T}\ G^i\ P^i$
  - x^{i+1} = MultilevelLayout(G^{i+1}, tol)
- The prolongation and refinement phase:
  - prolongate to get initial layout: $x^i = P^i\ x^{i+1}$
  - refinement: x^i = ForceDirectAlgorithm(G^i , x^i, tol)
  - return x^i

RESULTANT VISUALIZATIONS AND OUTPUTS

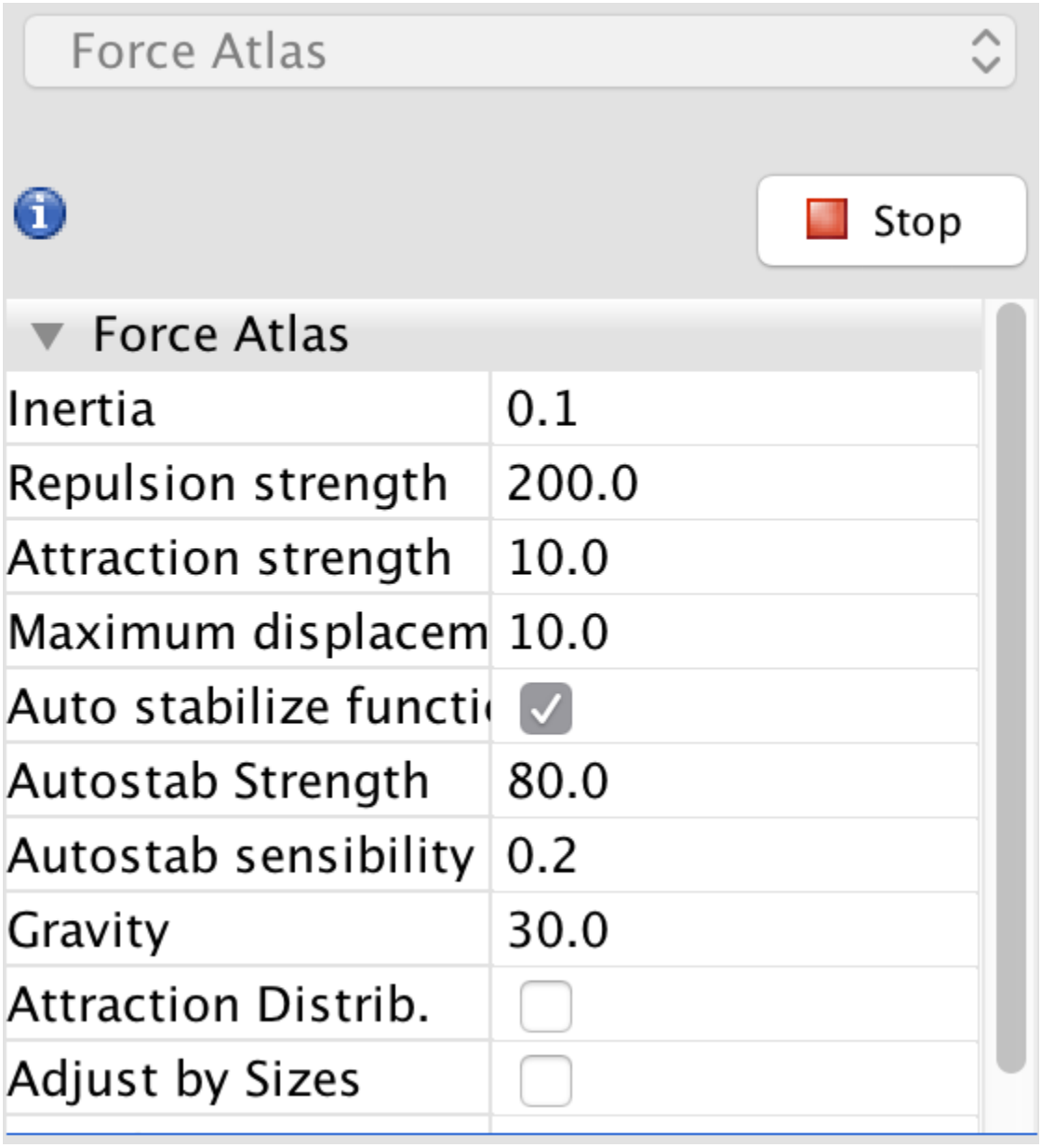
1. Default visualization



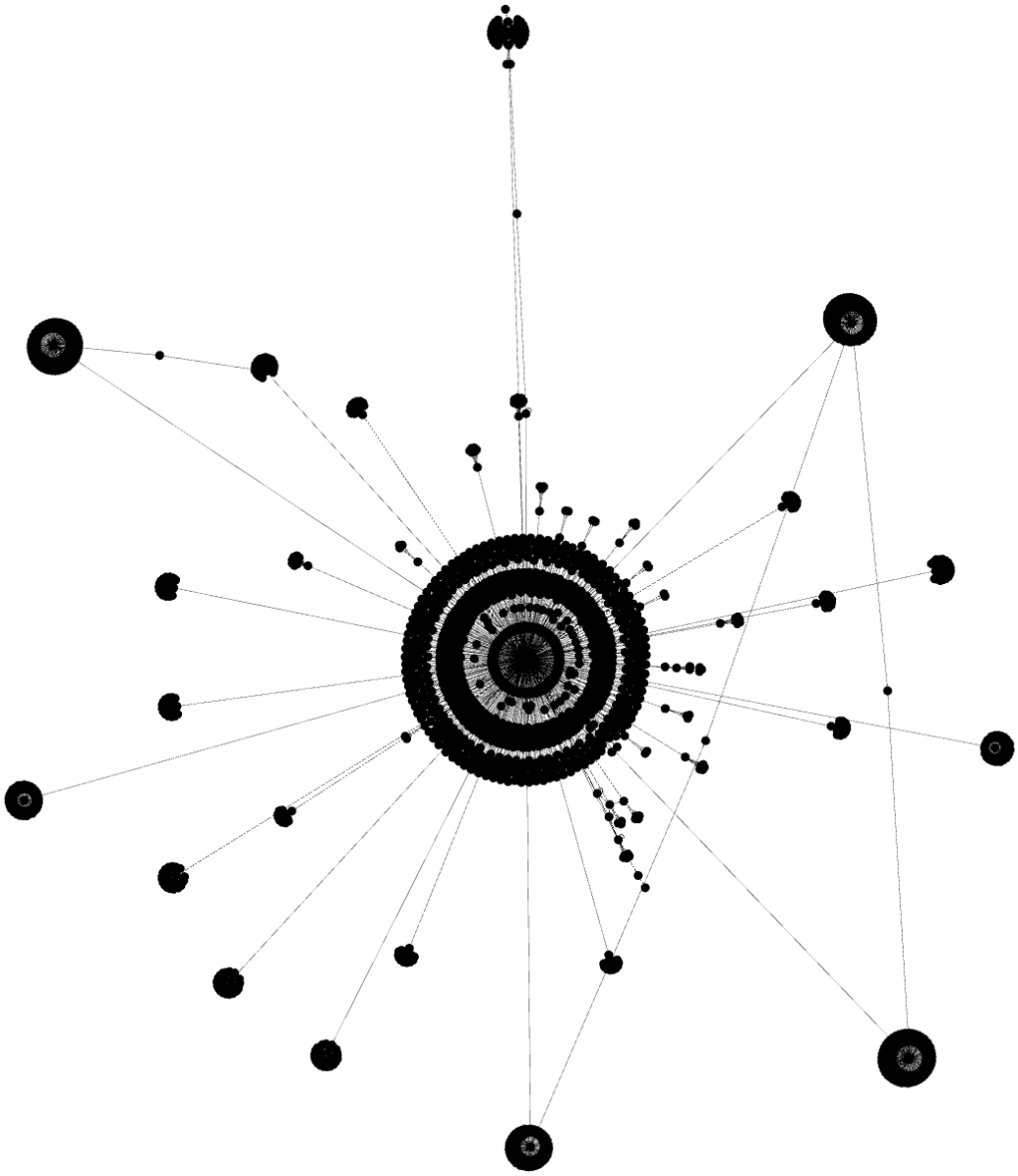
Default visualization of the dataset

The data was compiled into a dataset using Python and then converted into a GEXF file, executed on Gephi 0.9.2.

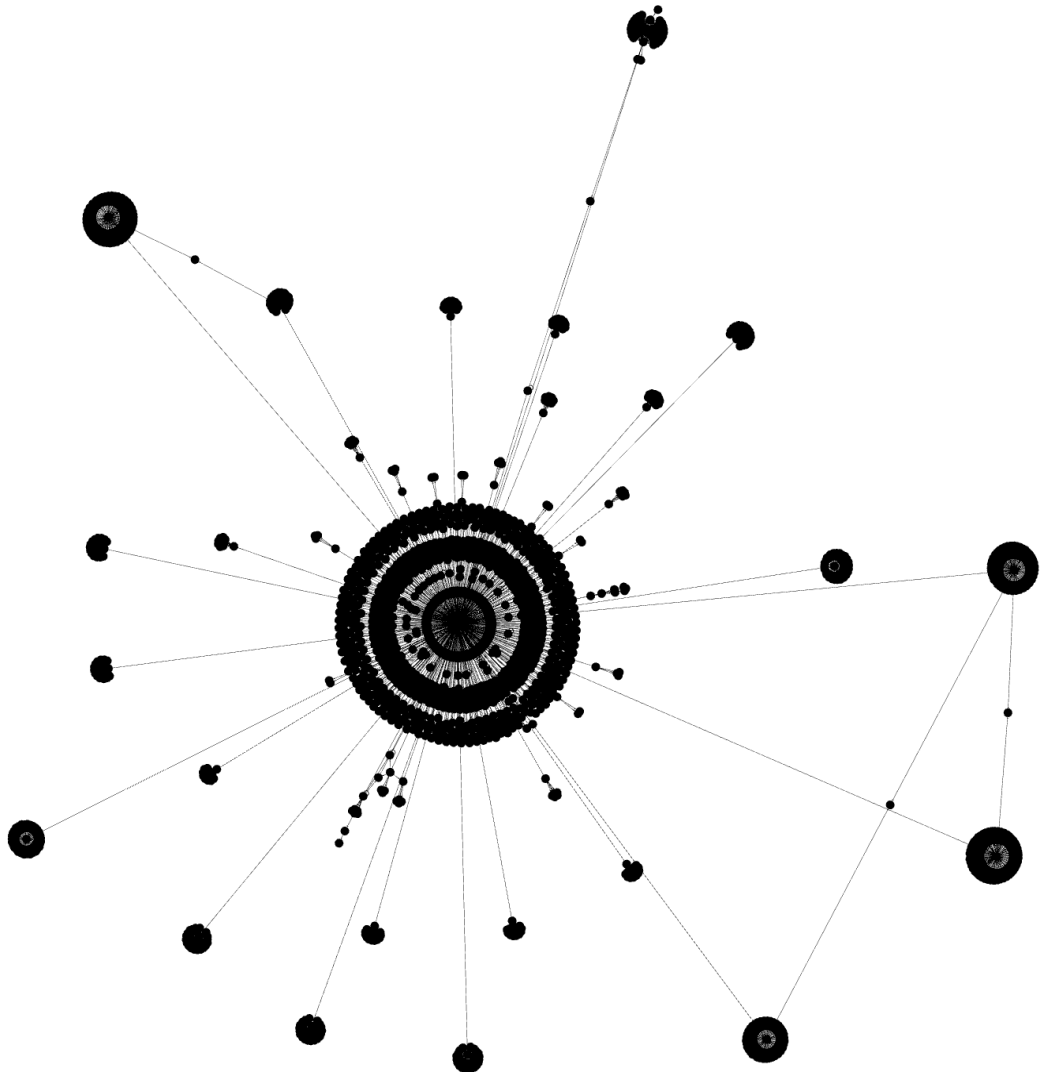
2. Force Atlas



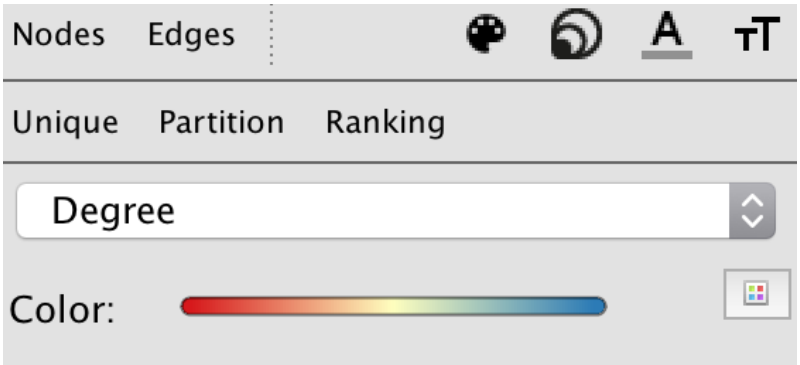
Next, we apply the above settings to the Forced Atlas layout algorithm to visualize the dataset.



The graph undergoes transformation when the Forced Atlas layout algorithm is applied.

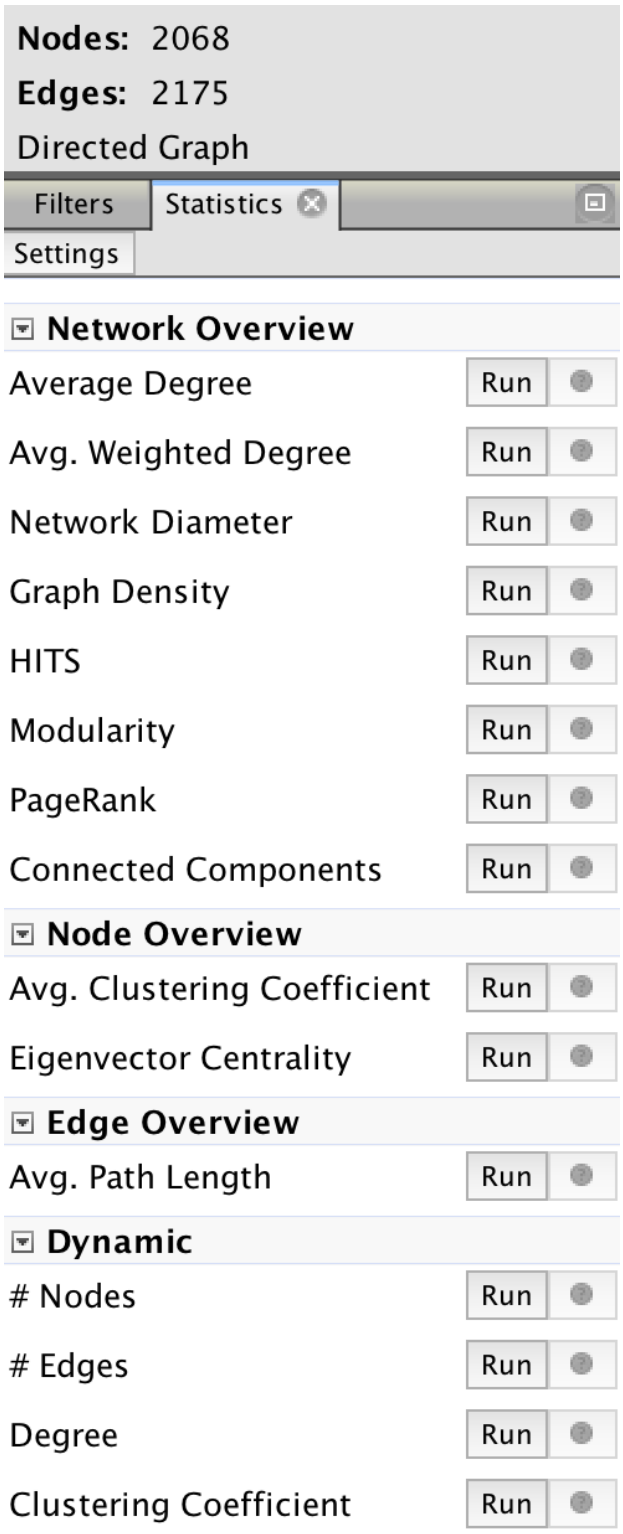


Visualization after Forced Atlas algorithm has finished executing.

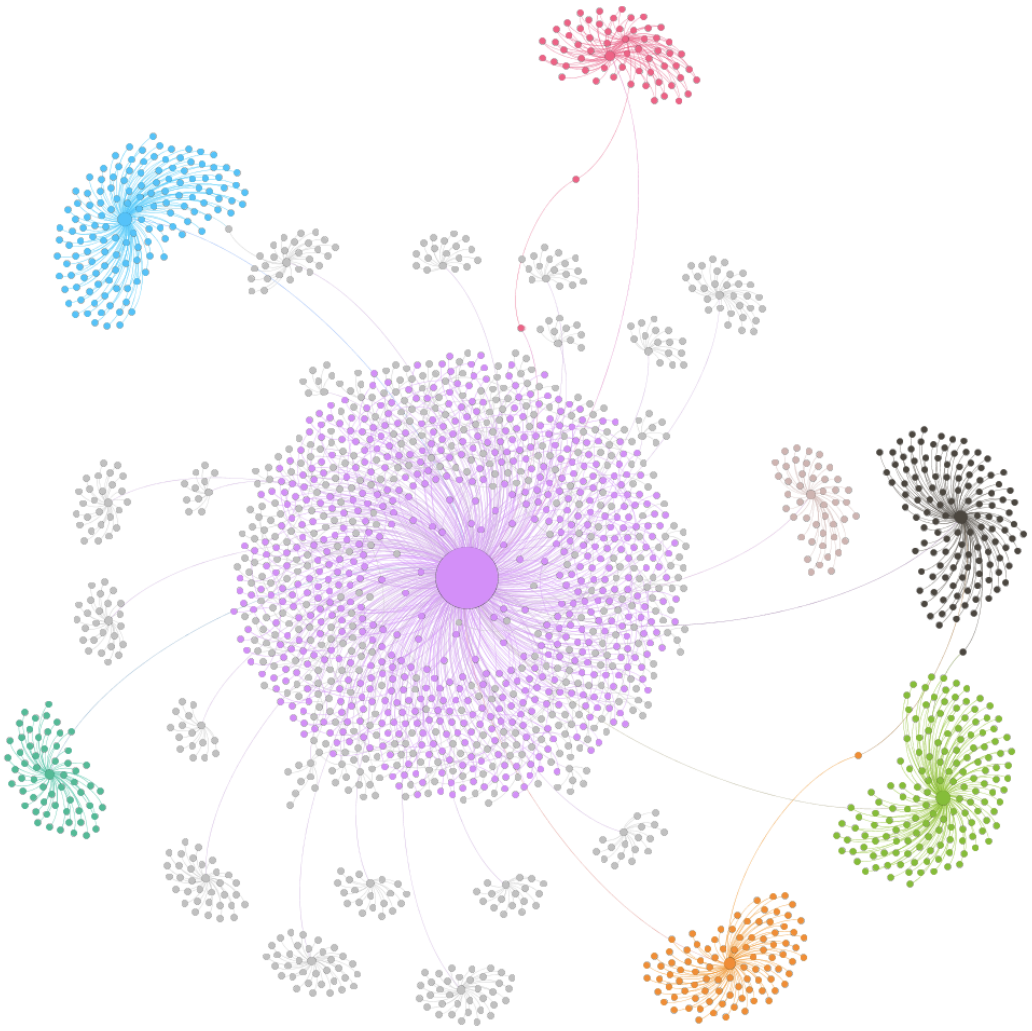


Then we apply settings to the nodes and select *Degree* to color the nodes.

Visualization after colorizing nodes according to their degrees.



The statistics of the graph are displayed before any calculation-operations are performed.

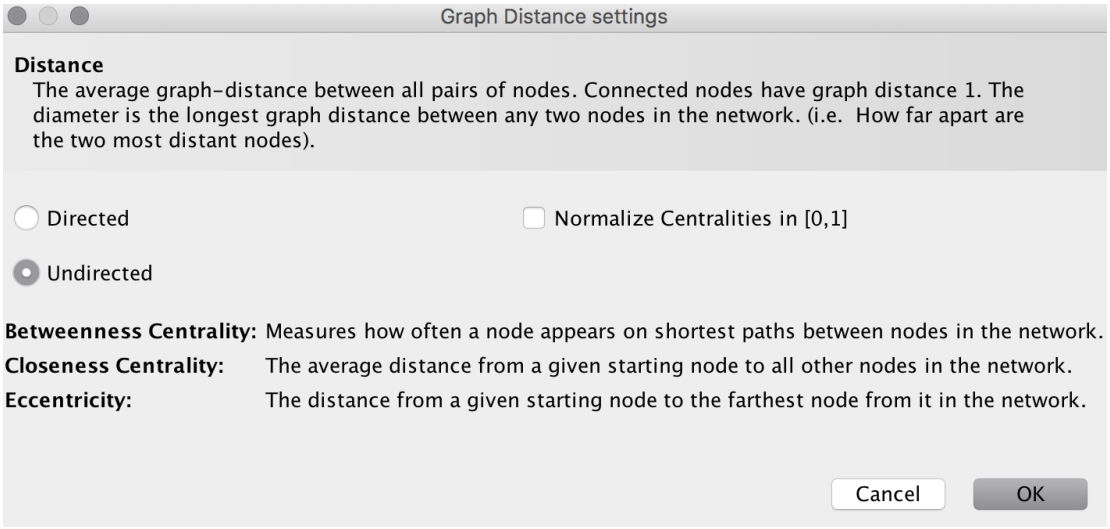


Force Atlas: final visualization[8]

---

8 This visualization was simulated after applying Noverlap algorithm, whose intermediate results are shown later in the report.

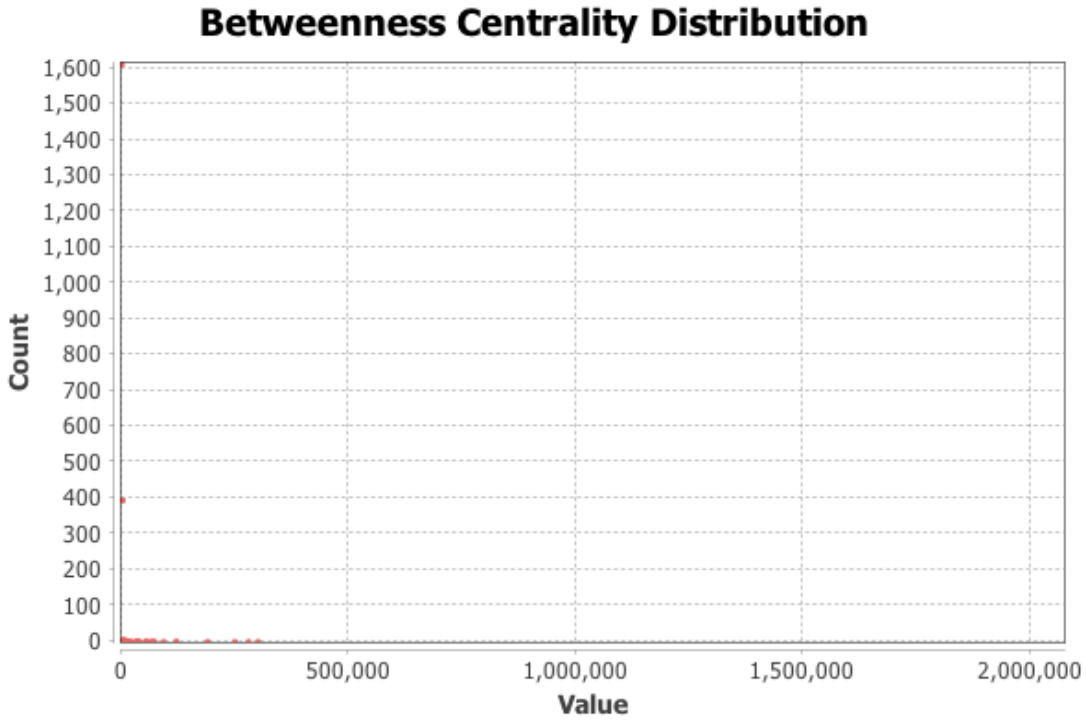
3. Graph Distance



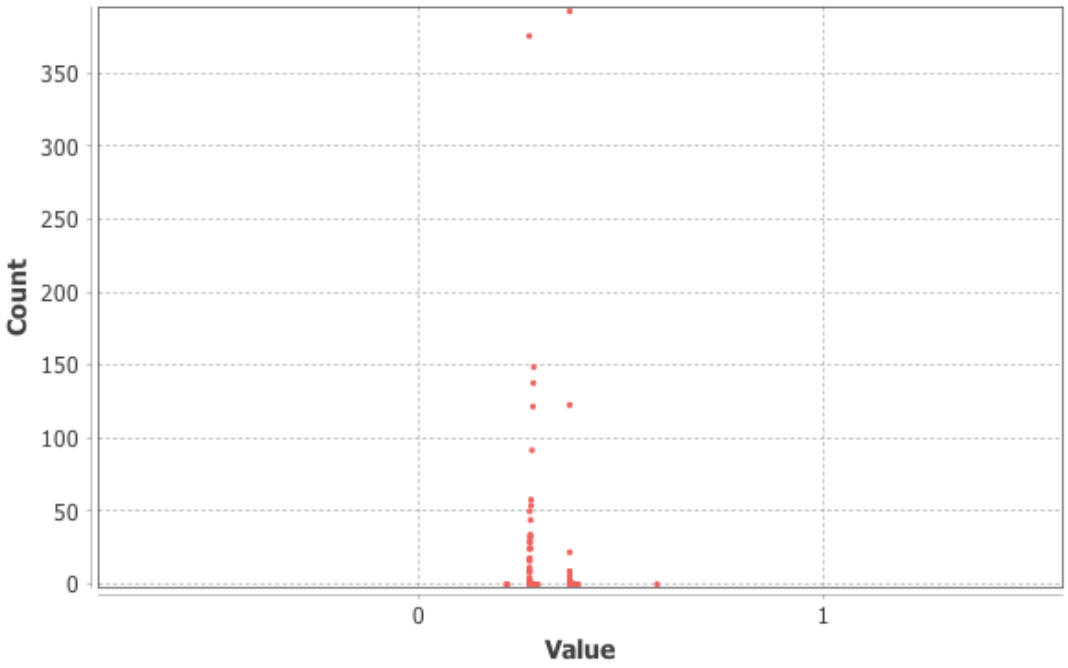
First, we calculate the graph-distance.

Parameters:

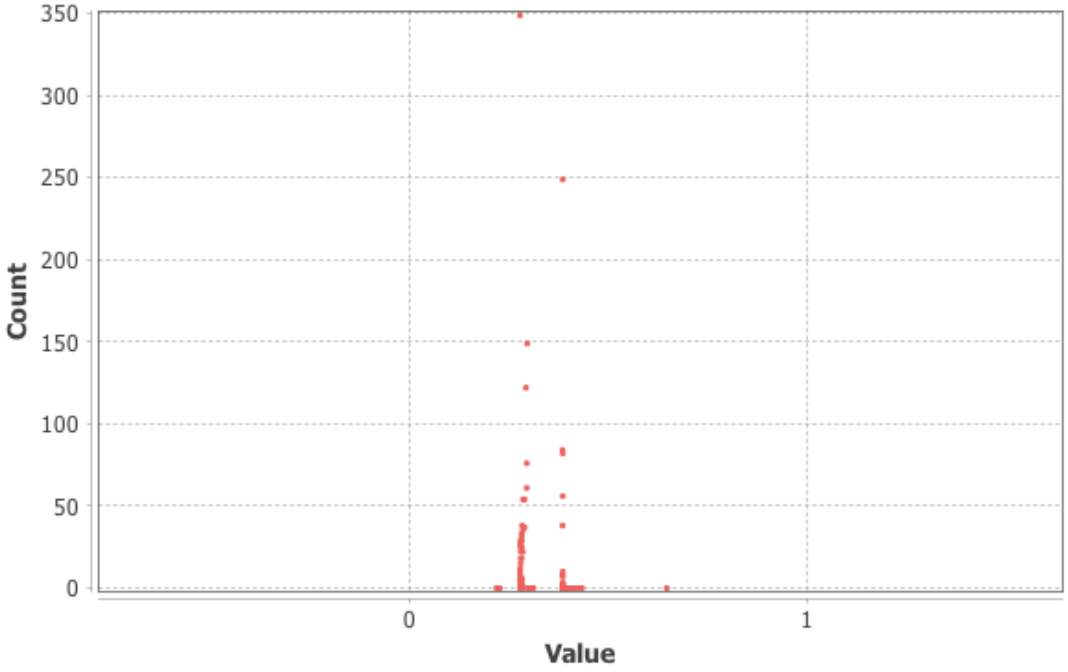
Network Interpretation: undirected

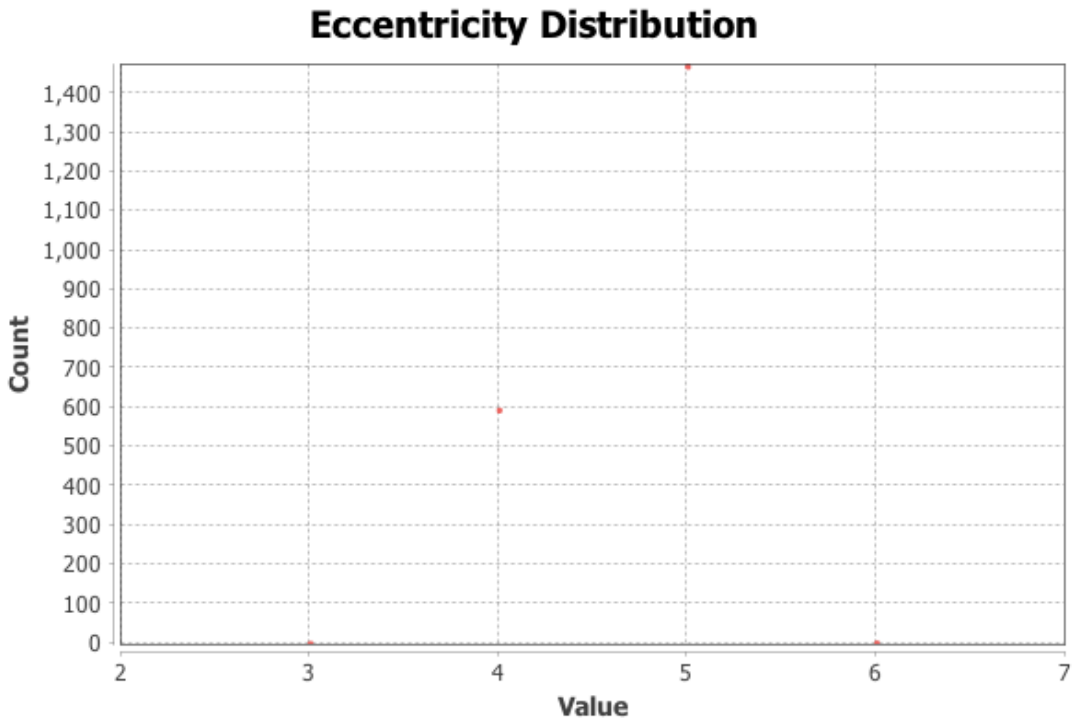


Closeness Centrality Distribution



Harmonic Closeness Centrality Distribution





Results:

Diameter: 6

Radius: 3

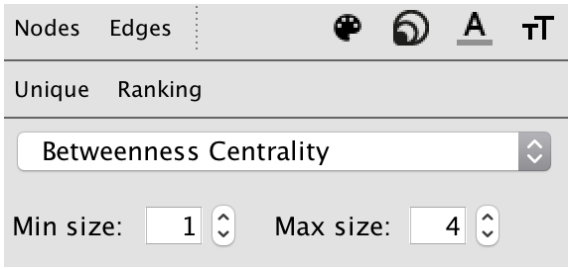
Average Path length: 3.3839341442713584

Algorithm:

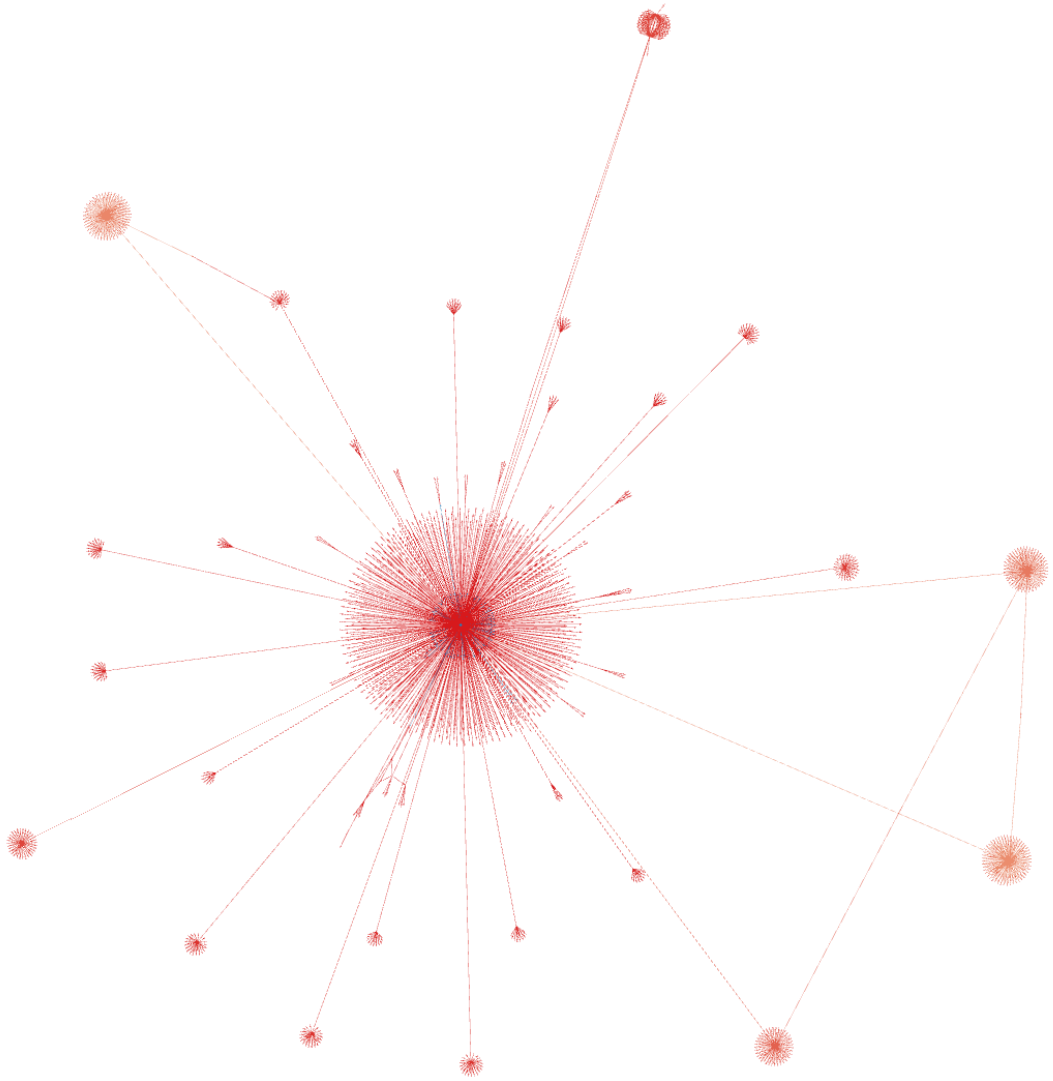
Ulrik Brandes, *A Faster Algorithm for Betweenness Centrality*, in Journal of

Mathematical Sociology 25(2):163-177, (2001)

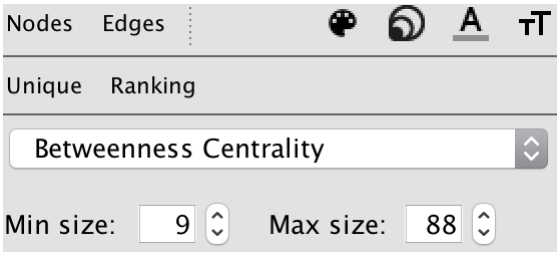
4. Betweenness Centrality



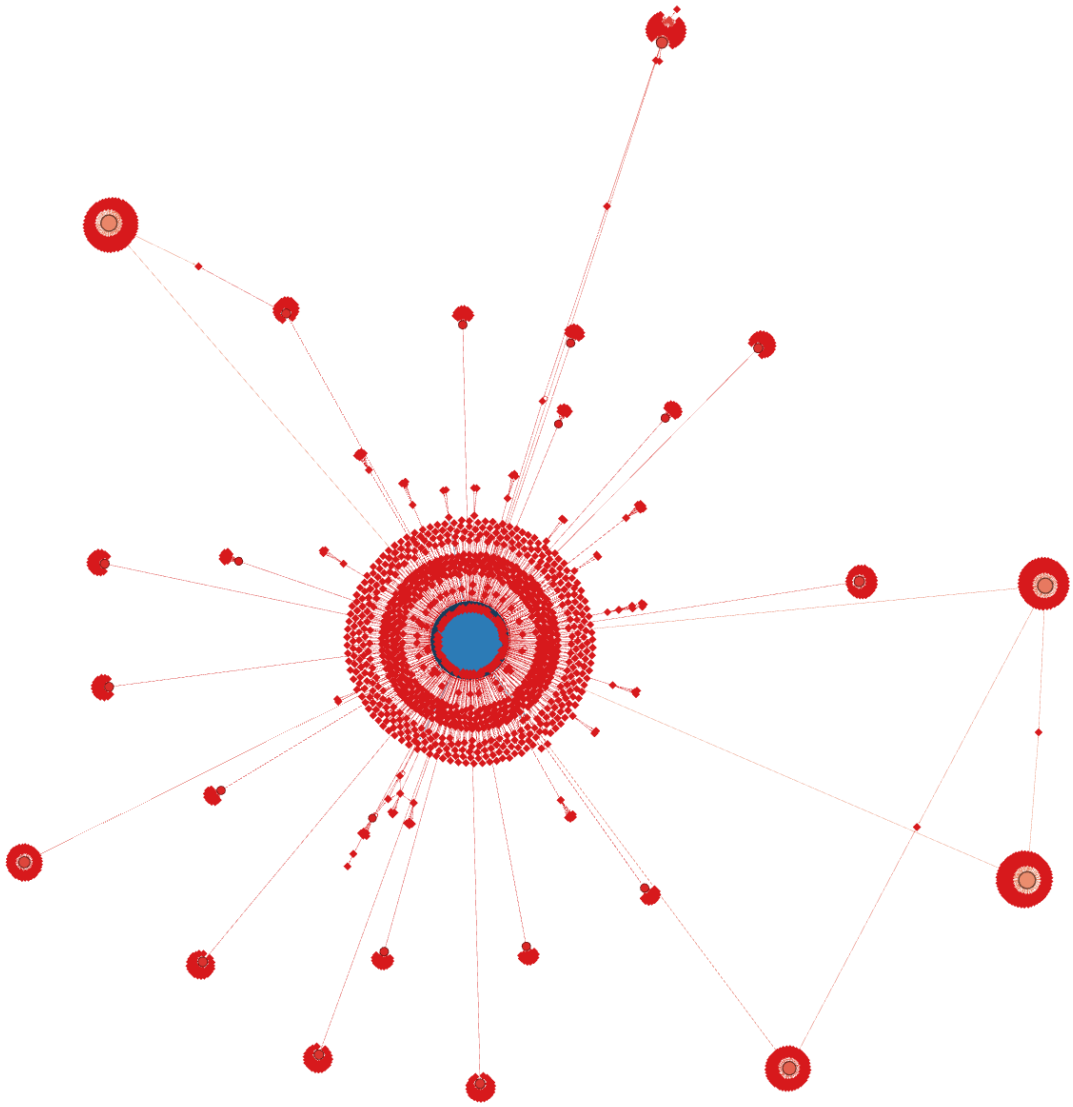
We select the node-size, ranking and then Betweenness Centrality with the above parameters.



Visualization of the dataset when min. size is 1 and maz. size is 4.

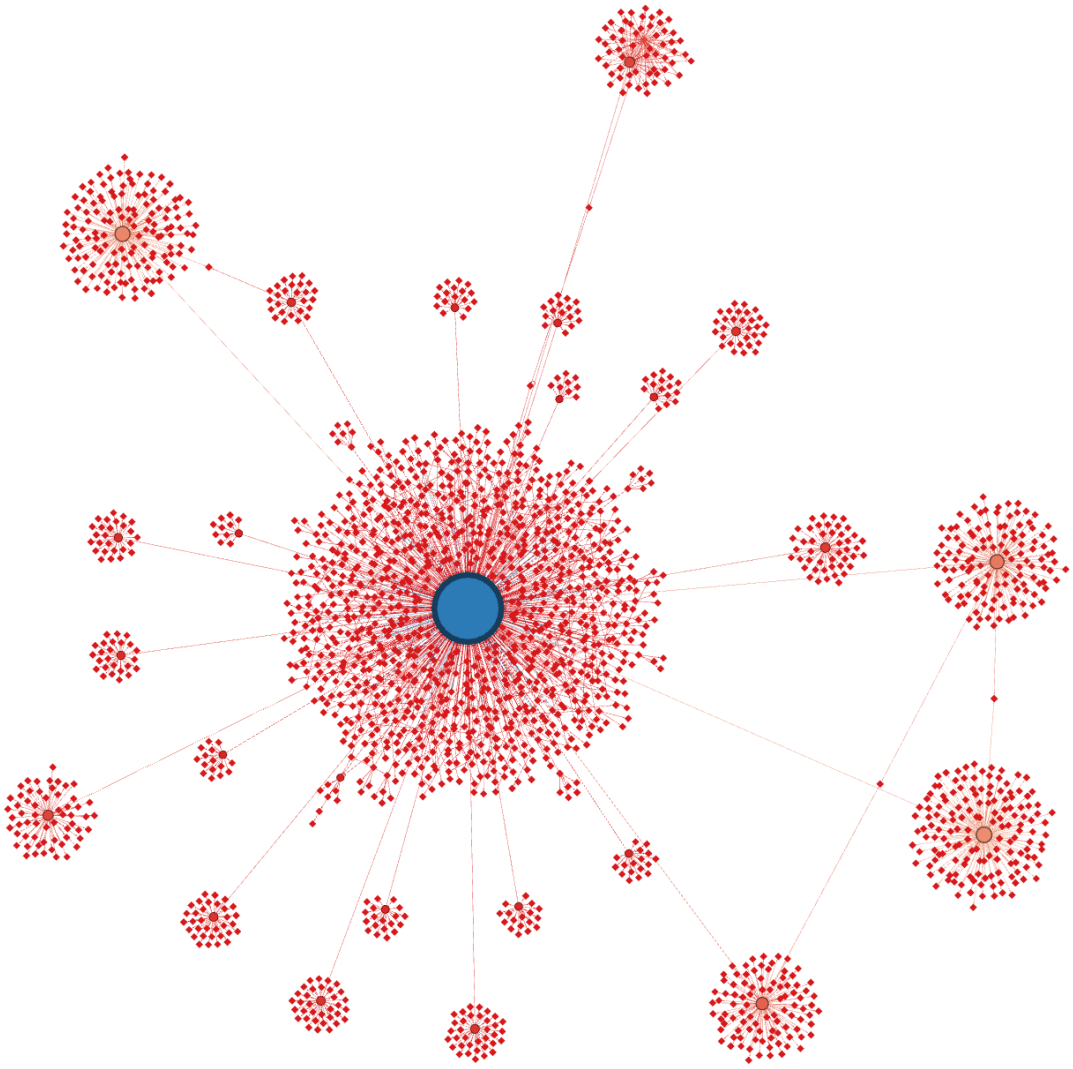


Then we repeat for min. size 9 and max. size 88.



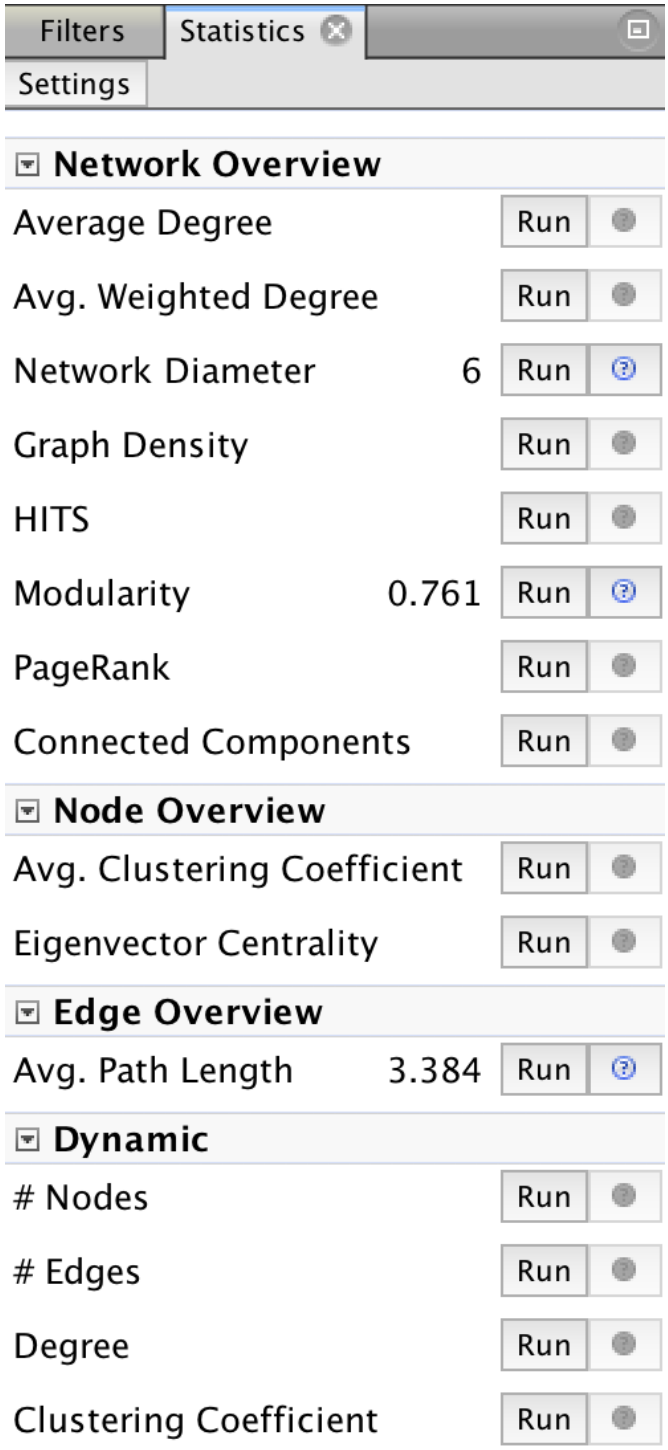
Visualization for above parameters for Betweenness Centrality.

We check adjust by sizes in the Layout panel.



We get the above visualization after adjusting for sizes.

5. Community detection/analysis



We run *Modularity* in the the Statistics panel and it's computed as 0.761

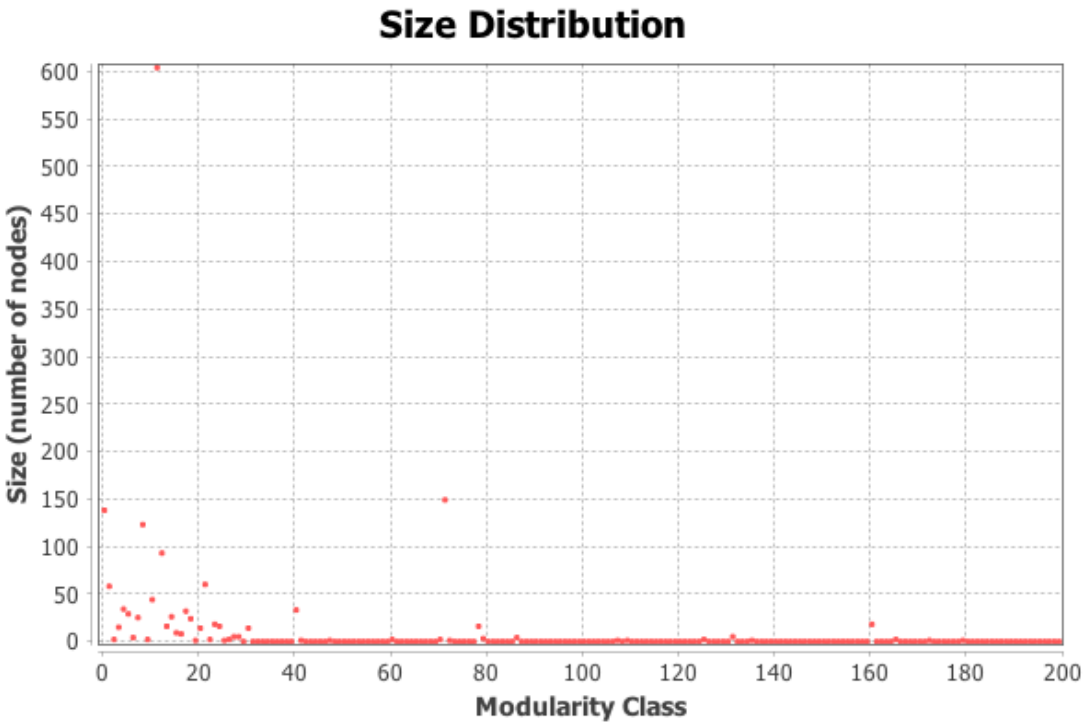
5.1. Modularity Report:

Parameters:

Randomize: On

Use edge weights: On

Resolution: 1.0



Results:

Modularity: 0.761

Modularity with resolution: 0.761

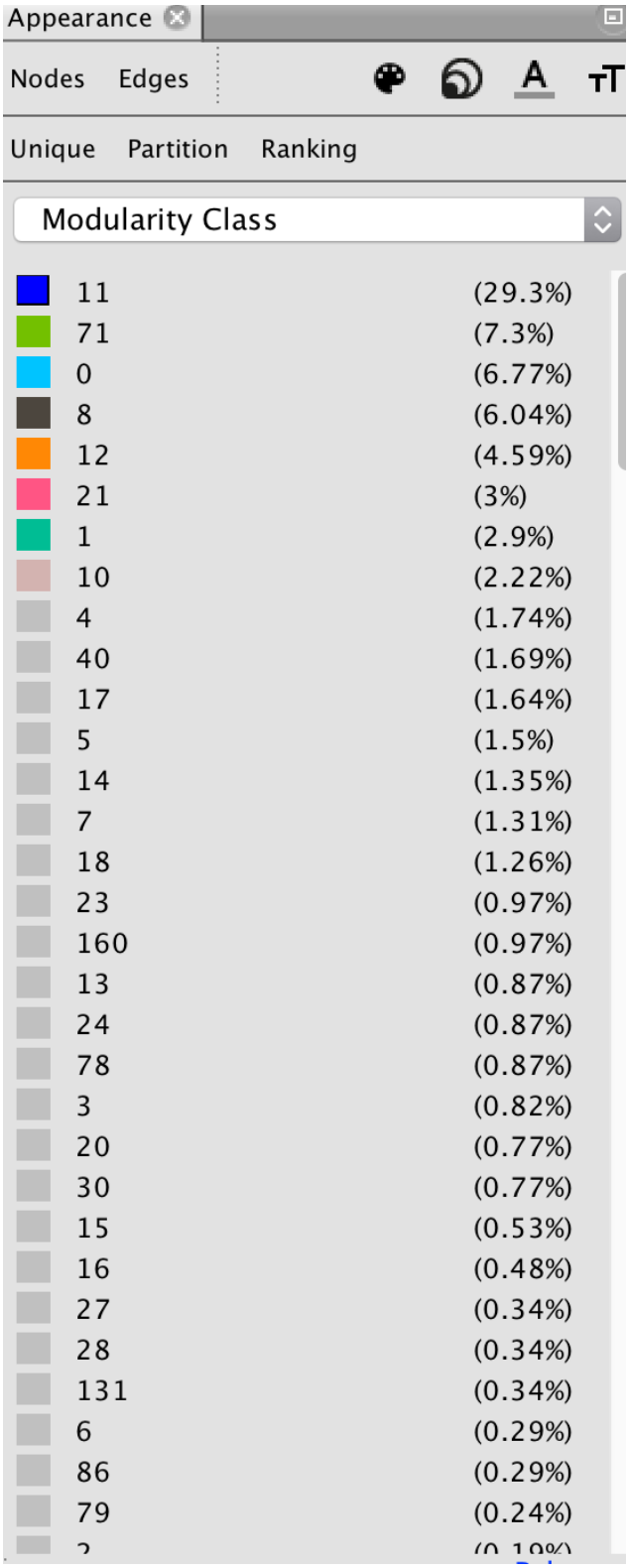
Number of Communities: 200

Algorithm:

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Etienne Lefebvre,
Fast unfolding of communities in large networks, in Journal of Statistical Mechanics: Theory and Experiment 2008 (10), P1000

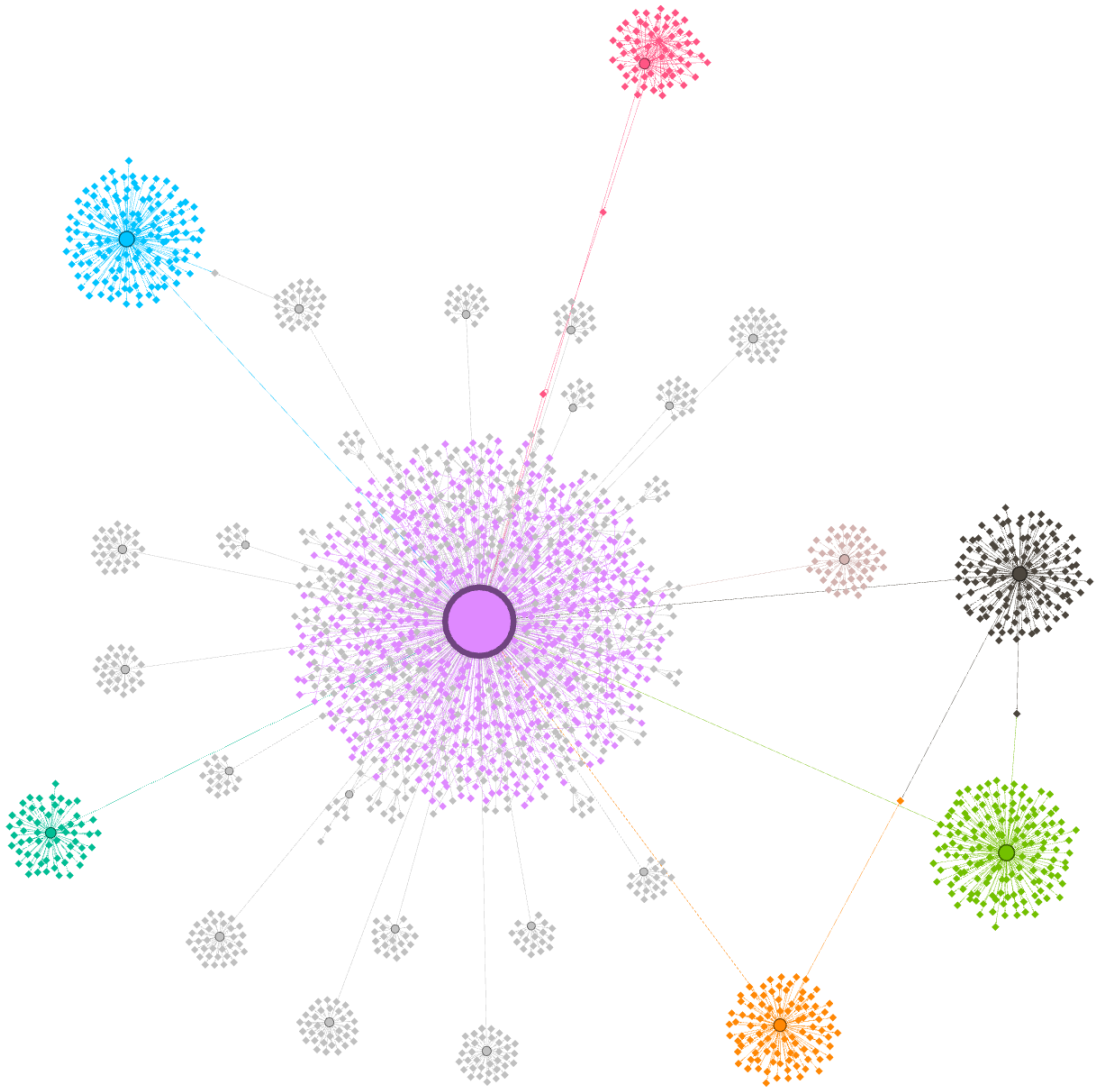
Resolution:

R. Lambiotte, J.-C. Delvenne, M. Barahona *Laplacian Dynamics and Multiscale*
Modular Structure in Networks 2009

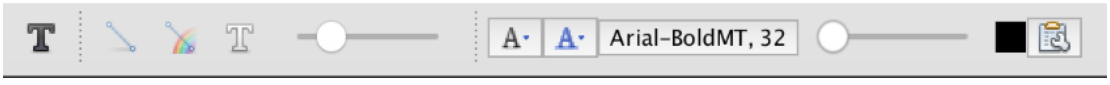


Then, in the Appearance panel, we select nodes and choose the color option

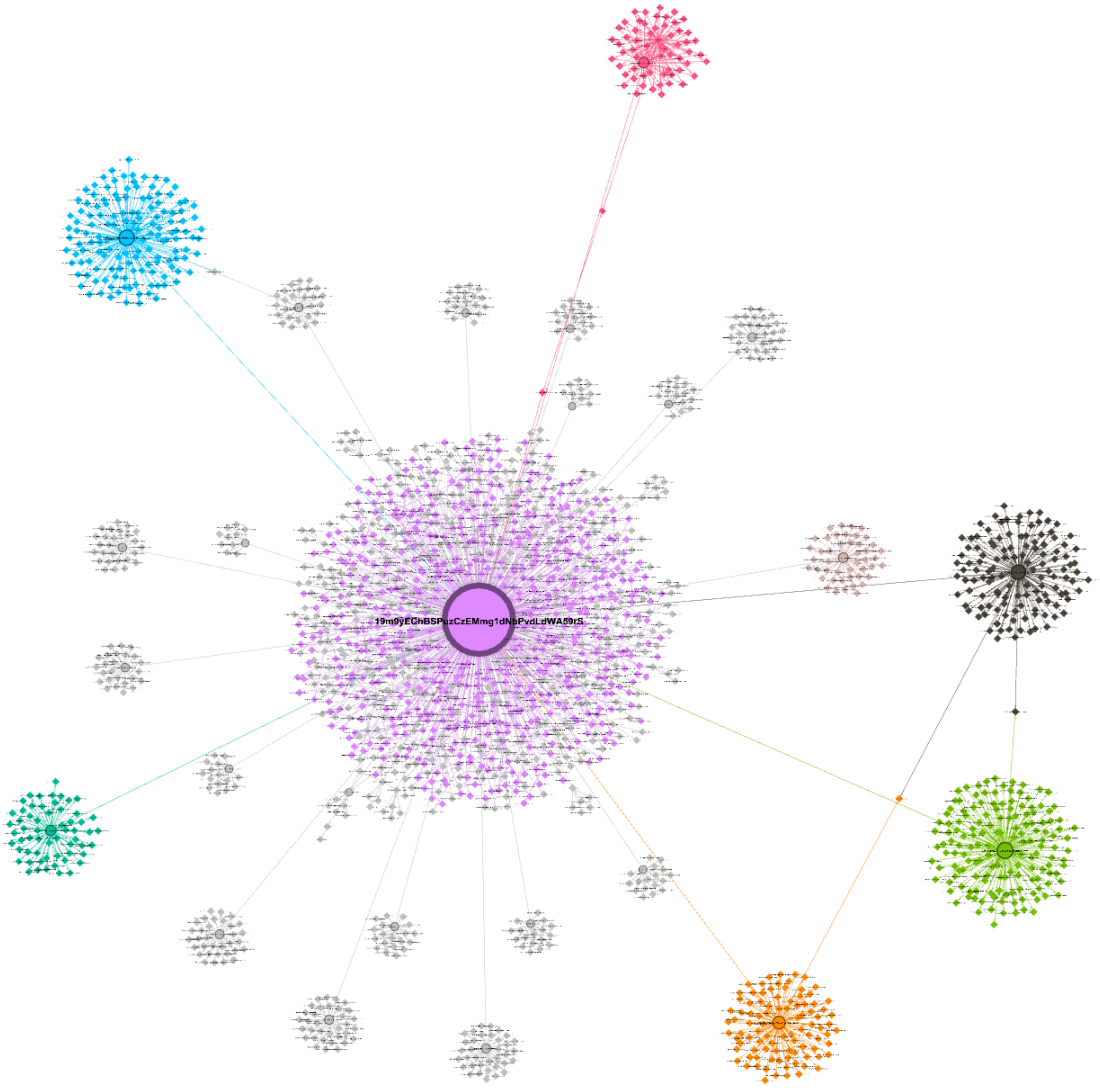
where we select Partition. Then we select the option *Modularity Class*.



This is the visualization we get for Betweenness Centrality.

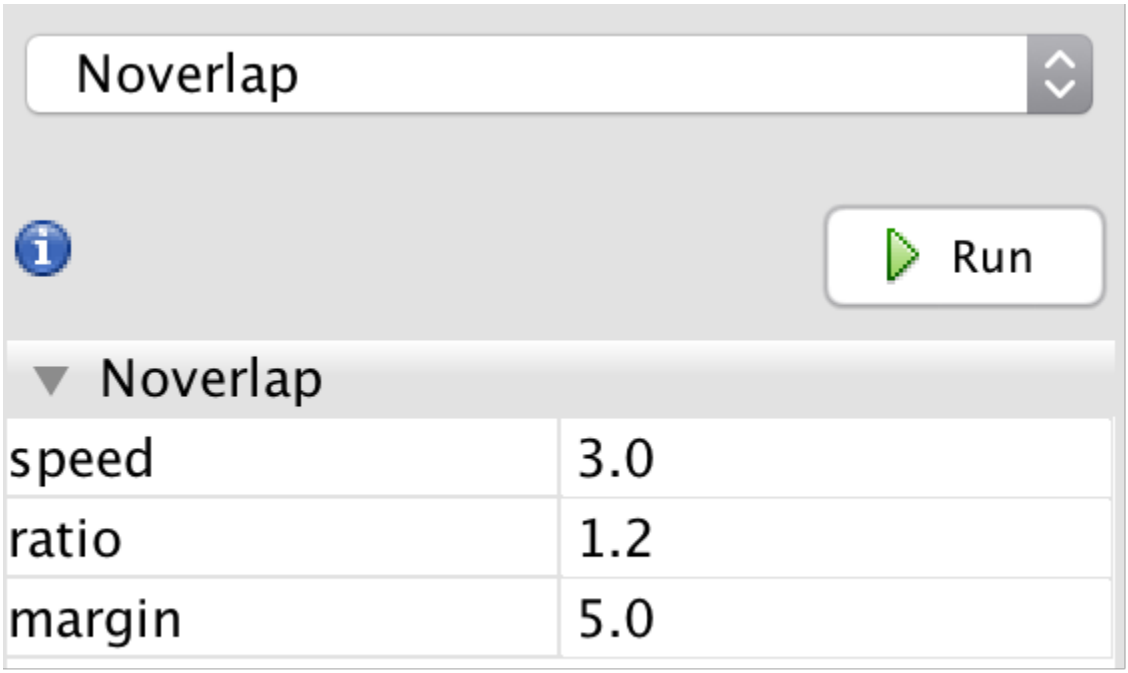


Then we scale the label-size to node-size.

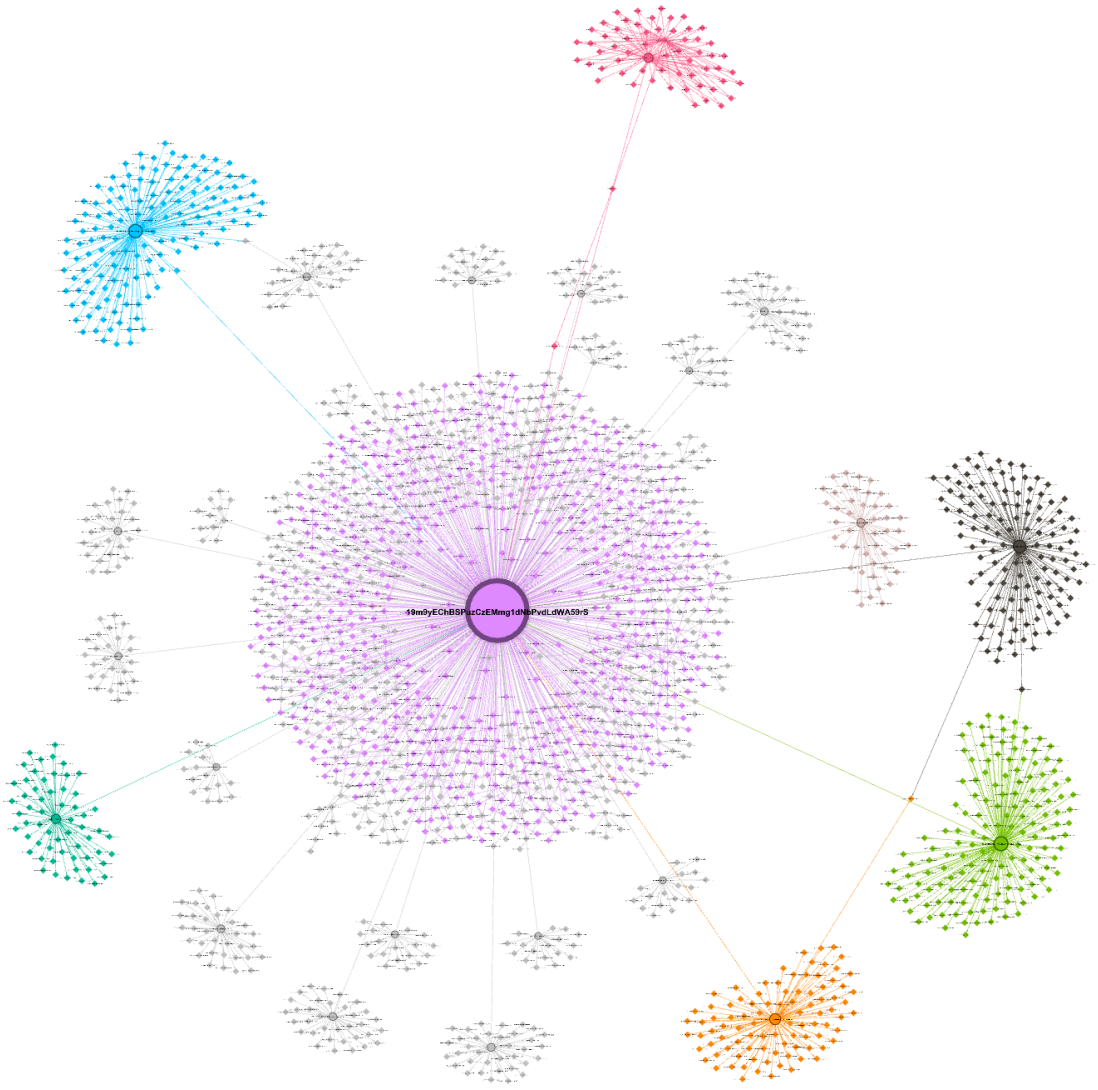


This is the visualization we get after scaling the label-size.

6. Noverlap



We run the Noverlap Layout algorithm.

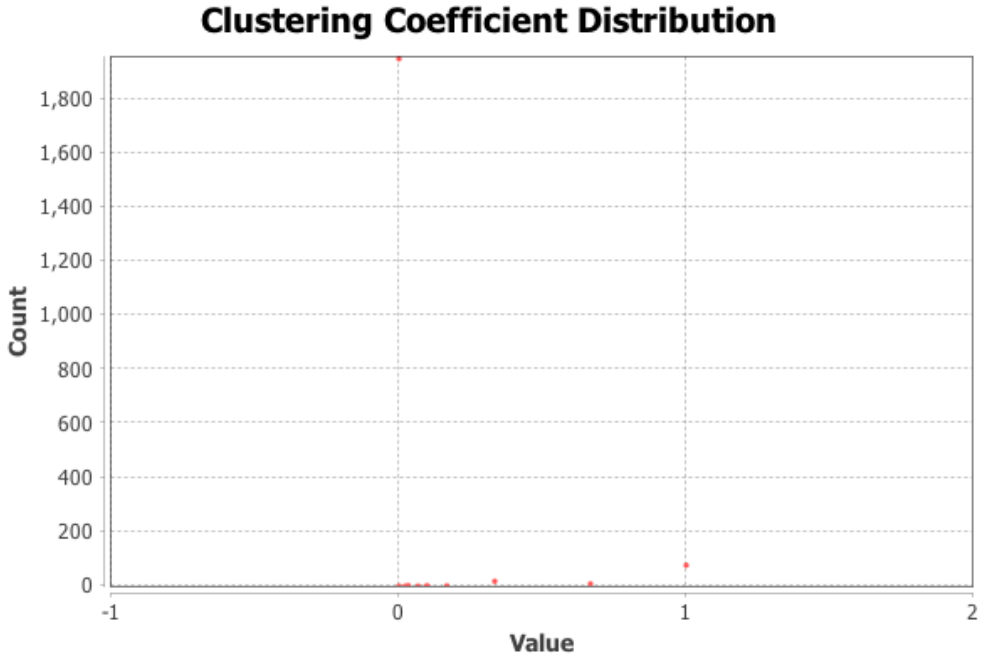


Visualization after applying Noverlap algorithm to the dataset.

7. Average Clustering Coefficient

Parameters:

Network Interpretation: undirected



Results:

Average Clustering Coefficient: 0.168

Total triangles: 89

The Average Clustering Coefficient is the mean value of individual
coefficients.

Algorithm:

Matthieu Latapy, Main-memory Triangle Computations for Very Large
(Sparse (Power-Law)) Graphs, in Theoretical Computer Science (TCS)
407 (1-3), pages 458-473, 2008

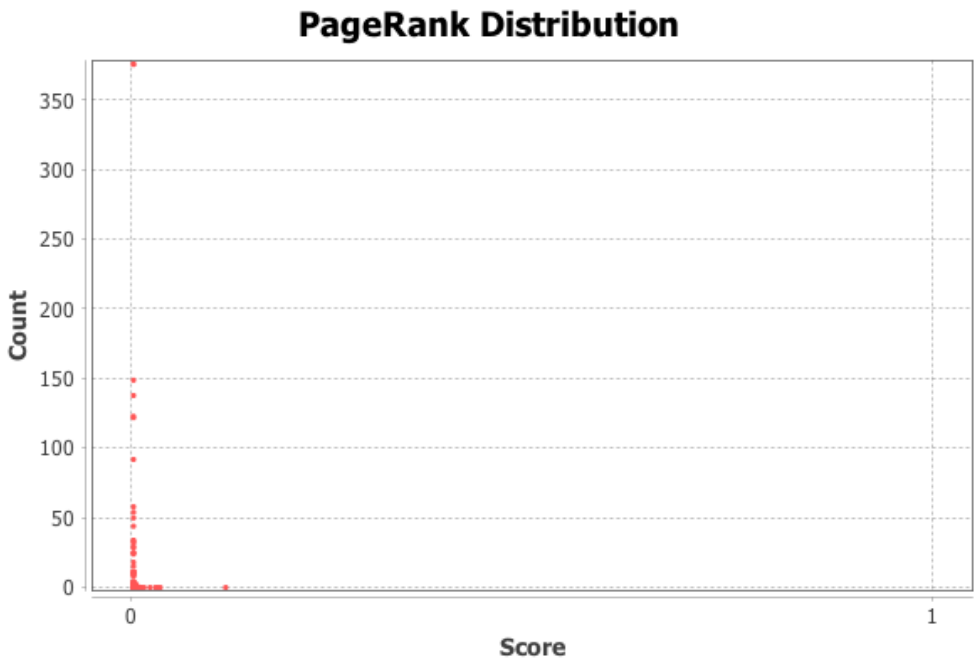
8. PageRank Distribution

Parameters:

Epsilon = 0.001

Probability = 0.85

Results:



Algorithm:

Sergey Brin, Lawrence Page, *The Anatomy of a Large-Scale Hypertextual*
Web Search Engine, in Proceedings of the seventh International
Conference on the World Wide Web (WWW 1998):107-117

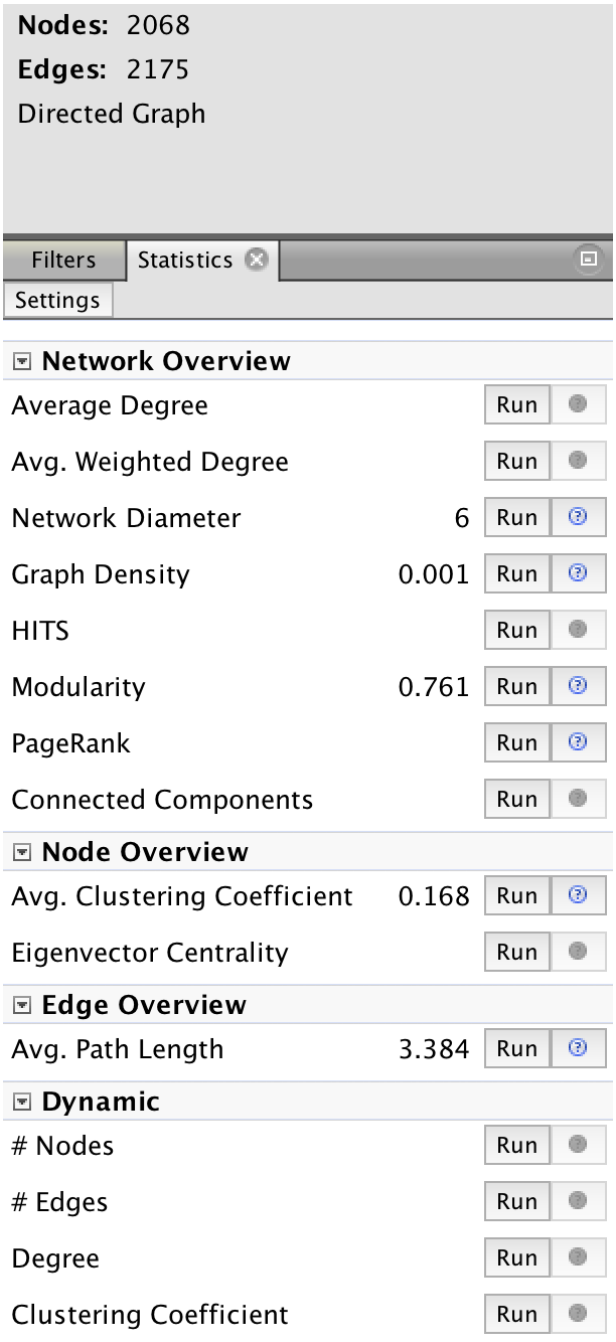
9. Graph Density

Parameters:

Network Interpretation: undirected

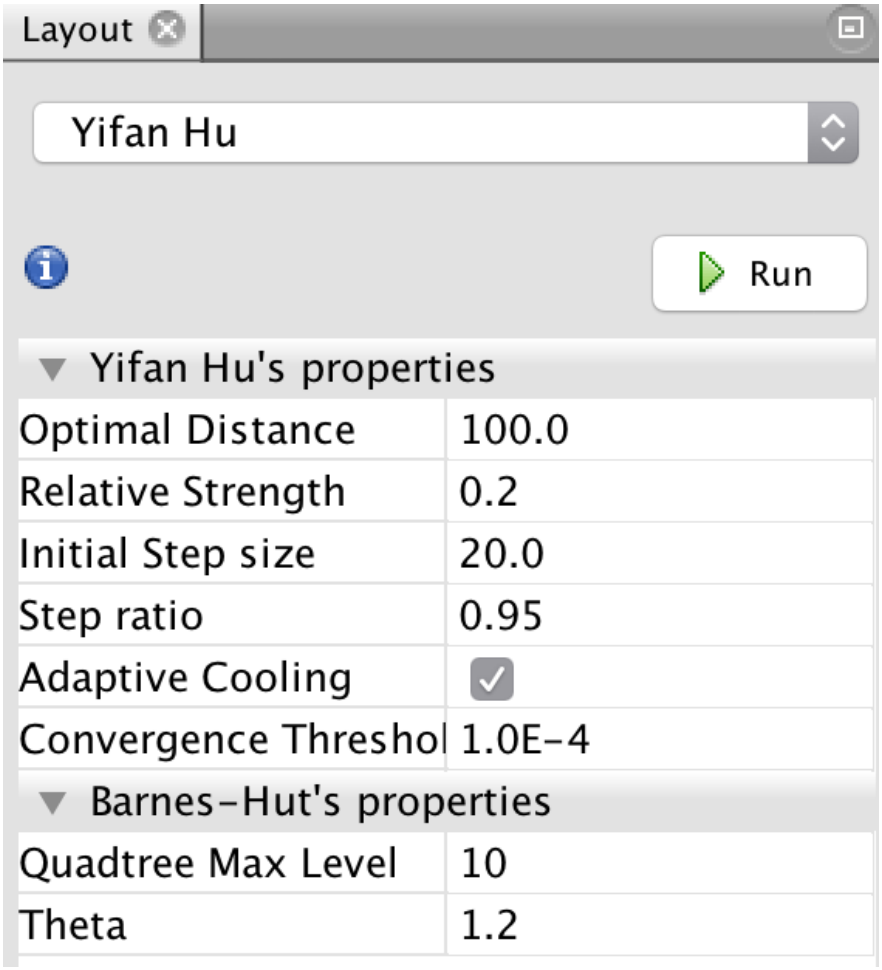
Results:

Density: 0.001

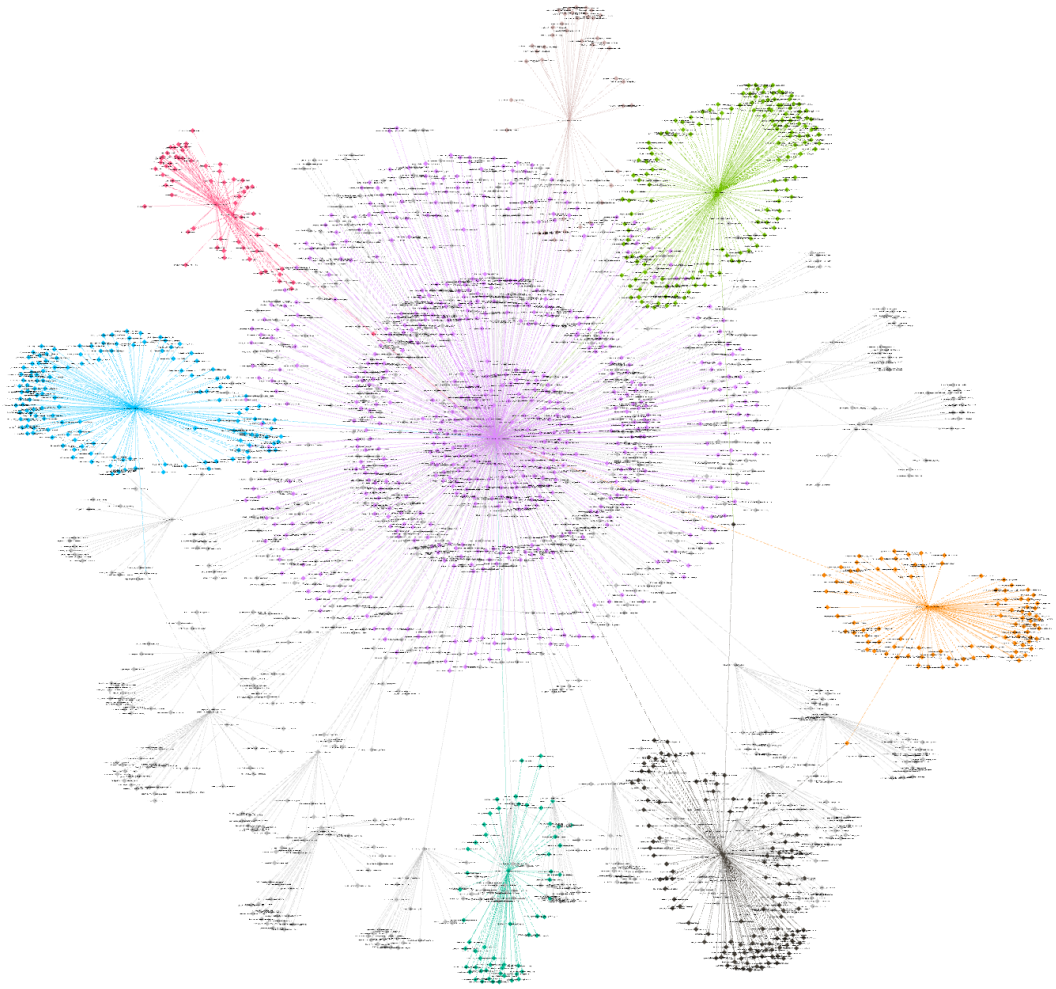


Summary of calculated metrics.

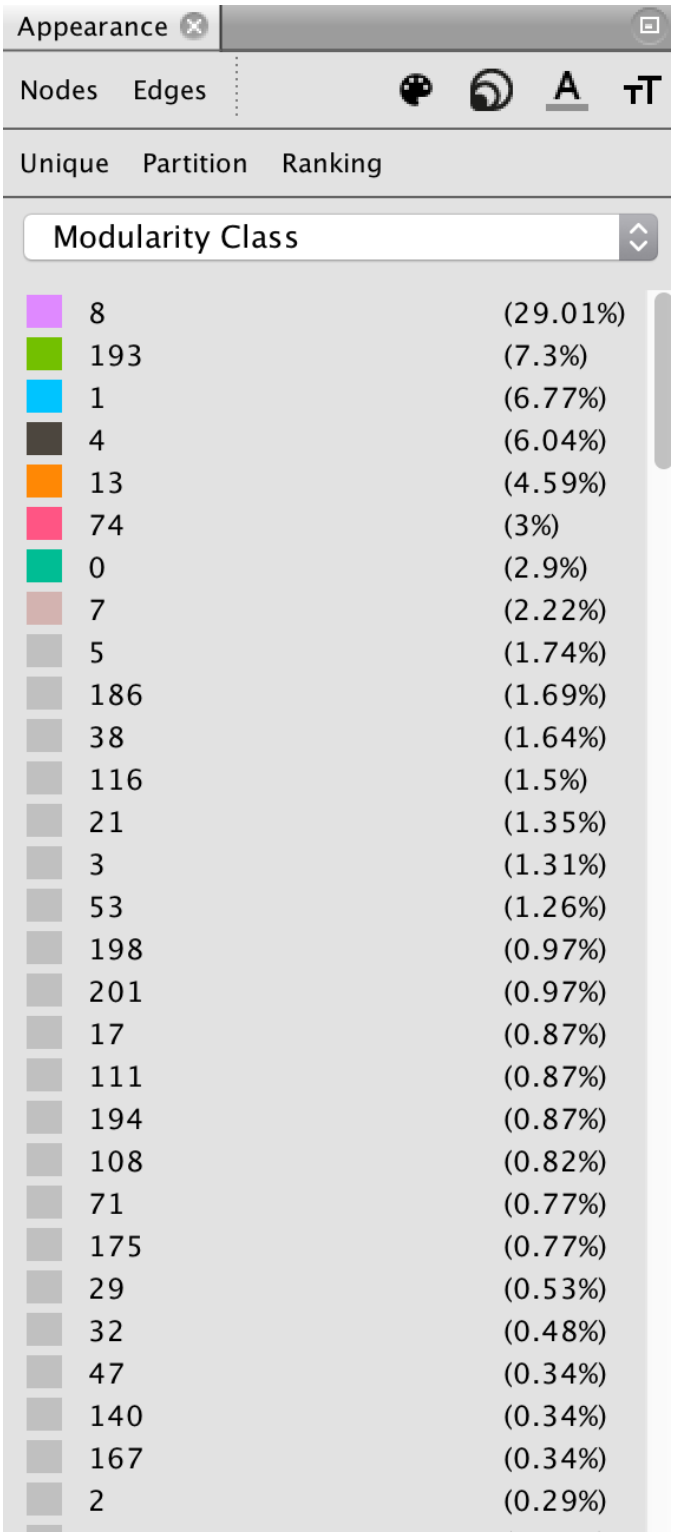
10. Yifan Hu



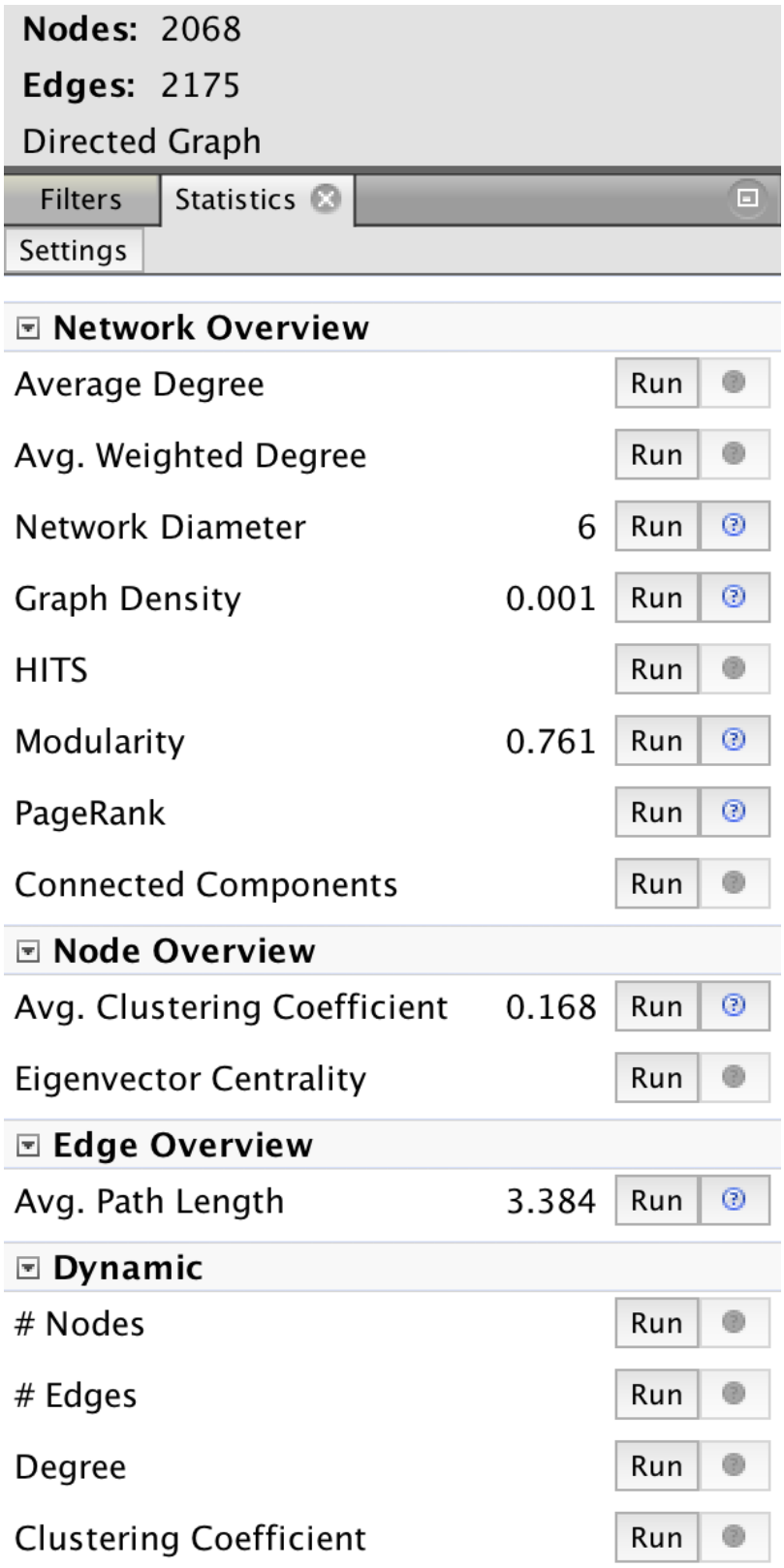
The above default properties are applied to Yifan Hu algorithm.



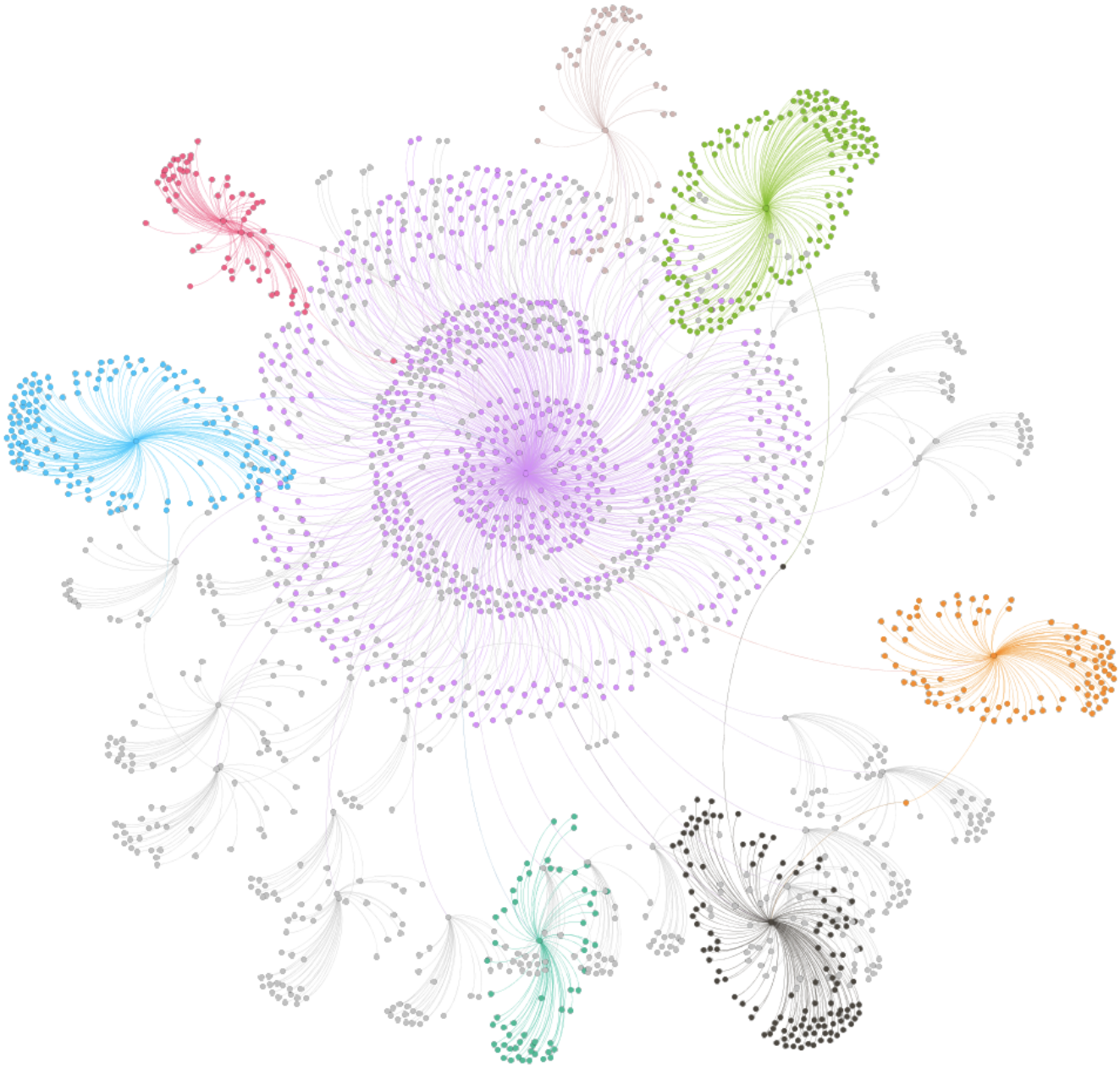
This is the visualization obtained after performing community analysis.



Modularity class: community-detection



Summary of Calculated of metrics for Yifan Hu.



Yifan Hu: final visualization

DISCUSSION AND CONCLUSION

We visualize the dataset using two layout algorithms; Force Atlas and Yifan Hu. The dataset used shows Bitcoin transactions for a unique Bitcoin address.

The only difference between the two implementations of the dataset is the layout algorithm used; viz. Force Atlas and Yifan Hu. The size of a node indicates the number of transactions. The color of a node ranges from red to yellow, to blue. The color red indicates that the degree of a node is low; whereas, the color blue indicates that a node has a high-degree.

An observation can be made immediately showing that Yifan Hu pushes nodes with low link count much more strongly to the periphery than Force Atlas does. Force Atlas' attraction distribution feature tries to differentiate the nodes but, pushing the low-degree nodes to the periphery while keeping the high-degree nodes in the center.

Both methods put some features of the graph to the forefront and the capacity for critical reading is as important as the willingness for "critical use" that does not gloss over the differences in tools used.

Hence, we can arrive at the conclusion that Yifan Hu is a better visualization layout algorithm as it can distinguish between disparate Bitcoin transaction sources, isolating the ones with higher transaction-volume towards the center.