Final Project Report: Entity Identification in Bitcoin

Afshan Anam Saeed (21-741-954), Migle Kasetaite (21-733-779), Nidhi Agarwal (21-717-673), Sanjana Warambhey (21-742-531)

Network Science

Faculty of Business, Economics and Informatics

17 December 2021

ABSTRACT

Bitcoin is a decentralized digital currency and electronic payment system that uses peer to peer technology to operate. One of its most essential characteristics is that it provides its users with a high level of anonymity which has eventually created regulatory concerns due to illegitimate activities that might be carried out through such networks. Despite the pseudo anonymity of Bitcoins, there exists measures of user identification. In this report, we will be using heuristics to generate an Address Correspondence Network, on which we will perform clustering using the Community Detection Algorithms. Based on these communities, we will identify addresses belonging to the same entity and will validate our results using the Ground Truth Data.

1 INTRODUCTION

Bitcoin has become one of the most favorable cryptocurrency being used in modern times. It has gained popularity because of its proven properties of maintaining pseudo anonymity of users, decentralized nature, speed, and effectiveness against counterfeits. Due to these advantages, several illegitimate activities are being carried out over the platform, and thus, there is an increasing concern amongst governmental institutions regarding the difficulty in identifying such users.

Although Bitcoin is anonymous, the addresses and transactions are public. It is difficult to link one person to an address, especially since anybody may generate an almost infinite number of addresses. However, the majority of users acquire bitcoin through large exchanges, which means that a firm has all of one's personal information (name, address, bank account information) and knows where one transferred their Bitcoin. If the originating address of the user is known, then the other addresses (input, output and change) related to user transactions can be possibly retrieved. In our report, we will be implementing one such way to deanonymize the user by clustering together the addresses belonging to the same entities. Our methodology comprises of applying different heuristics (identifying techniques) to determine if there is a connection between two addresses to form 'Address Correspondence Networks'. Our network will comprise of user addresses as nodes, and two nodes are linked if there are heuristics determining the two belonging to the same user. The project takes into account heuristics paired with the baseline multi input heuristic ($h0$) to build the said Address Correspondence Network.

We show that the Address Correspondence Networks have a strong community structure, and that general-purpose clustering algorithms are effective for analyzing them. This approach should be helpful in improving the detection of entities in the bitcoin network. This project is based on [Fischer et al. 2021], and has been extended to view results on its weighted and paired heuristics version. Our report is structured into 4 main sections, namely, Theory, Methodology, Results and the Conclusion. In the theory we describe the basic Bitcoin network, Methodology speaks about the process and the Results and Conclusion gives the output of our analysis.

2 THEORY

2.1 The Bitcoin Blockchain

The Bitcoin is the currency of a decentralized digital payment system that is independent of any third party verification. This payment system is based upon an immutable public ledger, called Blockchain, that allows users to pseudo anonymously exchange digital currencies. The process of exchanging these currencies digitally is called Transactions while the users conducting these transactions are called entities. While making a transaction, an entity operates at the input level and sends currencies to an output level. Here, a user defines an input and an output address, which act like the identifiers of any in-out transaction. These addresses have a property of being nameless and unique, and are often not associated with any real life addresses due to the blockchain's publicly accessible nature. These factors make addresses a key player in maintaining the blockchain's pseudo anonymity.

Every transaction is composed of a set of input and output addresses, and information about the amount of Bitcoin that has been transferred from an input to an output address. Generally, the sum of bitcoins at the input equals the sum of the bitcoins being transferred to the output plus the transaction fees. If the entire sum of bitcoin currency is not being spent on the output, the change is directed towards another address called the change address. These change addresses are again controlled by the input entity. If any output address does not proceed with another transaction, then this address is referred to as UTXOs (Unspent Transaction Outputs).

Within the Bitcoin network, every transaction gets replicated to multiple nodes. As entities conduct transactions from one node to another within the network, certain minor nodes can be incentivised for validating their transactions, grouping them into blocks. These blocks get sequentially appended to the blockchain. The number of blocks before a certain block is called its Block Height and the minimum block height specified by the entity before transaction is called the transaction's locktime. The blockchain must therefore reach a minimum locktime number of blocks to validate the transaction.

2.2 Address Clustering

As specified in the previous section, any entity is liable to generate many addresses for conducting transactions. These addresses are the closest to breaking the pseudo anonymity of the user. In the address clustering mechanism, the goal is to cluster these addresses together to identify whether they may have come from the same entity. To do so, there exist multiple heuristics that help in such clustering. Referencing the paper [Fischer et al. 2021], the seven heuristics are:

1. Multi-input. All input addresses of a transaction are assumed to be controlled by the same entity.
2. Change address type. If all input addresses of a transaction are of one address type (e.g. P2PKH or P2SH), the potential change addresses are of the same type.
3. Change address behavior. Since entities are advised to generate a new address for receiving change, an output address receiving Bitcoins for the first time may be a change address.
4. Change locktime. If a transaction's locktime is specified, outputs spent in different transactions on the same block as the specified locktime may be change addresses. Intuitively, this is because the entity initiating the transaction also knows its locktime.
5. Optimal change. If an output is smaller than any of the transaction inputs, it is likely a change address.
6. Peeling chain. In a peeling chain, a single address with a relatively large amount of Bitcoins begins by transferring a small amount of Bitcoins to an output address, with the rest being allocated to a one-time change address. This process repeats several times until the larger amount is reduced, meaning that addresses continuing the chain are potential change addresses.
7. Power of 10. This heuristic assumes that the sum of deliberately transferred Bitcoins in a transaction is a power of 10. If such an output is present, the other outputs may be change addresses.

Multiple studies have used the Multi input heuristic for identifying addresses of the same entity due to address reuse, high centrality of address clusters and incremental cluster growth. In our case, we will be using a combination of Multi input heuristic with all the other heuristics mentioned above. Our approach is to indicate the clustering as an undirected, weighted graph, where the nodes are addresses and edges indicate the number of times the heuristics identify the pair to belong to the same entity. Clustering for the above is carried out using the Label Propagation algorithm and the effectiveness of our result is understood by comparing our results to the ground truth data.

3 METHODOLOGY

Considering the addresses involved in transactions to be the nodes of the Address Correspondence Network, the edges are representative of the heuristics that identify them to belong to the same entity. We will be using the blockchain transaction details from the years 2012 and 2014 for sampling and building these Address Correspondence Graphs. There are also certain data sets that are designed such that the controlling entity of addresses are already known. These data sets are referred to as the ground truth data sets. Our project is based upon these ground truth data sets, wherein, they will help in the sampling of our transactions and also evaluate the quality of our clustering mechanisms.

Our methodology section is divided into 4 sections, where we will first speak about our methods of sampling data. We will then proceed with the Address Correspondence Network construction, clustering of our network, and finally we will describe the metrics to analyze the quality of our clustering.

3.1 Sampling

Our two data sets contain millions of rows detailing transactions carried out in the defined time frame. It contains address information of the input and output nodes and their heuristics. Since operating on a data set with such number of values is not computationally feasible, it is important for us to take sampling of these data sets for moving forward with our analysis. Therefore, our first step is to randomly select a subset of the addresses from the provided ground truth data set. We label this first sample address data set as A_0^S. Next, we take into account the transactions that have been carried out using these addresses as either inputs or outputs. This set of transactions is labelled as T_0^S.

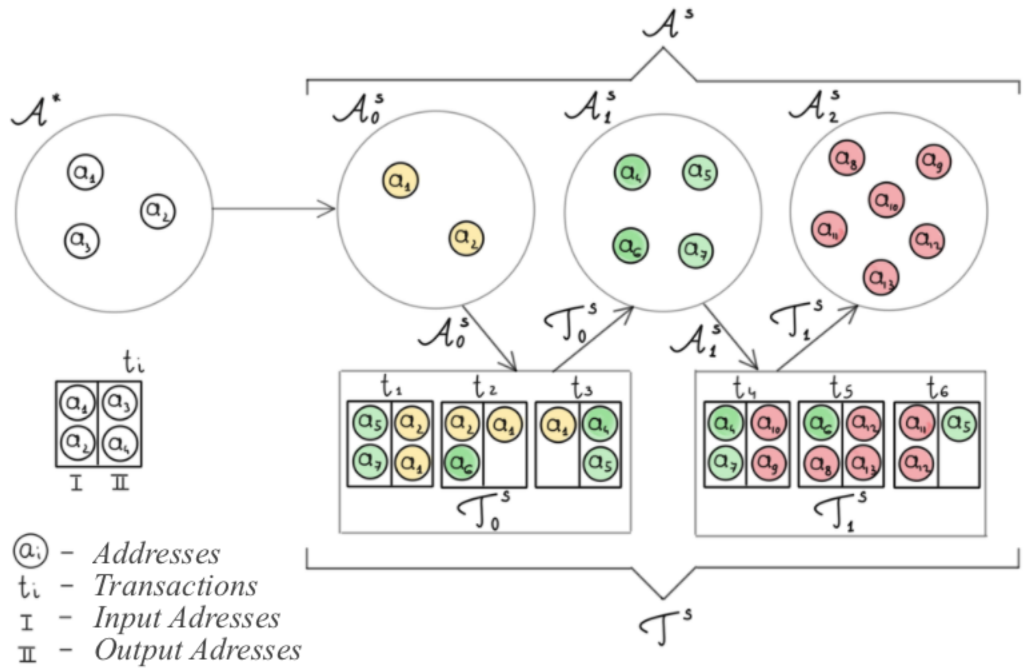

Fig. 1. The transaction sampling process.

Within this transaction sampling, it is evident that there are certain addresses in T_0^S, either at the inputs or outputs, that are not present in the sample set A_0^S. These new addresses are those which interacted with the ones in A_0^S during transactions. These addresses form a new sample set, which we will label as A_1^S. Repeating the process mentioned above, we will now make another transaction set,T_1^S, involving the components of A_1^S. It is important to note that this set of transactions does not contain the transactions that have been mentioned in the previous set T_0^S.

We follow the similar approach as written above for building another set of addresses that are mentioned in the transactions of T_1^S but are not a part of the address sample sets A_0^S and A_1^S. This new address sample set is called A_2^S. Thus, for our analysis, we have a transaction sample $T^S = T_0^S \cup T_1^S$ and an address sample $A^S = A_0^S \cup A_1^S \cup A_2^S$. It is to carefully note that the basis of our sampling has been around the ground truth data set, which also contains all the sample addresses. This makes it easy for us to compare the final outcome of our results with the ground truth.

3.2 Address Correspondence Network Construction

The Address Correspondence Network is one which defines the existence of relationships between addresses based on how often they get detected by the heuristics. The addresses make up the nodes and the edges indicate the presence of a relationship detecting heuristic. The more the number of times any heuristic detected any relation between two addresses, the greater is the weight of that connection. Thus, the network we are going to be constructing in this section will have the properties of an undirected, weighted network.

This information about the existence of heuristics has been provided to us in our data, where for every transaction defined by an input and output node, a numeral count of the number of times any heuristic identified two addresses to belong to the same entity has been given. Unlike the procedures used in [Fischer et al. 2021], our goal is concerned with using pairs of heuristics to define weights of linkages. Thus, we take iterative sums of the number of times heuristic $h0$ (multi input) detected relations with the number of times the other heuristic detected relationships. We finally use the values of these sums to construct seven (sole $h0$ included) Address Correspondence Networks, each corresponding to a pair of heuristics.

3.3 Address Correspondence Network Clustering

Having obtained our Address Correspondence Network, we will now proceed with the clustering of addresses in this network. Clustering of nodes for our analysis is performed in order to find areas of similarity amongst different addresses to identify them as belonging to a similar user. Our approach here is to identify these communities within the Address Correspondence Network using Community Detection Algorithms like the Label Propagation Algorithm (LPA) and pass these clusters for comparison with the ground truth data.

The Label Propagation Algorithm, in operation, firstly initializes each node with a unique label. It denotes in it the part of the cluster it belongs to, which here is the entity it belongs to. Mostly, this initialization is done randomly, after which random nodes are revisited to assign them labels of majority of their neighbors. This is repeated until all the nodes in the network are assigned a label. In our project, we have initialized part of the nodes using random entity labels from the ground truth data set, addresses of which are also contained in our sampling. We label these sample of addresses as $A^* \subset A^S$ and the labelling entities as e^*.

The choice now lies in the proportion of nodes that we wish to initialize in the LPA. We start with a minimum number and proceed further by varying the proportion by a factor such that p lies in the range 0 and 0.4. We run our algorithm for every exceeding proportion of the labelled nodes to obtain a disjoint set of clusters for each, such that $\cup_{i=1}^{(k)} C^{(i)} = A^S$. We run the entire process mentioned earlier in this section several times by making random choices of entity labels each time so that our results are not based upon a single random selection. We finally obtain an output that is unbiased and generalized and which can account for the valid comparison that we have to make with our ground truth data.

3.4 Cluster Quality Analysis

Post obtaining the set of clusters and the entity sizes, we move on to define the metrics that we will use to compare the clusters with the ground truth labels. All these metrics are measured as a function of the proportion of initialised nodes p.

The first metric that we would like to define is the *Modularity*. *Modularity* is a measure that gives us information about the intrinsic quality of the clusters by comparing the clusters with a random baseline. The value is obtained by calculating the difference in the number of edges inside the cluster with their expected numbers on using the same cluster with random connections between the nodes. If the indicated value is 0, the community structure resembles the random network, else, if the value is close to 1, it depicts a strong community structure. A strong community structure means that there are dense connections inside the communities and sparse connections between them.

The other metrics that will be defined are associated with laying comparison of our clustering algorithm with the ground truth. The *Homogeneity* is a measure that associates how ideal our clusters are, and whether every address in our cluster belongs to the same entity. If it is so, the clusters are called a homogeneous clusters and the measure attains a value of 1. Otherwise, the measure has a value lower than 1.

Another important metric to analyze our results is the *Mutual Information (MI)*. It is defined as the amount of information obtained about any cluster in our clustering output by observing any other cluster within our output. The *Adjusted Mutual Information (AMI)* helps compare our clustering output with the ground truth. It takes a value between 0 and 1. The value of 1 depicts the two comparing values (our clustering output and the ground truth) to be identical. The *Rand Index (RI)* computes a similarity measure between two clusterings by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clusterings.

4 RESULTS

After performing the LPA on the Address Correspondence Network, we obtained a set of clusters for every file used for our analysis. We have obtained results for each metric as a function of p, which is the proportion of initialization of nodes set up before the implication of the community detection algorithm. Every metric has been generated for a pair of detecting heuristic, in which one of the heuristic is always $h0$. We have also considered $h0$ to be our baseline for analyzing the optimal pair for our entity detection algorithm.

Number of Clusters: This is a direct output that we obtain after performing LPA on our Address Correspondence Network. In figure 2, we have shown the number of clusters generated as a function of p for different networks, which include heuristic pairs for 2012 and 2014 data sets. Also plotted in the graphs is the baseline $h0$, for every input file. As observed in the figure, the 2014 data set has produced more clusters than the 2012 data set. On comparing the two with their baselines, we observe that the 2012 data set is closer to its baseline for all heuristic pairs except the $h0 - h5$ pair. The 2014 data set deviates more from its baseline, except for the pairs $h0 - h4$ and $h0 - h5$. Overall, on the basis of deviation from the baseline, $h0 - h4$ pair is giving us the most optimal result. We obtain similar trends of changing clusters as a function of p in the 2012 and 2014 data set. As the value of p increases, the number of clusters are decreasing. They decrease more rapidly in the beginning, but attain steadiness post the p value between 0.15 and 0.2. Thus, we can only initialize a proportion of nodes of about 0.15 to obtain favorable number of clusters.

Adjusted Mutual Information (AMI): Our output for the AMI is given in figure 3. As noticed, the 2014 data set has a better AMI score than the 2012 data set. Both the functions of 2014 and 2012 data increases with increasing value of p, but their nature of change differs slightly. The 2012 version shows slight change for lower values of p in its AMI score, and beyond the value of $p = 0.05$, it increases gradually. On the other hand, the 2014 version shows a sharp increase in the lower values of p, after which it increases smoothly. This shows that the minimum proportion of labelling should at least be 0.05. On comparing the AMI result with the baseline for all the pairs generated, we notice that there is not much divergence of the scores from the baseline result. However, the least variation is seen in the $h0 - h5$ pair of heuristics.

Adjusted Random Index (ARI): In the figure 4, we have obtained the results for the ARI of our two data sets. We notice little difference in the values of ARI for our two data sets. For the 2012 data, the value is seen to increase rapidly from 0 at lower p values, while 2014 data shows lesser, but significant increase at the same values. The 2012 data seems to become stable and show saturation at a value of $p = 0.05$, while the 2014 data gives a steady high value at a lower p value of 0. Thus, good ARI scores are being obtained at low labelling proportion of the nodes in the network. These two data sets also do not seem to diverge from the baseline and all the pairs of heuristics show similar good results.

Homogeneity: Figure 5 shows the results for the Homogeneity of our output. As can be seen, in the case of 2014 data set, as the value of p rises, the value of homogeneity increases. The increase is subtle from p= 0 to 0.10, where the value of homogeneity ≈ 0.8. It increases more sharply from p= 0.10 onward. For the 2012 data set, the value of homogeneity is much lower than that of 2014 data set. Here, as the value of p increases from 0 to ≈ 0.10, the homogeneity value decreases rapidly. After reaching the minimum value, the graphs show a sharp increase in homogeneity values afterwards. It reaches a maximum value of $\approx$ 0.4 at p = 0.40 in all the pair-of-heuristics graphs. Thus, for lower values of p, the clustering becomes less homogeneous for the 2012 data set, and increases after a threshold of $p \approx 0.1$. Both the data sets show little difference from the baseline $h0$, with the difference being minimum in the $h0 - h5$ pair.

Modularity: Figure 6 shows the results for the Modularity measure of our output. As can be seen, the modularity of the 2014 data set is higher than that of the 2012 data set. As the value of p rises, modularity of 2014 data set shows a very slight decrease and then remains roughly the same at the value of 0.8, whereas in the case of 2012 data set, from the value of p = 0 to p =0.05, modularity starts dropping steeply until it reaches zero and remains constant afterwards. This graph demonstrates that the measure of modularity remains highest when the values of p is at the lowest, implying that the community structure remains more rigid and has denser connections with lower values of p. For most of the part, the modularity remains same for all pairs of heuristics, with the baseline not differing in the cases of $h0 - h1$ and $h0 - h5$ graphs.

5 CONCLUSION AND FUTURE WORK

In our report, we have analyzed the clusters obtained after applying Label Propagation Algorithm to the Address Correspondence Networks derived from the data sets of year 2012 and 2014. In these acquired clusters we detect strong communities which signify that the addresses in the cluster belong to the same entity. With the results obtained from the pair of heuristics graphs, it is safe to assume that the recognition of entities in the network is robust and reliable. For the sake of comparison, we have used heuristic $h0$ as a baseline to compare each pair of heuristics of the data sets. It was also observed that the heuristics with false positives do not affect the process significantly. It can be ascertained that even for a few labeling entities from the ground truth data set for clustering, the results observed were optimal. Thus, it can be concluded that given the strong community structure of Address Correspondence Networks, identifying user entities in the Bitcoin network using clustering algorithms is a realizable and viable method.

This project can be taken forward by comparing the results with the randomized versions of the Address Correspondence Networks. Additionally, the values of the number of times the heuristics detect addresses belonging to the same entity can be varied, and the change in the clustering outputs can be observed.

6 CONTRIBUTIONS

The initial code and paper understanding (provided) was done by all authors. Migle Kasetaite and Nidhi Agrawal contributed to the the initial writing of the remaining code and running of the virtual environment. The code was further validated by Afshan Anam Saeed and Sanjana Warambhey. Afshan Anam Saeed and Sanjana Warambhey wrote the first draft of the documentation. The documentation was proof read by Nidhi Agrawal and Migle Kasetaite.

REFERENCES

Fischer, Jan Alexander, Andres Palechor, Daniele Dell'Aglio, Abraham Bernstein, and Claudio J. Tessone (2021) *The Complex Community Structure of the Bitcoin Address Correspondence Network*. arXiv: `2105.09078 [cs.SI■]`

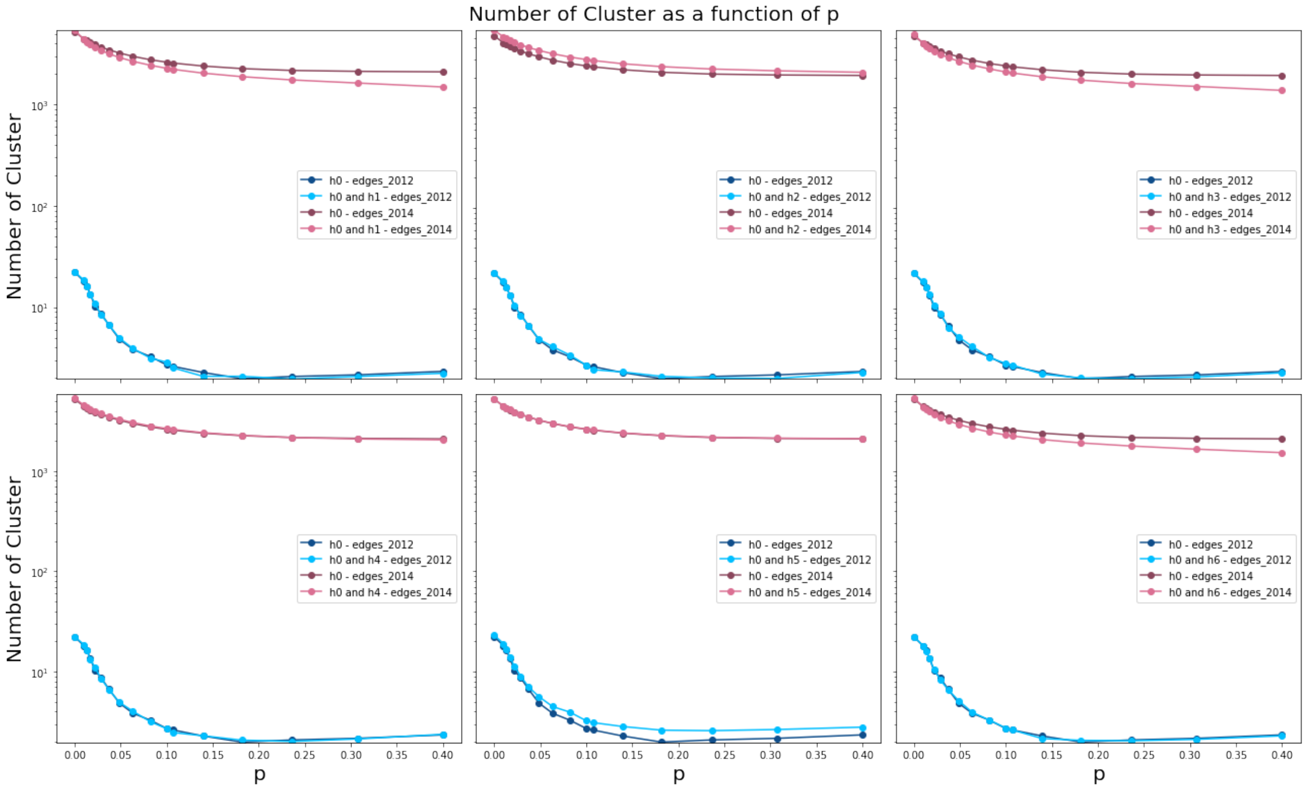

Fig. 2. Number of Clusters.

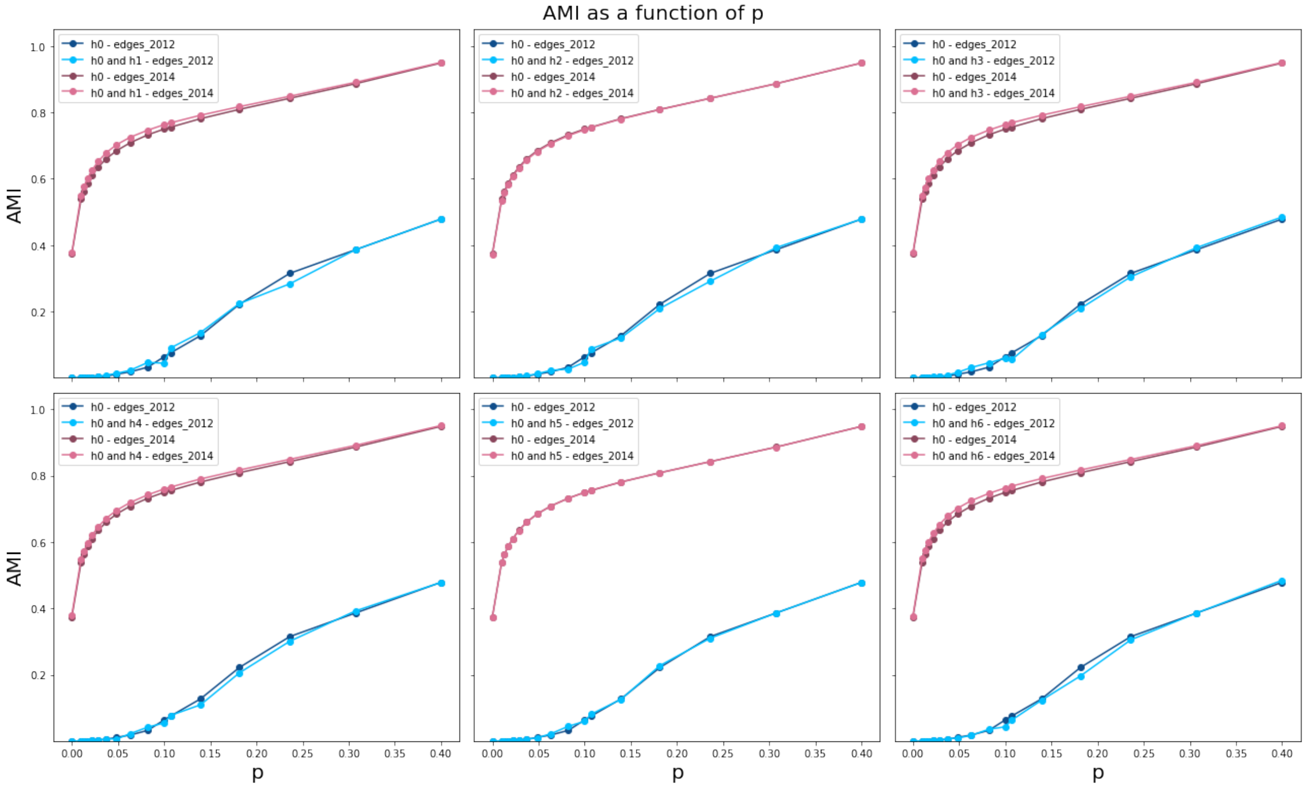

Fig. 3. Adjusted Mutual Information (AMI).

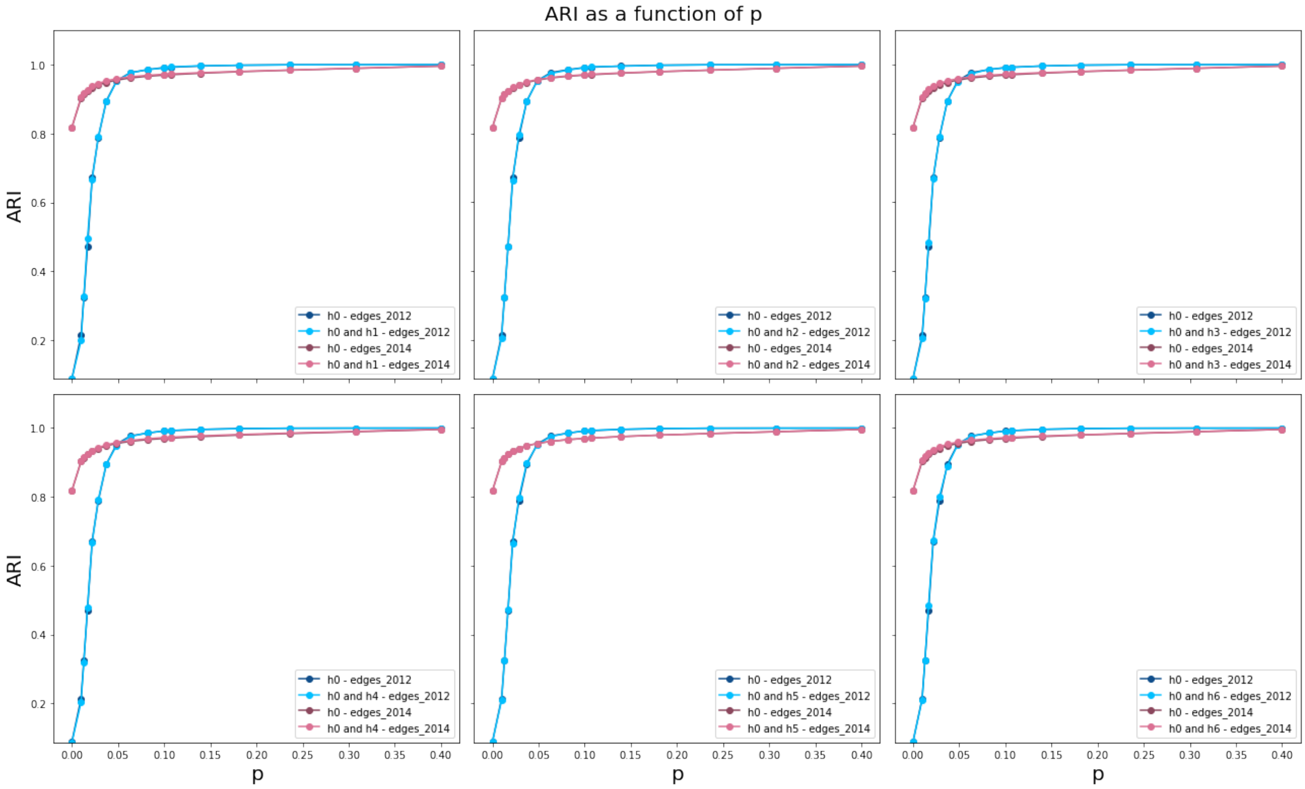

Fig. 4. Adjusted Random Index (ARI).

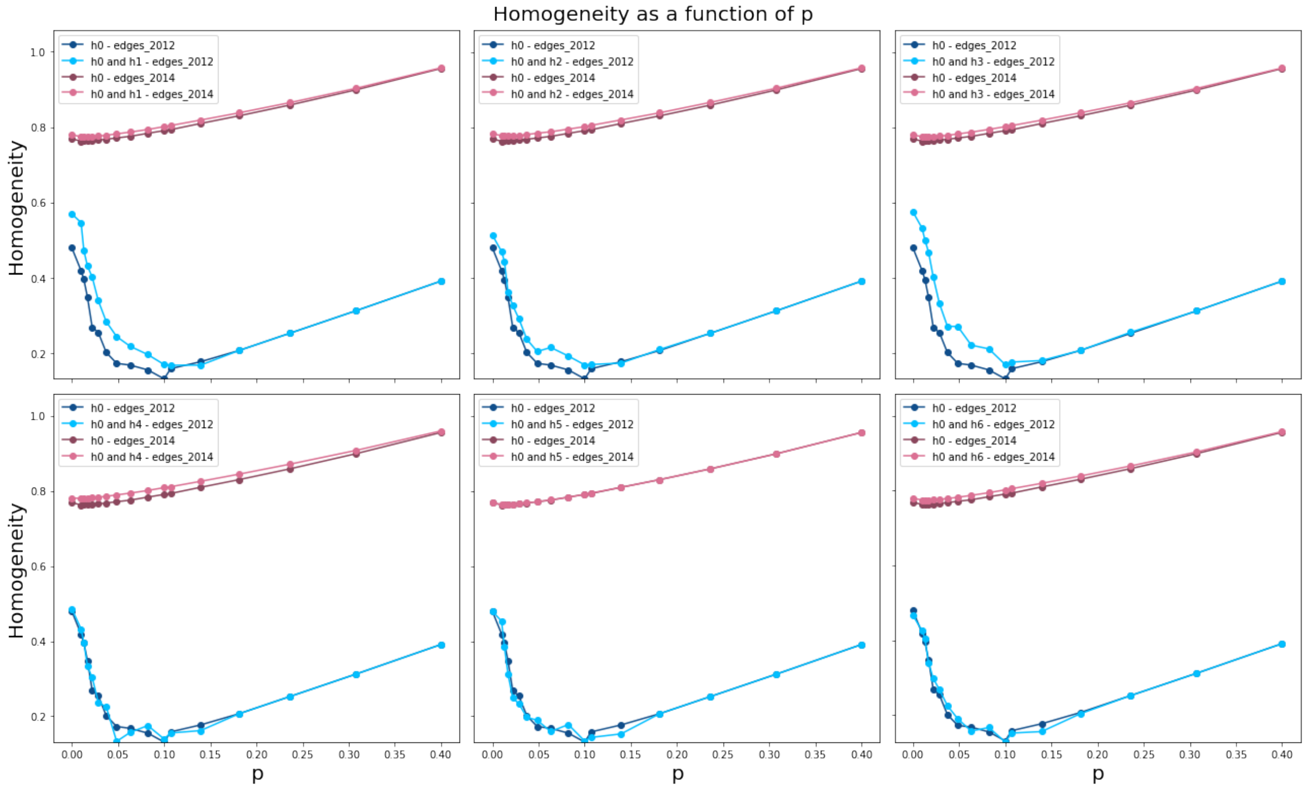

Fig. 5. Homogenity.

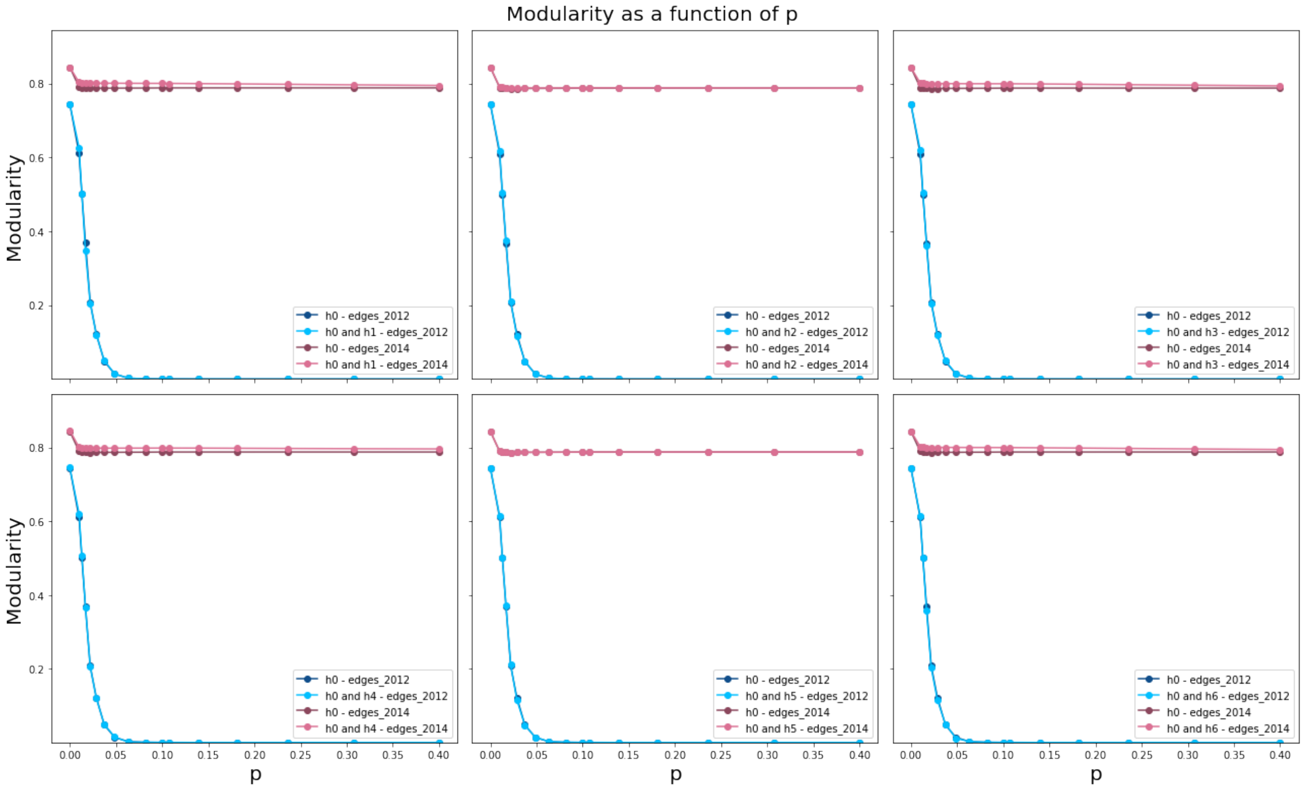

Fig. 6. Modularity.