Check for updates

The Complex Community Structure of the Bitcoin Address Correspondence Network

Jan Alexander Fischer[1†], Andres Palechor[1†], Daniele Dell'Aglio[2,3]*, Abraham Bernstein[3] and Claudio J. Tessone[4]

[1]Faculty of Business, Economics and Informatics, Universität Zürich, Zürich, Switzerland, [2]Department of Computer Science, Aalborg University, Aalborg, Denmark, [3]Department of Informatics, Universität Zürich, Zürich, Switzerland, [4]UZH Blockchain Center and URPP Social Networks, Universität Zürich, Zürich, Switzerland

Bitcoin is built on a blockchain, an immutable decentralized ledger that allows entities (users) to exchange Bitcoins in a pseudonymous manner. Bitcoins are associated with alpha-numeric addresses and are transferred *via* transactions. Each transaction is composed of a set of input addresses (associated with unspent outputs received from previous transactions) and a set of output addresses (to which Bitcoins are transferred). Despite Bitcoin was designed with anonymity in mind, different heuristic approaches exist to detect which addresses in a specific transaction belong to the same entity. By applying these heuristics, we build an Address Correspondence Network: in this representation, addresses are nodes are connected with edges if at least one heuristic detects them as belonging to the same entity. In this paper, we analyze for the first time the Address Correspondence Network and show it is characterized by a complex topology, signaled by a broad, skewed degree distribution and a power-law component size distribution. Using a large-scale dataset of addresses for which the controlling entities are known, we show that a combination of external data coupled with standard community detection algorithms can reliably identify entities. The complex nature of the Address Correspondence Network reveals that usage patterns of individual entities create statistical regularities; and that these regularities can be leveraged to more accurately identify entities and gain a deeper understanding of the Bitcoin economy as a whole.

Keywords: blockchain technology, bitcoin (BTC), label propagarion algorithm, network science, deanonymization

1 INTRODUCTION

Cryptocurrencies are rapidly growing in interest, becoming a popular mechanism to perform pseudonymous exchanges between users (entities). They also allow payments in a decentralized manner without needing a trusted third party. The first and most popular cryptocurrency is Bitcoin, which uses an immutable and publicly available ledger to facilitate transactions between entities. Moreover, given its pseudo-anonymity, Bitcoin has also been used to perform activities in illegal markets. For example, Foley et al. [1] estimate that one-quarter of entities in the Bitcoin network are associated with illegal activity. Consequently, several governing challenges have arisen, and law enforcement agents are particularly interested in techniques that allow tracing the origin of funds. Specifically, in Bitcoin, given the ledger's public nature, tracing the funds can be achieved by inspecting the history of transactions in the system. However, identifying the entities is a complex

task because they can use different pseudonyms (addresses) in the system. By the Bitcoin protocol, it is impossible to completely de-anonymize the entities; however, not all entities prioritize anonymity [2], and it is possible to find recoverable traces of their activity in the transaction history.

The structure of the transactions allows, in some cases, tracing back address pseudonyms that potentially belong to the same entity. For example, Meiklejohn et al. [3] apply heuristics and then cluster together pseudonyms based on evidence of shared spending authority. In this paper, we study the application of several heuristics that leads to creating a sequence of Address Correspondence Networks. Each of these networks includes weighted links between addresses that potentially belong to the same entity, thus approaching entity identification from a network science perspective. Even though other approaches use networks to model some parts of the Bitcoin economic dynamics (e.g. [4–7]), to the best of our knowledge, network science approaches have not addressed the problem of analyzing the Address Correspondence Network to date. In this study, we show that the Address Correspondence Networks have a strong community structure and general-purpose clustering approaches are suitable for analyzing them. Furthermore, our experiments suggest that having a set of identified entities generates large gains in cluster quality—however, this gain quickly declines, and a small number of known entities is enough to produce significant increase in the quality of the detection.

The rest of this paper is organized as follows: **Section 2** explains the basics of the Bitcoin blockchain, heuristics, entity identification and related work. **Section 3** presents our methods for constructing Address Correspondence Networks, the clustering technique and its quality metrics. In **Section 4**, we discuss our findings, and finally, in **Section 5**, we discuss conclusion and future work.

2 BACKGROUND AND RELATED WORK

This section introduces the main concepts related to Bitcoin. Next, it discusses the task of identifying addresses controlled by the same entity, followed by a reviews of the main studies in the area.

2.1 The Bitcoin Blockchain

Bitcoin was introduced in [8] as a decentralized payment network and digital currency which would be independent of central bank authorities. It is built on a blockchain, an immutable decentralized ledger that allows users, i.e. entities, to exchange the units of account (Bitcoins) in a pseudonymous manner. Entities transacting in the Bitcoin network control addresses—unique identifiers which have the right to transfer specific amounts of Bitcoins.

There are different types of addresses, which determine how the associated Bitcoins are accessed. For example, to spend Bitcoins associated with an address of type Pay to Public Key Hash (P2PKH), the entity needs to present a valid signature based on their private key, and a public key that hashes to the P2PKH value. Another example is the Pay to Script Hash (P2SH) address type: it defines a script for custom validation, which may include several signatures, passwords and other user-defined requirements. We denote with a an address and with $\mathcal{A}$ the set of $\{a_1, \ldots, a_n\}$ addresses appearing in the Bitcoin blockchain. Furthermore, we denote an entity as e, with $\mathcal{E}$ representing the set $\{e_1, \ldots, e_k\}$ of entities that own Bitcoin addresses.

To spend or receive Bitcoins, entities create transactions. A transaction t is composed of a set of input addresses, a set of output addresses, and information specifying the amount of Bitcoins to be allocated to each output address. Formally, let $\mathcal{T}$ be the set of transactions stored in the Bitcoin blockchain, and $\mathcal{P}(\mathcal{A})$ be the power set of $\mathcal{A}$. We model with i : $\mathcal{T} \rightarrow \mathcal{P}(\mathcal{A})$ and o : $\mathcal{T} \rightarrow \mathcal{P}(\mathcal{A})$ the mappings between a transaction and its input and output address sets. The sum of Bitcoins associated with the input addresses equals the sum of Bitcoins associated with the output addresses plus transaction fees. Therefore, if an entity wishes to spend only a partial amount of Bitcoins associated with the input addresses, the remainder is typically sent to an existing or newly created change address controlled by the initiating entity. Transaction outputs that have not yet been used as inputs to other transactions are referred to as UTXOs (unspent transaction outputs).

The transaction history is replicated on multiple nodes in the Bitcoin network. Entities broadcast new transactions to other nodes in the network. As part of Bitcoin's decentralized consensus protocol, specialized miner nodes are incentivized to solve proof-of-work puzzles that validate new transactions and group them into blocks. Blocks are sequentially appended to the blockchain; the number of blocks preceding a particular block is known as its block height. Furthermore, entities may specify a transaction's locktime. This is the minimum block height the blockchain must reach before miners should consider validating the transaction, i.e. a transaction with locktime j is added to block $j + 1$ or later.

A peculiar property of the Bitcoin network is the pseudonymity: entities conceal their identity through the use of nameless addresses (pseudonyms), linking an address to a real-world entity exposes their entire activity on the Bitcoin network, since the transaction history is publicly available. Entities are therefore advised to generate a new address for every transaction, so that each address is used once as a transaction output and once as a transaction input.

2.2 Address Clustering

The objective of address clustering is to find sets of addresses $\mathcal{A}_i \subseteq \mathcal{A}$ that are controlled by the same entity e_i. Formally, the objective is to find a map e : $\mathcal{A} \rightarrow \mathcal{E}$ such that $\mathcal{A}_i = \{a_j | e(a_j) = e_i\}$. There exist multiple heuristics for identifying address pairs controlled by the same entity. We consider seven heuristics implemented by Kalodner et al. [9], the majority of which seek to identify change addresses in the outputs of a transaction (linking these with the transaction inputs).

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 Meiklejohn et al. [3].

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.

2.3 Related Work

Address clustering in Bitcoin has been the subject of numerous studies. Initial studies focused on the multi-input heuristic. For example, Nick [10] identify more than 69% vulnerable addresses using only this heuristic. Also Harrigan and Fretter [11] consider the multi-input heuristic and attribute its effectiveness to frequent address reuse, as well as the presence of large address clusters having high centrality measures with respect to transactions between clusters. Furthermore, they suggest that incremental cluster growth and the avoidable merging of large clusters makes the multi-input heuristic suitable for real-time analysis. Fleder et al. [12] construct directed transaction graphs for periods of 24 h and 7 months. In such graphs, the nodes are addresses and each edge represents a transaction from an input address to an output address. They obtain address entity labels by scraping public forums and social networks. By applying the multi-input heuristic, they identify transactions where labeled addresses have interacted with a large number of known entities such as SatoshiDICE and Wikileaks.

Meiklejohn et al. [3] combines the multi-input heuristic with a second one, similar to the change address behavior heuristic. They identify major entities and interactions between them, and note that the change address heuristic tends to collapse address groups into large super-clusters. Zhang et al. [13] consider another variation of the change address behavior heuristic, and show that it improves clustering quality when address reduction is used as a performance measure. In this study, we focus on the heuristics introduced in Section 2.2 by Kalodner et al. [9].

Patel [14] proposes novel approaches to Bitcoin address clustering. He considers clustering an undirected, weighted heuristic graph, where the nodes are addresses, and each edge indicates the presence of at least one of eight heuristics (a superset of those introduced in Section 2.2) linking those addresses to the same entity. Each heuristic is assigned a positive weight, such that their sum is equal to one. The edge weight is the sum of the heuristic weights for which the corresponding heuristic is present between two addresses. The author applies a variety of generic graph clustering algorithms (e.g. k-means, spectral, DBSCAN) as well as graph sparsification and coarsening techniques to the constructed heuristic graph. In this study, we propose the address correspondence network, which is similar to the network built by Patel [14] However, in our correspondence network, an edge between two addresses represents the number of times the heuristics identify the pair as controlled by the same entity. We use a label propagation algorithm to build the clusters, using ground truth information to drive the algorithm.

There exist other approaches and extensions to address clustering. Ermilov et al. [15] show that higher cluster homogeneity can be achieved when transaction data is augmented with off-chain information from the internet. Biryukov and Tikhomirov [16] propose incorporating lower-level network information to enhance deanonymization. Furthermore, Harlev et al. [17] extend address clustering by using supervised machine learning to predict the type of entity controlling addresses in an unlabeled cluster. In our study, in addition to using a ground truth to guide the clustering construction, we introduce a temporal component in the analysis. We build address correspondence networks for various time intervals. In this way, we can analyze the evolution of the network over time.

3 METHODOLOGY

We expand upon the work of Patel [14] by performing address clustering on so-called Address Correspondence Networks, denoted $\mathcal{G}_{[o,c]}$, where $[o, c]$ is a time interval. Nodes are Bitcoin addresses that are involved in transactions between a time instant o and a time instant c. $\mathcal{G}_{[o,c]}$ contains an undirected link (a_i, a_j) between two addresses when at least one of the heuristics introduced in Section 2.2 detects a_i and a_j as belonging to the same entity. We posit that the topology of $\mathcal{G}_{[o,c]}$ encodes further insights on the identity of the entities and, ultimately, on the e (a_j) map.

For some addresses a_j, the controlling entity is known. Using the block explorer tool provided by Janda [18], we obtain entity labels for 28 million addresses involved in transactions before 2017. We refer to this data set as the ground truth. The mapping information contained in the ground truth is denoted with $e^\star$, such that $\mathcal{A}^\star = \{a_j | \exists e^\star (a_j)\} \subseteq \mathcal{A}$ is the set of addresses for which the entity label is known. We use the ground truth to 1) sample from $\mathcal{T}$ and 2) to evaluate the quality of address clustering methods.

The remainder of this section is organized as follows. Section 3.1 describes the method for sampling from $\mathcal{T}$. This sample is divided further into cumulative and partial subsets, which are described in Section 3.2. Section 3.3 details the construction of the
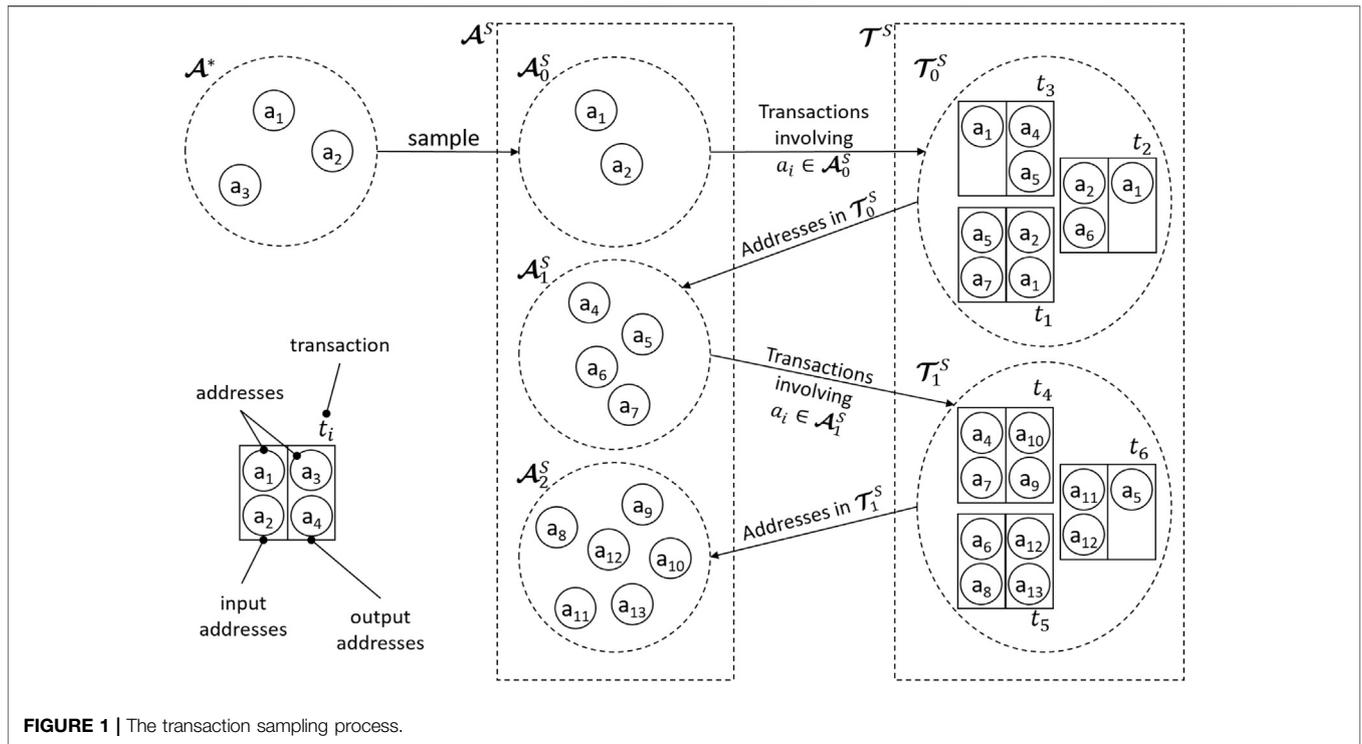
FIGURE 1 | The transaction sampling process.

Address Correspondence Networks. We explain our approach to clustering these networks in Section 3.4, while the metrics used to evaluate clustering quality are introduced in Section 3.5.

3.1 Transaction Sampling

For computational feasibility, we restrict our analysis to a sample of $\mathcal{T}$, as depicted in **Figure 1**. First, we randomly select a subset $\mathcal{A}_0^S \subseteq \mathcal{A}^\star$ of the addresses in the ground truth. Next, we select all transactions involving an address $a \in \mathcal{A}_0^S$ as an input or output, i.e., $\mathcal{T}_0^S = \{t \mid \exists a \in \mathcal{A}_0^S : a \in i(t) \cup o(t)\}$. We then build the set $\mathcal{A}_1^S$ of addresses that appear in transactions of $\mathcal{T}_0$ but not in $\mathcal{A}_0^S$, i.e. $\mathcal{A}_1^S = \{a \mid a \notin \mathcal{A}_0^S \wedge \exists t \in \mathcal{T}_0^S : a \in i(t) \cup o(t)\}$. The aforementioned process is then repeated in a similar manner. This involves finding the set $\mathcal{T}_1^S$ of transactions which include at least two addresses in $\mathcal{A}_1^S$, i.e., $\mathcal{T}_1^S = \{t \mid t \notin \mathcal{T}_0^S \wedge \exists a_1, a_2 \in \mathcal{A}_1^S : a_1 \in i(t) \cup o(t) \wedge a_2 \in i(t) \cup o(t) \wedge a_1 \neq a_2\}$. We set the condition on two addresses per transaction to reduce the size of the subsequently constructed Address Correspondence Networks. Finally, we build $\mathcal{A}_2^S$ as the addresses appearing in transactions of $\mathcal{T}_1^S$ and not already in $\mathcal{A}_0^S$ or $\mathcal{A}_1^S$, i.e. $\mathcal{A}_2^S = \{a \mid a \notin \mathcal{A}_0^S \cup \mathcal{A}_1^S \wedge \exists t \in \mathcal{T}_1^S : a \in i(t) \cup o(t)\}$.

As a result, this process constructs a set of sampled transactions $\mathcal{T}^S = T_0^S \cup T_1^S$ having addresses $\mathcal{A}^S = \mathcal{A}_0^S \cup \mathcal{A}_1^S \cup \mathcal{A}_2^S$. An advantage of this sampling method is that the constructed Address Correspondence Networks are centered around ground truth seed addresses, thereby exploiting the previous knowledge of controlling entities.

3.2 Partial and Cumulative Transaction Sets

To study the evolution of the Bitcoin Address Correspondence Network over time, we create temporal subsets of the transactions in $\mathcal{T}^S$. Each subset includes only the transactions in $\mathcal{T}^S$ that were generated in a specific time interval. We create time intervals using two different strategies, which we name cumulative and partial, summarized in **Figure 2**.

The cumulative strategy creates eight time intervals of progressively increasing width,[1] $\{[01.07.11, 30.06.y], [01.07.11, 31.12.y] \mid y \in [12, 15]\}$, while the partial strategy creates eight time intervals of fixed width, $\{[01.01.y, 30.06.y], [01.07.y, 31.12.y] \mid y \in [12, 15]\}$. It follows that cumulative time intervals overlap, while partial time intervals are disjoint.

Cumulative transaction sets are denoted with $\mathcal{T}_{[11s2, yss]}^S$, which refers to all transactions in $\mathcal{T}^S$ that were generated between the second semester of 2011 and the sth semester of y, e.g., $\mathcal{T}_{[11s2, 14s1]}^S$ includes transactions generated in the interval $[01.07.11, 30.06.14]$. Partial transaction sets are denoted with $\mathcal{T}_{[yss, yss]}^S \equiv \mathcal{T}_{[yss]}^S$, e.g., $\mathcal{T}_{[14s1]}^S$ refers to transactions generated in the interval $[01.01.14, 30.06.14]$. It is worth noting that while partial transaction sets do not share transactions, they may still share addresses which are used in multiple transactions.

3.3 Address Correspondence Network Construction

Let $w : \mathcal{A} \times \mathcal{A} \to \mathbb{N}$ be a function that counts how often an address pair, (a_1, a_2), is detected by any of the seven heuristics introduced in Section 2.2 as being controlled by the same entity

---

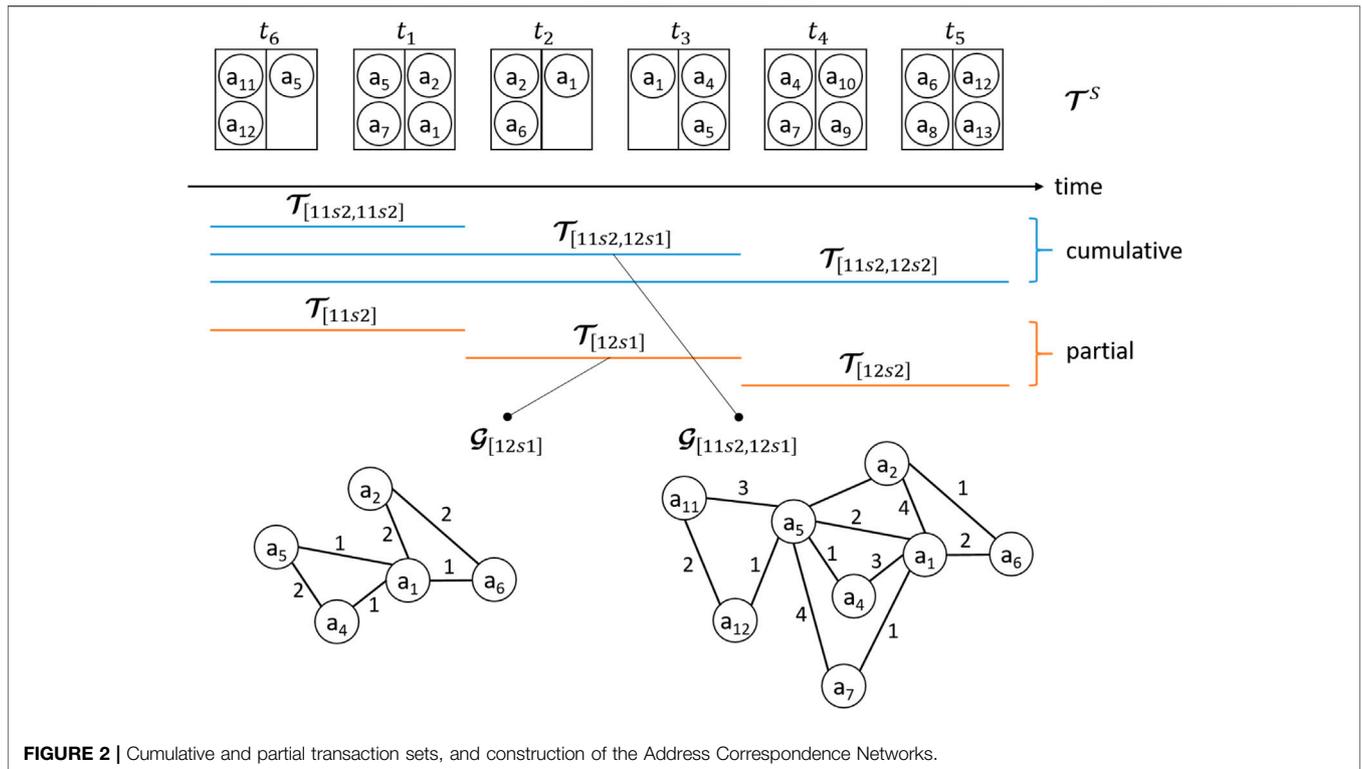[1]We represent dates in the use the DD.MM.YY format.

FIGURE 2 | Cumulative and partial transaction sets, and construction of the Address Correspondence Networks.

(considering only transactions in $\mathcal{T}^S_{[o,c]}$). It is worth noting that w is symmetric (or undirected), i.e. $w(a_1, a_2) = w(a_2, a_1)$.

The information captured by applying w to each pair of addresses in $\mathcal{A}^S_{[o,c]}$ is collected in Address Correspondence Networks, defined as undirected weighted graphs $\mathcal{G}_{[o,c]} = (\mathcal{A}^S_{[o,c]}, \mathcal{L}_{[o,c]}, w)$. The construction process is depicted in **Figure 2**. The addresses in $\mathcal{A}^S_{[o,c]}$ are the vertices of the graph, and w is the weight function. $\mathcal{L}_{[o,c]} \subseteq \mathcal{A}^S_{[o,c]} \times \mathcal{A}^S_{[o,c]}$ is the set of edges connecting address in two ways:

1) Pairs (a_i, a_o) such that it exists a transaction $t \in \mathcal{T}^S_{[o,c]}$ having respectively a_i and a_o in its input and output address sets $i(t)$ and $o(t)$, and having $w(a_i, a_o) > 0$.
2) Pairs (a_{i_1}, a_{i_2}) such that it exists a transaction $t \in \mathcal{T}^S_{[o,c]}$ having both a_{i_1} and a_{i_2} in its input set $i(t)$, and having $w(a_{i_1}, a_{i_2}) > 0$.

Note that in a transaction, different heuristics can concur by identifying the same address as a change address, increasing the weights of the edges related to such an address. **Figure 3** shows the degree distribution of the Address Correspondence Networks $\mathcal{G}_{[11s2,12s1]}$ and $\mathcal{G}_{[11s2]}$. The two distributions show a similar shape, but note that the left plot is a cumulative graph and the right plot is a partial graph; this indicates that the correspondence networks appear to preserve common properties across time. **Table 1** provides descriptive statistics of the 16 Address Correspondence Networks we constructed from the eight partial and cumulative transaction sets. While the degree distributions cannot be assimilated to a single statistical distribution, they are skewed and fat-tailed, features that are

recognized in complex networks of different contexts like biological, technological or social interactions [19].

Figure 4 shows the distribution of ground truth entities in the Address Correspondence Networks. In each plot, we compare a cumulative network and the partial network from its last six months, e.g. $\mathcal{G}_{[11s2,13s1]}$ with $\mathcal{G}_{[13s1]}$. The number of known entities in the networks from 2012 is small, $\mathcal{G}_{[12s1]}$ and $\mathcal{G}_{[12s2]}$ do not show any relation with their pairs. However, from 2013, the similarity between distributions of known entities of partial and cumulative networks is notorious.

3.4 Address Correspondence Network Clustering

Let $\mathcal{G}_{[o,c]} = (\mathcal{A}^S_{[o,c]}, \mathcal{L}_{[o,c]}, w)$ be the Address Correspondence Network for the time interval $[o, c]$. We approach the entity identification problem by applying a community detection algorithm to $\mathcal{L}_{[o,c]}$ (therefore assuming that communities are sets of addresses belonging to the same entity). In $\mathcal{G}_{[o,c]}$, highly interconnected vertices are clusters (communities) of addresses linked by one or several heuristics. Community detection algorithms find clusters of vertices highly interconnected but with sparse links between clusters. Specifically, the Label Propagation Algorithm (LPA) by Raghavan et al. [20] finds communities and has linear complexity on the number of edges $\mathcal{O}(\mathcal{L}_{[o,c]})$. The comparative study by Yang et al. [21] shows that the scalability of LPA outperforms other fast clustering algorithms, including Leading Eigenvector by Newman [22], Walktrap by Pons and Latapy [23], and Multilevel by Blondel et al. [24]. In LPA, each node is
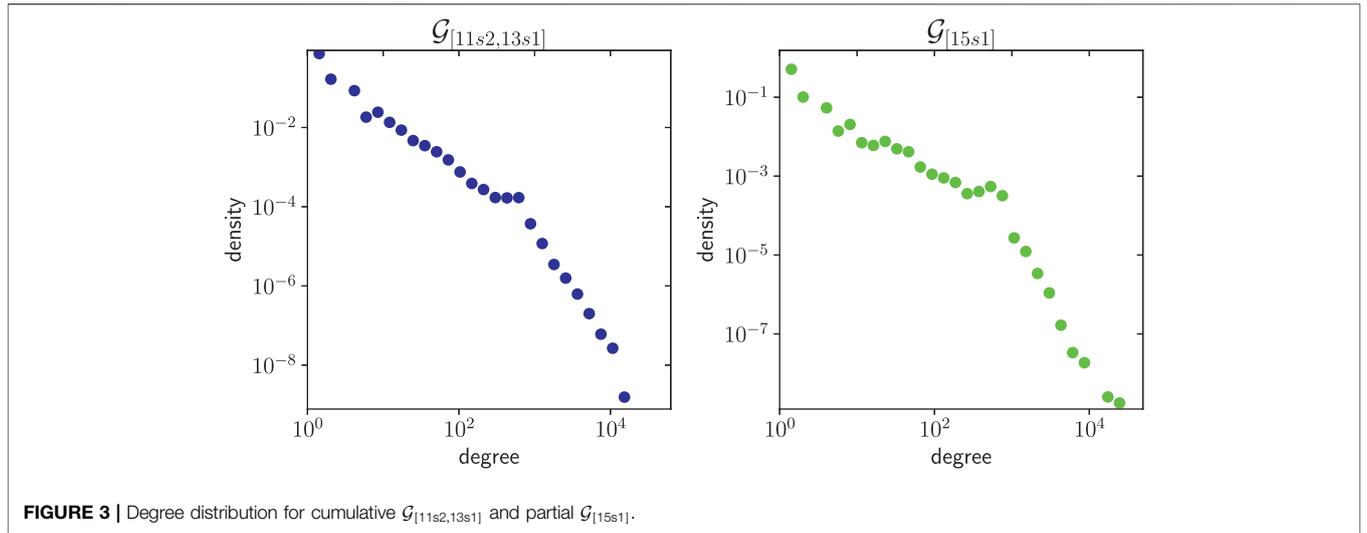
FIGURE 3 | Degree distribution for cumulative $\mathcal{G}_{[11s2,13s1]}$ and partial $\mathcal{G}_{[15s1]}$.

TABLE 1 | Number of nodes, edges and ground truth addresses of the partial and cumulative Address Correspondence Networks for each semester from 2012 to 2015.

y	s	Partial: $\mathcal{G}_{[yss]}$			Cumulative: $\mathcal{G}_{[11s2,yss]}$		
		$\left\|\mathcal{A}^{\mathbf{S}}_{[\mathbf{yss}]}\right\|$	$\left\|\mathcal{L}_{[\mathbf{yss}]}\right\|$	$\left\|\mathcal{A}^{\star}_{[\mathbf{yss}]}\right\|$	$\left\|\mathcal{A}^{\mathbf{S}}_{[\mathbf{11s2,yss}]}\right\|$	$\left\|\mathcal{L}_{[\mathbf{11s2,yss}]}\right\|$	$\left\|\mathcal{A}^{\star}_{[\mathbf{11s2,yss}]}\right\|$
2012	1	12	46	10	3,750	164,408	1,553
	2	5,054	1,239,850	5,029	8,804	1,404,258	6,582
2013	1	131,252	3,183,594	39,161	139,918	4,587,813	45,613
	2	191,453	45,965,678	155,449	329,240	50,552,843	199,614
2014	1	360,002	81,891,103	268,228	607,098	131,854,323	396,548
	2	505,748	31,121,336	233,609	1,092,560	162,948,611	621,919
2015	1	232,781	16,836,377	120,191	1,270,261	179,725,740	734,185
	2	990,117	52,732,659	211,174	2,184,445	232,416,368	935,599

FIGURE 4 | Distribution of ground truth entity sizes, $\left|\mathcal{E}^{\star}_{[o,c]}\right|$.
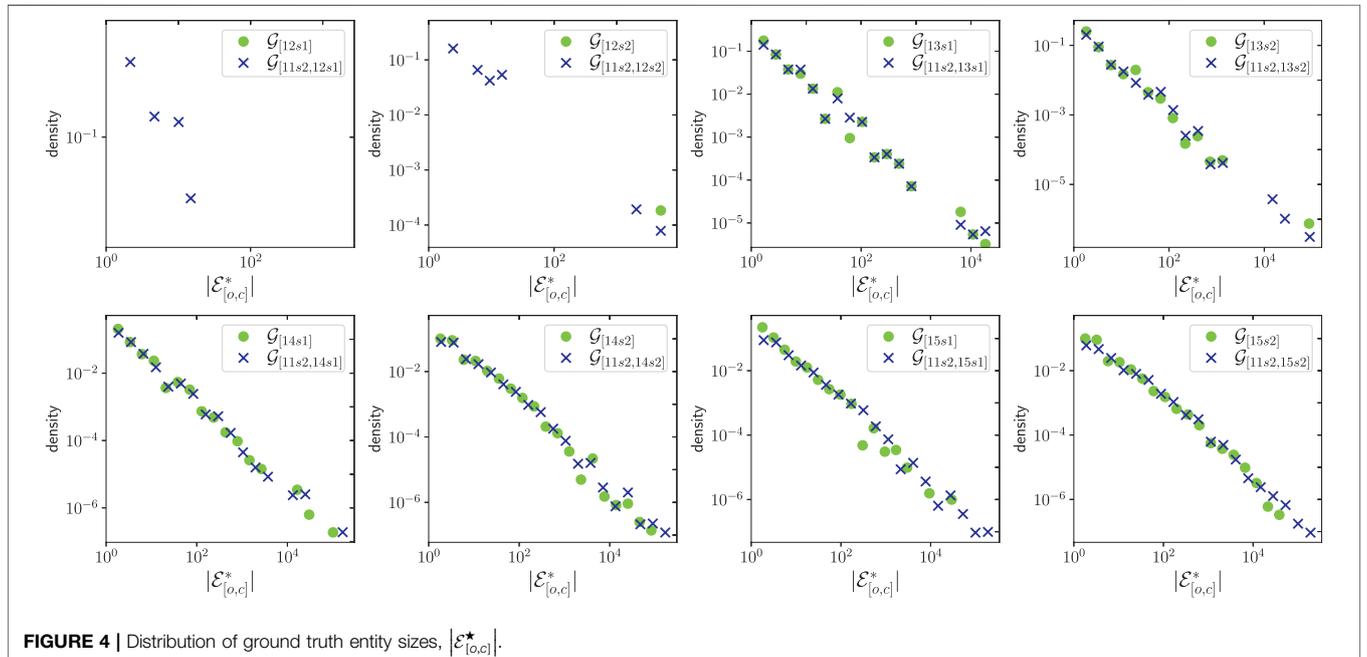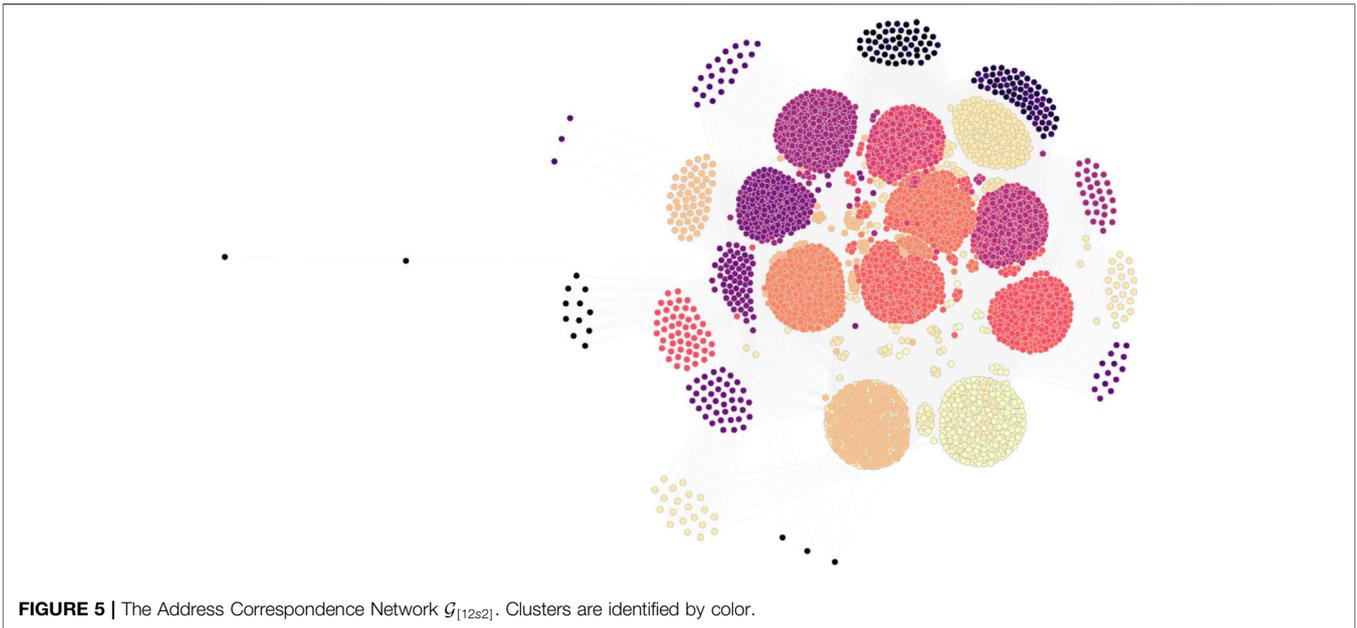
FIGURE 5 | The Address Correspondence Network $\mathcal{G}_{[12s2]}$. Clusters are identified by color.

initialized with a unique label, denoting the cluster it is part of (the controlling entity of an address). In the basic case, all the nodes are initially assigned a random label. Afterward, each node is randomly visited and assigned a label according to the majority voting of its neighbors. The process repeats until every node in the network gets a label to which most of its neighbors belong. **Figure 5** shows a clustering for the partial network $\mathcal{G}_{[12s2]}$.

To initialize parts of the nodes, we use the information from the ground truth $e^\star$. Let $\mathcal{A}^\star_{[o,c]}$ denote the set of ground truth addresses in $\mathcal{G}_{[o,c]}$, i.e., $\mathcal{A}^\star_{[o,c]} = \mathcal{A}^S_{[o,c]} \cap \mathcal{A}^\star$, and let $\mathcal{E}^\star_{[o,c]}$ be the set of ground truth entities in $\mathcal{G}_{[o,c]}$, i.e. $\mathcal{E}^\star_{[o,c]} = \{e^\star(a) | \exists a \in \mathcal{A}^\star_{[o,c]}\}$. We assign to a subset of nodes $\mathcal{A}^I_{[o,c]} \subseteq \mathcal{A}^\star_{[o,c]}$ the label from the ground truth, i.e. $e^\star(a)$. It holds that $\mathcal{A}^I_{[o,c]} \subseteq \mathcal{A}^\star_{[o,c]} \subseteq \mathcal{A}^S_{[o,c]}$ and, concomitantly, $\left|\mathcal{A}^I_{[o,c]}\right| \le \left|\mathcal{A}^\star_{[o,c]}\right| \le \left|\mathcal{A}^S_{[o,c]}\right|$.

In this paper, we are interested in exploring the ability of community detection algorithms to provide additional information about the true identities of users. We hypothesize that the Address Correspondence Network encodes additional information about the entities that control specific addresses. We argue that successive applications of heuristics may lead to connections between addresses controlled by the same entity that are denser and higher weighted than connections between addresses of different entities. Following this argument, we apply LPA to obtain a disjoint set of clusters $\mathcal{C}_{[o,c]} = \{\mathcal{C}^{(1)}_{[o,c]}, \ldots, \mathcal{C}^{(k)}_{[o,c]}\}$, such that $\cup_{i=1}^k \mathcal{C}^{(i)}_{[o,c]} = \mathcal{A}^S_{[o,c]}$. Because of the additional information provided by the ground truth, we modified LPA to avoid that the addresses in $\mathcal{A}^I_{[o,c]}$ can change label, as they are associated with the actual entity according to the ground truth information.

In the experiments, we vary the proportion p of initialized nodes, that is defined as:

$$p = \left|\mathcal{A}^I_{[o,c]}\right| / \left|\mathcal{A}^S_{[o,c]}\right|.$$

Since $\left|\mathcal{A}^\star_{[o,c]}\right| / \left|\mathcal{A}^S_{[o,c]}\right|$ varies across networks and is an upper bound on the proportion of initialized nodes, the domains of the approximated functions also vary.

3.5 Cluster Quality Analysis

Finally, we quantify the clustering quality as a function of cluster size and entity size. Given an Address Correspondence Network $\mathcal{G}_{[o,c]}$ and set of clusters $\mathcal{C}_{[o,c]} = \{\mathcal{C}^{(1)}_{[o,c]}, \ldots, \mathcal{C}^{(k)}_{[o,c]}\}$ produced by LPA, we analyze the quality of $\mathcal{C}_{[o,c]}$ by defining a set of discrete random variables to describe characteristics of the network, and by five metrics: modularity to give information about the intrinsic quality of the clusters (and inherent topological structure of the network), homogeneity, entropy, Adjusted Mutual Information (AMI) and Adjusted Rand Index (ARI) to compare the clusters with the ground truth labels. Furthermore, all metrics are measured as functions of the proportion of initialized nodes p.

3.5.1 Random Variables

To study the characteristics of the network, we define the following discrete random variables associated with the distributions of entities, addresses, and known addresses in the address correspondence network.

The first random variable, E, assumes a value from the set of entities according to their frequency in the correspondence network. More specifically, E can assume the value $e \in \mathcal{E}^\star_{[o,c]}$ with probability equal to the numbers of addresses in $\mathcal{A}^\star_{[o,c]}$ mapped to e, divided by the total number of addresses in $\mathcal{A}^\star_{[o,c]}$, i.e.:

$$P(e) = \frac{\left|\{a \in \mathcal{A}^\star_{[o,c]} | e = e^\star(a)\}\right|}{\left|\mathcal{A}^\star_{[o,c]}\right|}.$$

In addition to E, we also define variables that assume values in the entity set according to their frequency in specific clusters. Let

E_i be the variable associated to the ith cluster, i.e. $i \in [1, |\mathcal{C}_{[o,c]}|]$. For each i, we build a histogram of the frequency of entities in $\mathcal{C}_{[o,c]}^{(i)}$, by counting for each entity e the number of addresses associated to e through the ground truth data in $\mathcal{C}_{[o,c]}^{(i)}$. Such a histogram is used to approximate the distribution of entities over $\mathcal{C}_{[o,c]}^{(i)}$ and serves to describe E_i. Formally let $\mathcal{A}_{[o,c]}^{(i)} = \mathcal{A}_{[o,c]}^{\star} \cap \mathcal{C}_{[o,c]}^{(i)}$ be the set of addresses in $\mathcal{C}_{[o,c]}^{(i)}$ which are part of the ground truth. E_i can assume a value e in $\mathcal{E}_{[o,c]}^{(i)} = \{e^{\star}(a) | a \in \mathcal{A}_{[o,c]}^{(i)}\}$ with probability:

$$P(e) = \frac{|\{a \in \mathcal{A}_{[o,c]}^{(i)} | e = e^{\star}(a)\}|}{|\mathcal{A}_{[o,c]}^{(i)}|}.$$

The variable C assumes a cluster identifier according to its frequency over the addresses in the ground truth. C can assume a value $\mathcal{C}_{[o,c]}^{(i)} \in \mathcal{C}_{[o,c]}$ with probability defined by the number of addresses in $\mathcal{A}_{[o,c]}^{\star}$ and $\mathcal{C}_{[o,c]}^{(i)}$ (i.e. $\mathcal{A}_{[o,c]}^{(i)}$) divided by the total number of addresses in $\mathcal{A}_{[o,c]}^{\star}$, i.e.:

$$P\left(\mathcal{C}_{[o,c]}^{(i)}\right) = \frac{|\mathcal{A}_{[o,c]}^{(i)}|}{|\mathcal{A}_{[o,c]}^{\star}|}.$$

Finally, we define variables complementary to E_i to describe the frequency of clusters among each entity. We indicate with C_j the variable associated to the jth entity e_j, with $j \in [1, |\mathcal{E}_{[o,c]}^{\star}|]$. Given the entity e_j, we build the histogram of the appearance of e_j in each cluster of $\mathcal{C}_{[o,c]}$. As for the E_i variables, we approximate the real distribution using the ground truth data, and considering only the addresses from $\mathcal{A}^{\star}$ to build the bins. Formally, C_j can assume values in $\mathcal{C}_{[o,c]}$ with probability:

$$P\left(\mathcal{C}_{[o,c]}^{(i)}\right) = \frac{|\{a \in \mathcal{C}_{[o,c]}^{(i)} | e_j = e^{\star}(a)\}|}{|\{a \in \mathcal{A}_{[o,c]}^{\star} | e_j = e^{\star}(a)\}|}.$$

3.5.2 Metrics

Modularity, initially proposed by Newman and Girvan [25], compares the clusters with a random baseline. This is done by computing the difference between the number of edges inside the clusters with the expected value of edges using the same clusters but with random connections between the nodes. Let $|\mathcal{C}_{[o,c]}|$ be the number of clusters in the Address Correspondence Network $\mathcal{G}_{[o,c]}$, q_{ij} the ratio of edges connecting addresses between cluster $\mathcal{C}_{[o,c]}^{(i)}$ and cluster $\mathcal{C}_{[o,c]}^{(j)}$, and $r_i = \sum_j q_{ij}$ the ratio of edges with at least one end in $\mathcal{C}_{[o,c]}^{(i)}$. The modularity is defined as:

$$Q = \sum_{i=1}^{|\mathcal{C}_{[o,c]}|} (q_{ii} - r_i^2).$$

A value close to 0 indicates that the community structure is akin to a random network, while values close to 1 indicate strong community structures, meaning dense connections inside the communities and sparse connections between them.

Information Theory Metrics: Entropy, introduced in an information theory context by Shannon [26], quantifies the expected amount of information or uncertainty contained in a random variable. Let X be a discrete random variable, which can assume values $\{x_1, x_2, \ldots, x_k\}$ with probability $\{P(x_1), P(x_2), \ldots, P(x_k)\}$. The entropy of X is defined as:

$$H(X) = -\sum_{x \in 1}^{k} P(x)\log_2 P(x),$$

while the normalized Shannon entropy is:

$$\widehat{H}(X) = \frac{H(X)}{H_{max}(X)} = \frac{H(X)}{\log_2(k)}.$$

We use the normalized entropy of E_i and C_j to study the clusters by the perspective of the entities and the one of the cluster themselves.

Entropy also gives important information of the interrelation between random variables. Let us consider two variables X and Y, and let $P(X, Y)$ be the joint probability distribution. The conditional entropy $H(Y|X)$ is defined as:

$$H(Y|X) = -\sum_{x \in X, y \in Y} P(x,y) \frac{\log_2 P(x,y)}{P(x)}$$

The conditional entropy indicates how much extra information is needed to describe Y given that X is known. Additionally, the amount of information needed on average to specify the value of two random variables is $H(X, Y) = H(X|Y) + H(Y)$.

We use conditional entropy to measure the quality of the clusters. We do it by comparing them with the distribution of the entities in the Address Correspondence Network, exploiting the variables E and C. Such a measure is named homogeneity and is initially introduced by Rosenberg and Hirschberg [27]. Ideally, a cluster should only contain addresses that are controlled by the same entity. In such a case, clusters are homogeneous and it holds $H(E|C) = 0$. The homogeneity score $h \in [0, 1]$ is defined by:

$$h = \begin{cases} 1 & \text{if } H(E, C) = 0 \\ 1 - H(E|C)/H(E) & \text{otherwise} \end{cases}.$$

The fundamental Mutual Information (MI) [28] quantifies the agreement between partitions. In addition to $\mathcal{C}_{[o,c]}$, let $\mathcal{K}_{[o,c]} = \{\mathcal{K}_{[o,c]}^{(1)}, \ldots, \mathcal{K}_{[o,c]}^{(k)}\}$ be an alternative set of clusters. We introduce the variable K to describe the distribution of the addresses in $\mathcal{K}_{[o,c]}$, similarly to how we defined C for $\mathcal{C}_{[o,c]}$ in Section 3.5.1. The MI of C and K is defined as:

$$MI(C, K) = H(K) - H(K|C),$$

and quantifies the reduction of the uncertainty of $\mathcal{C}_{[o,c]}$ due to the knowledge of $\mathcal{K}_{[o,c]}$. The average MI value between $\mathcal{C}_{[o,c]}$ and $\mathcal{K}_{[o,c]}$ tends to increase as the number of clusters increases, even if there is no difference in the clustering methodology, e.g. if the partitions are assigned clusters randomly. The Adjusted Mutual Information defined by Vinh et al. [29] takes into account the randomness using the expected value of MI $E[MI]$ and normalizes its value:

$$AMI(C, K) = \frac{MI(C, K) - E[MI(C, K)]}{\langle H(C, K) \rangle - E[MI(C, K)]}.$$

AMI gets values in the $[0, 1]$ interval, and when two partitions perfectly match, AMI = 1.

Finally, we consider the Rand Index (RI), initially proposed by Rand [30], which compares two set of clusters while ignoring permutations. Let $\mathcal{C}_{[o,c]}$ and $\mathcal{K}_{[o,c]}$ be two sets of clusters. Let $x(\mathcal{C}_{[o,c]}, \mathcal{K}_{[o,c]})$ be the number of pairs of addresses from the ground truth $\mathcal{A}^{\star}_{[o,c]}$ which are in the same cluster in $\mathcal{C}_{[o,c]}$ and in the same cluster in $\mathcal{K}_{[o,c]}$, i.e.:

$$x(\mathcal{C}_{[o,c]}, \mathcal{K}_{[o,c]}) = |\{(a_1, a_2)|a_1, a_2 \in \mathcal{A}^{\star}_{[o,c]}, a_1 \neq a_2$$
$$\wedge \, \exists \mathcal{C}^{(i)}_{[o,c]} \in \mathcal{C}_{[o,c]} : a_1, a_2 \in \mathcal{C}^{(i)}_{[o,c]}$$
$$\wedge \, \exists \mathcal{K}^{(j)}_{[o,c]} \in \mathcal{K}_{[o,c]} : a_1, a_2 \in \mathcal{K}^{(j)}_{[o,c]}\}$$

and let $y(\mathcal{C}_{[o,c]}, \mathcal{K}_{[o,c]})$ be the number of address pairs from the ground truth $\mathcal{A}^{\star}_{[o,c]}$ which are in different clusters of $\mathcal{C}_{[o,c]}$ and in different clusters of $\mathcal{K}_{[o,c]}$, i.e.:

$$y(\mathcal{C}_{[o,c]}, \mathcal{K}_{[o,c]}) = |\{(a_1, a_2)|a_1, a_2 \in \mathcal{A}^{\star}_{[o,c]}, a_1 \neq a_2$$
$$\wedge \, \exists \mathcal{C}^{(i)}_{[o,c]}, \mathcal{C}^{(j)}_{[o,c]} \in \mathcal{C}_{[o,c]} : a_1 \in \mathcal{C}^{(i)}_{[o,c]}, a_2 \in \mathcal{C}^{(j)}_{[o,c]}, i \neq j$$
$$\wedge \, \exists \mathcal{K}^{(k)}_{[o,c]}, \mathcal{K}^{(l)}_{[o,c]} \in \mathcal{K}_{[o,c]} : a_1 \in \mathcal{K}^{(k)}_{[o,c]}, a_2 \in \mathcal{K}^{(l)}_{[o,c]}, k \neq l\}$$

The Rand Index is defined as:

$$\mathrm{RI}(\mathcal{C}_{[o,c]}, \mathcal{K}_{[o,c]}) = \frac{x(\mathcal{C}_{[o,c]}, \mathcal{K}_{[o,c]}) + y(\mathcal{C}_{[o,c]}, \mathcal{K}_{[o,c]})}{\left|\mathcal{A}^{\star}_{[o,c]}\right| \times \left(\left|\mathcal{A}^{\star}_{[o,c]}\right| - 1\right)},$$

where the denominator is the number of address pairs in $\mathcal{A}^{\star}_{[o,c]}$. As with MI/AMI, we consider an adjusted version of RI, the Adjusted Rand Index (ARI) as proposed by Hubert and Arabie [31], which accounts for chance:

$$\mathrm{ARI}(\mathcal{C}_{[o,c]}, \mathcal{K}_{[o,c]}) = \frac{\mathrm{RI}(\mathcal{C}_{[o,c]}, \mathcal{K}_{[o,c]}) - E[\mathrm{RI}(\mathcal{C}_{[o,c]}, \mathcal{K}_{[o,c]})]}{\max\langle \mathrm{RI}(\mathcal{C}_{[o,c]}, \mathcal{K}_{[o,c]})\rangle - E[\mathrm{RI}(\mathcal{C}_{[o,c]}, \mathcal{K}_{[o,c]})]},$$

where $E[\mathrm{RI}(\mathcal{C}_{[o,c]}, \mathcal{K}_{[o,c]})]$ denotes the expected value of $\mathrm{RI}(\mathcal{C}_{[o,c]}, \mathcal{K}_{[o,c]})$. As for AMI, an ARI value of 1 indicates perfectly matching partitions, while a value of 0 indicates independent partitions. Warrens [32] shows that ARI is equivalent to Cohen's Kappa Cohen [33], which is well suited for the evaluation of community detection methods, as discussed by Liu et al. [34].

4 RESULTS

We first analyze the size of the clusters identified by LPA for the Address Correspondence Networks described in **Section 3**, whose statistics are shown in **Table 1**. **Figure 6** shows the cluster size distribution of $\mathcal{G}_{[11s2,13s1]}$ and $\mathcal{G}_{[15s1]}$, for initialization proportions $p = 0$ and $p = 0.1$. Note that the density of the small clusters, in both cases, shifts to reach larger cluster sizes when $p = 0.1$, as well as the maximum cluster size of $\mathcal{G}_{[11s2,13s1]}$. This indicates that even a small proportion of initialized nodes, such as $p = 0.1$, considerably modifies the cluster distribution in the networks.
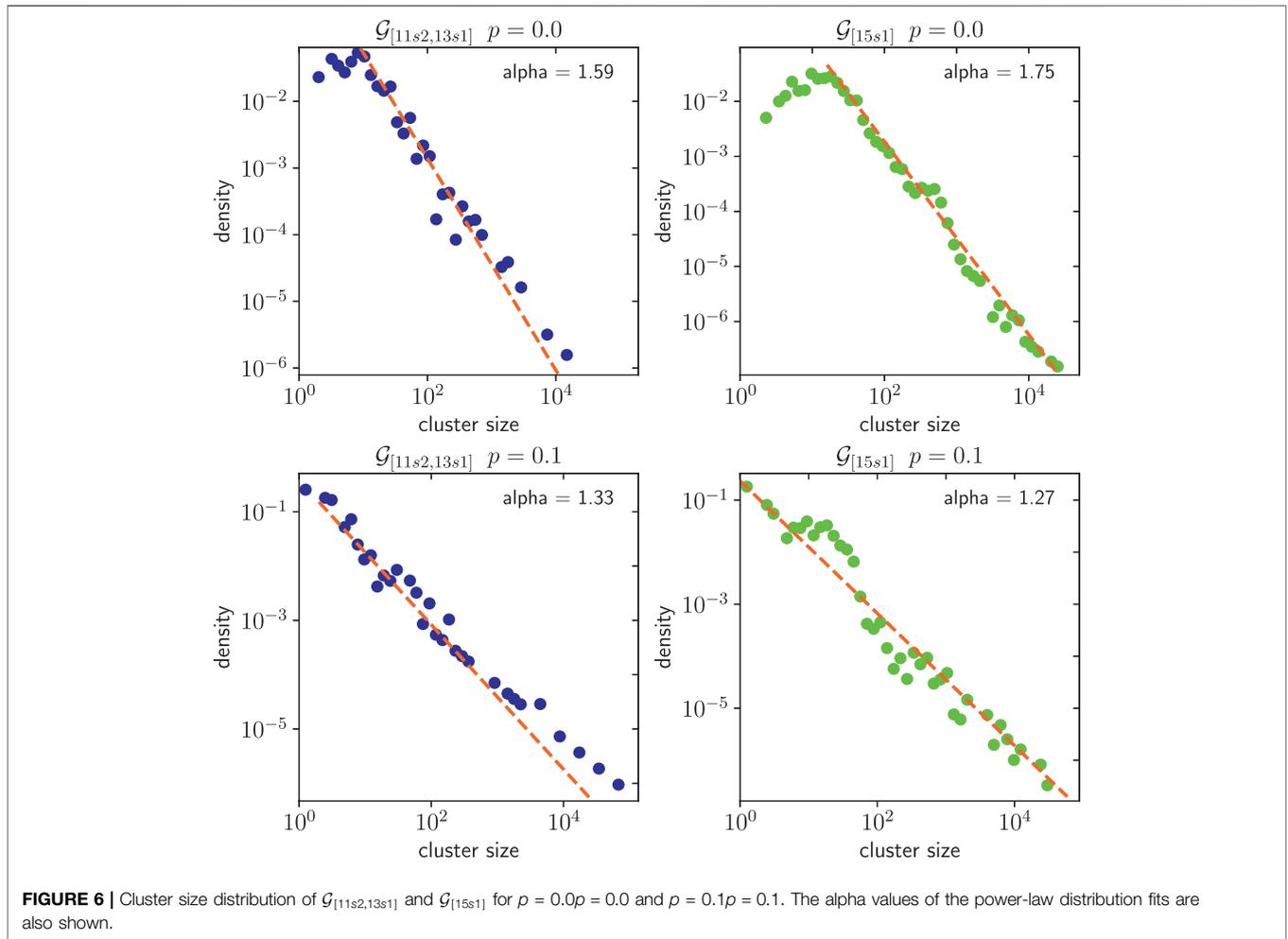
We also fit a power-law distribution to the cluster size distribution, shown by the dotted red lines with the corresponding alpha values in **Figure 6**. Furthermore, the power-law distribution fits the data significantly better than an exponential distribution, resulting in p-values of less than 0.1% using likelihood ratio tests [35]. The exponents are larger for $p = 0$ than for $p = 0.1$, in agreement with the observation related to the range of values in the cluster size. In general, the distributions are very heterogeneous. Additionally, the cluster size distribution suggests that, from a Correspondence Network perspective, there is a preferential attachment dynamic in the address generation where entities that control many addresses are likely to generate more addresses than others.

Next, we study the behavior of the intra-cluster total degree (number of edges connecting nodes that belong to the same cluster) and the inter-cluster degree (number of edges between nodes that belong to different clusters) as functions of the cluster size. For the total intra-cluster degree, there are two extreme behaviors that can be expected. On the one hand, a linear dependency on cluster size would signal that address reuse is negligible (therefore that privacy-preserving usage are commonplace), and the topology of the correspondence network encodes no additional information about the identity of the users that control the addresses. On the other hand, a quadratic relationship (close to the theoretical maximum $\propto c(c-1)/2$) would signal that the clusters are very densely interconnected, and the actual address reuse is high. Therefore, it would be possible to infer actual information about the users by directly inspecting the correspondence network through network science methods. In **Figure 7**, the extreme values of the intra-cluster degree of $\mathcal{G}_{11s2,13s1}$ and $\mathcal{G}_{15s1}$ are above a linear function (red dotted line) and below a quadratic function (yellow dashed line) of the cluster size. The same lines are depicted in the inter-cluster degree distributions showing that the intra-cluster degree grows faster. By applying an Ordinary Least Squares regression (OLS), the slope of a fitting line is in both networks bigger in the intra-cluster case. Furthermore, bigger entities preserve this behavior, showing that the correspondence network has an inherent community structure. Thus, this result is not valid only for entities that control a small number of addresses, and it follows that it is a general property of the network.

Figure 8 shows the number of clusters returned by LPA, $|\mathcal{C}_{[o,c]}|$, as a function of p. The dashed lines indicate the number of entities $|\mathcal{E}^{\star}_{[o,c]}|$ for each Address Correspondence Network. $|\mathcal{E}^{\star}_{[o,c]}|$ is a lower bound of the true number of entities, since each network also contains addresses not in the ground truth. This is supported by $|\mathcal{C}_{[o,c]}| \geq |\mathcal{E}^{\star}_{[o,c]}|$ holding for each test point. In general, $|\mathcal{C}_{[o,c]}|$ decreases sharply at small p, after which the rate of decrease slows and stabilises. $|\mathcal{C}_{[o,c]}|$ tends to be lower for partial networks than for cumulative networks, and can be explained by partial networks having a lower $|\mathcal{E}^{\star}_{[o,c]}|$.

The complexity and structure of the Address Correspondence Network are stable over time: **Figures 9–11** show AMI, ARI and homogeneity as functions of p. Since these metrics require ground truth labels, they are computed only for addresses in $\mathcal{A}^{\star}_{[o,c]}$. We observe that AMI and ARI lead to similar results: they rapidly increase before converging to the maximum value as p increases. In contrast, homogeneity exhibits no such initial rapid increase, and instead increases linearly with p. The mean levels of AMI, ARI and homogeneity do not consistently increase or decrease

FIGURE 6 | Cluster size distribution of $\mathcal{G}_{[11s2,13s1]}$ and $\mathcal{G}_{[15s1]}$ for $p = 0.0p = 0.0$ and $p = 0.1p = 0.1$. The alpha values of the power-law distribution fits are also shown.

with increasing half-year. Furthermore, the mean metric levels for the partial networks appear to be comparable to those for the cumulative networks. This suggests that the complexity and structure of the Address Correspondence Network communities remain stable over time.
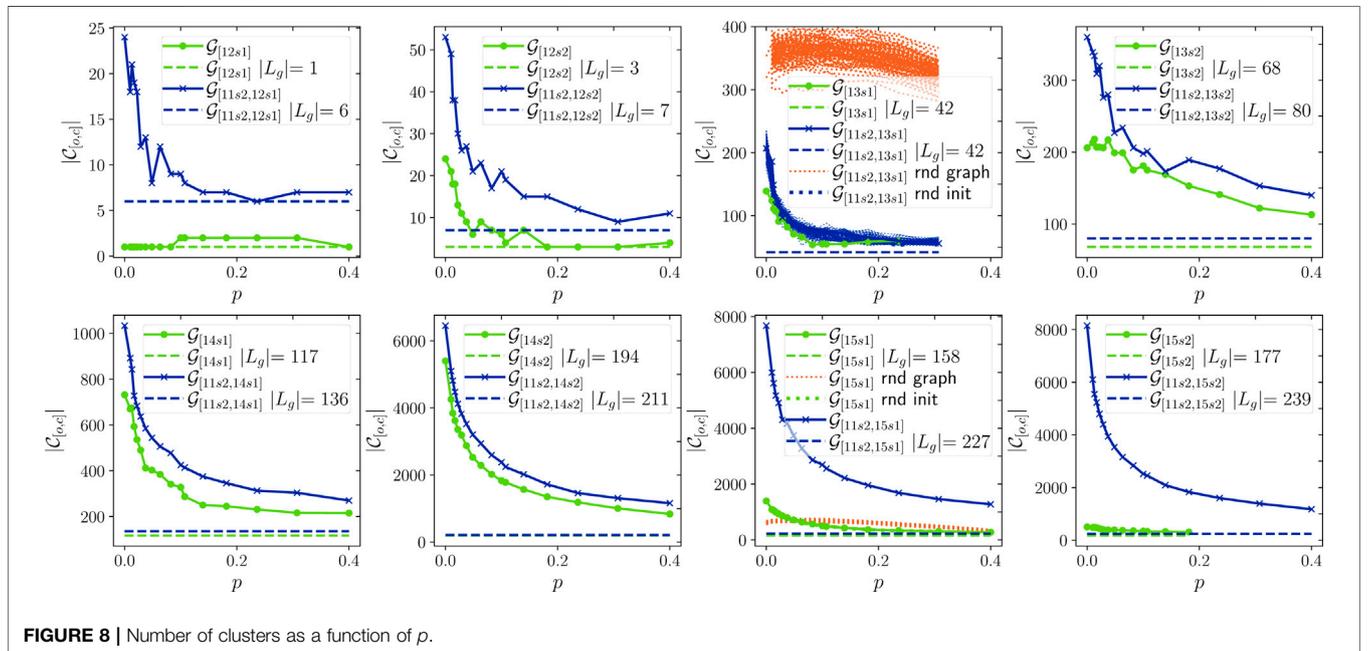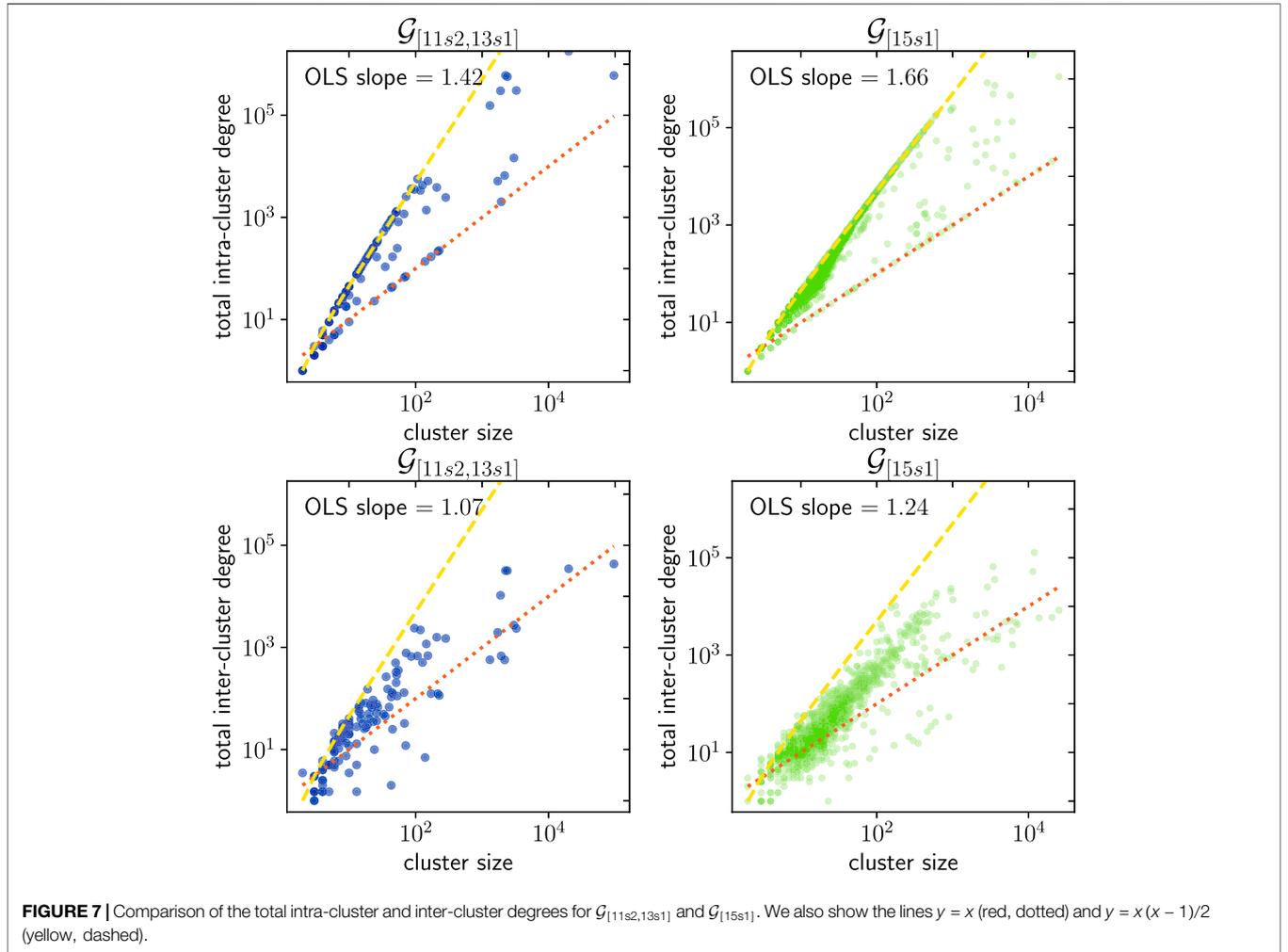
The effect of the node initialization: If the cost of labeling a Bitcoin address is assumed to be constant, the marginal gain in clustering quality per unit cost from increasing p quickly declines. Considering that homogeneity remains constant across all p, it appears that increasing p is cost-effective until around $p = 0.1$. At this point, $\mathcal{A}_{[o,c]}^{I}$ contains most of the information required to describe the community structure. The observed saturations in $|\mathcal{C}_{[o,c]}|$, AMI and ARI suggest that increasing p beyond 0.1 adds only idiosyncratic community information, yielding little improvement in clustering quality. This is further confirmed by studying clustering modularity as a function of p in **Figure 11**. Modularity appears mostly constant except for a sharp initial change, showing a robust community topology that is consistently detected after initializing a small proportion of nodes.
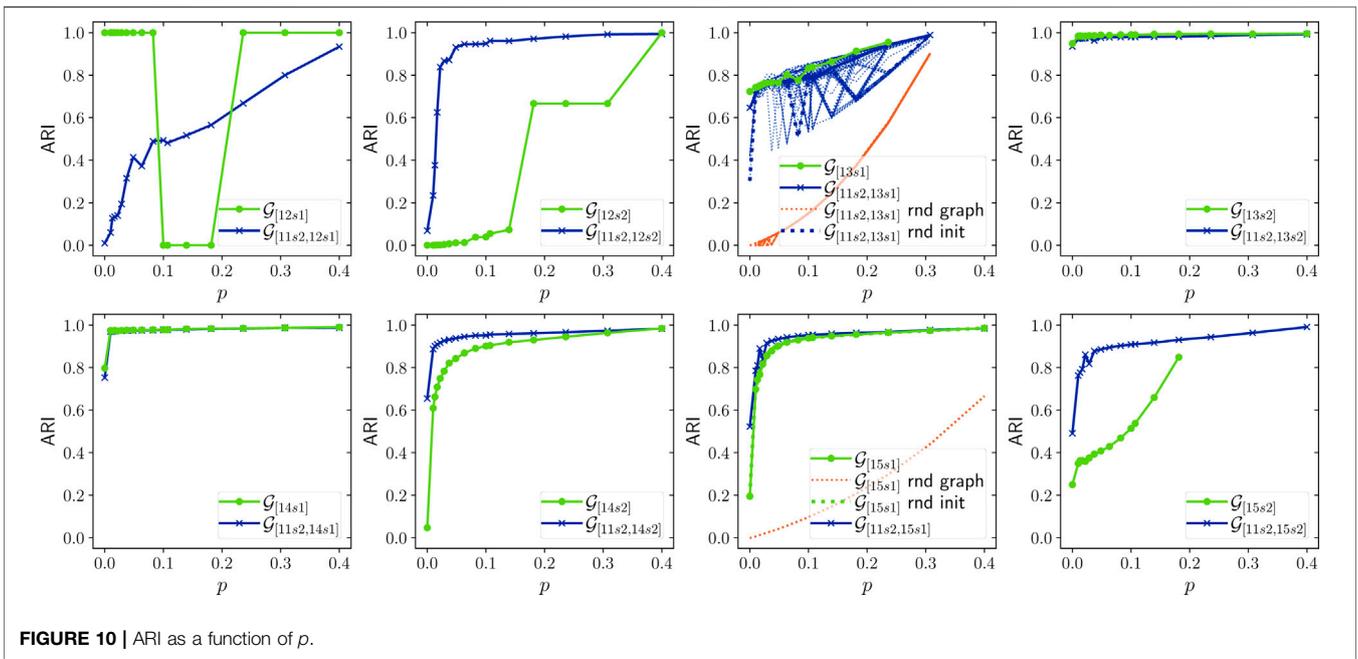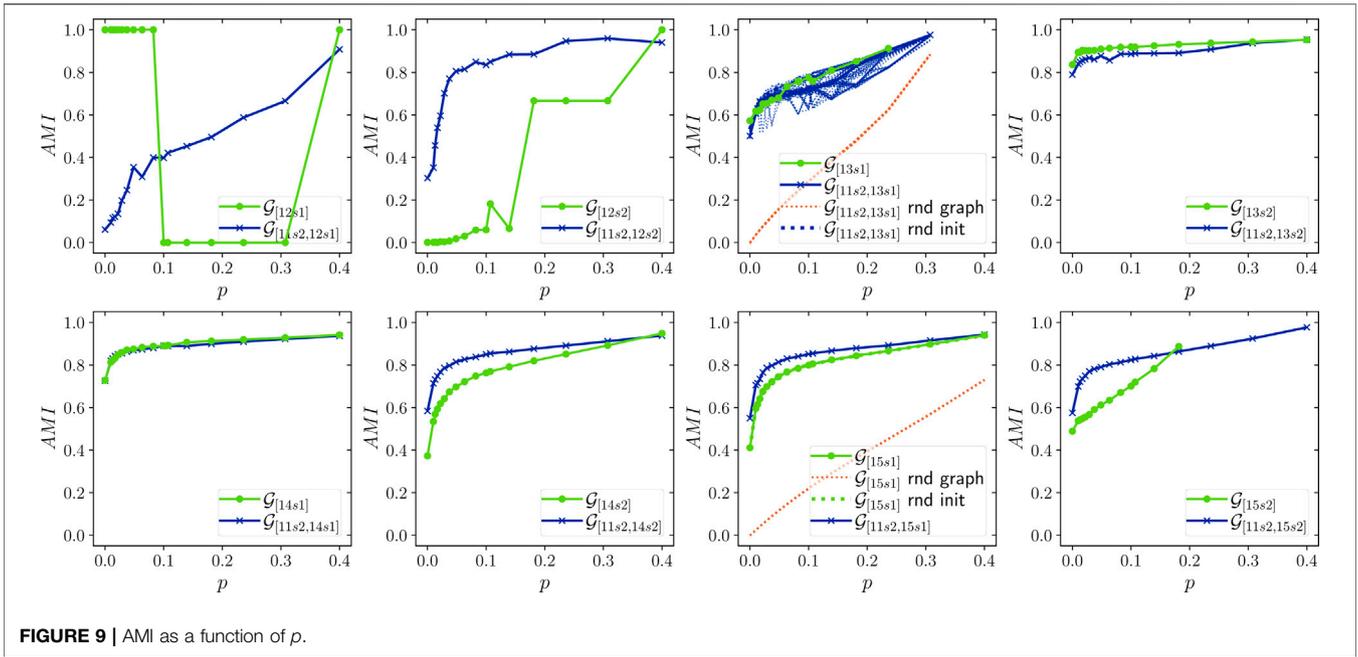
To assert the significance of the results presented in **Figures 8–12**, we repeated the experiments for 100 randomized versions

of the $\mathcal{G}_{[11s2,13s1]}$ and $\mathcal{G}_{[15s1]}$ Address Correspondence Networks. The ii–th randomized network was obtained by performing $4i \cdot |\mathcal{L}_{[o,c]}|$ edge swaps on the original network, according to the algorithm proposed by Maslov [36], which preserves the network's degree distribution. With the exception of $|\mathcal{C}_{[11s2,13s1]}|$ for $\mathcal{G}_{[11s2,13s1]}$, the randomized results show little variation. However, all randomized results appear significantly different to those for the original networks. This suggests that the (non-randomized) results shown in **Figures 8–12** are a consequence of more complex network properties rather than solely the degree distribution.

Furthermore, the effect of node initialization order was studied by repeating the experiments for the $\mathcal{G}_{[11s2,13s1]}$ and $\mathcal{G}_{[15s1]}$ networks using 100 random orderings. The node initialization order does not seem to affect the general level and shape of the curves. Small perturbations observed in **Figures 8–12** appear to be idiosyncrasies of the chosen ordering, and may be larger for smaller networks (since the curves for $\mathcal{G}_{[11s2,13s1]}$ vary more than the ones for $\mathcal{G}_{[15s1]}$).

The effect of cluster and entity sizes: **Figure 13** shows $\hat{H}(E_i)$ and $\hat{H}(C_j)$ for the $\mathcal{G}_{[14s1]}$, $\mathcal{G}_{[11s2,14s1]}$, $\mathcal{G}_{[15s2]}$ and $\mathcal{G}_{[11s2,15s2]}$ networks. $\hat{H}(E_i)$ and $\hat{H}(C_j)$ are expressed as functions of the

FIGURE 7 | Comparison of the total intra-cluster and inter-cluster degrees for $\mathcal{G}_{[11s2,13s1]}$ and $\mathcal{G}_{[15s1]}$. We also show the lines $y = x$ (red, dotted) and $y = x(x-1)/2$ (yellow, dashed).

FIGURE 8 | Number of clusters as a function of p.

FIGURE 9 | AMI as a function of p.

FIGURE 10 | ARI as a function of p.

relative cluster and entity sizes, i.e. normalized to $\left|\mathcal{A}_{[o,c]}^{\star}\right|$, respectively. We run experiments with $p = 0$ and $p = 0.1$. We note that $\widehat{H}(E_i)$ correlates negatively with the relative cluster size, and $\widehat{H}(C_j)$ correlates negatively with relative entity size. For small clusters and entities, there are strips of points located at the minimum and maximum values of $\widehat{H}(E_i)$ and $\widehat{H}(C_j)$. This is to be expected: if we consider a cluster with only two addresses, both associated with the same entity, $\widehat{H}(E_i)$ is minimum. If two addresses are mapped to different entities, we obtain a

uniform entity label distribution, and $\widehat{H}(E_i)$ is maximum. Such extreme fluctuations become less likely as cluster size increases. Large clusters, therefore, tend to be purer than smaller clusters, corresponding to a higher clustering quality. Similarly, entities represented by more addresses are distributed more asymmetrically across clusters, again corresponding to a higher clustering quality. This is in agreement with the results in **Figure 7**, where the community structure is shown to become more apparent for larger clusters.

FIGURE 11 | Homogeneity as a function of p.

FIGURE 12 | Modularity as a function of p.
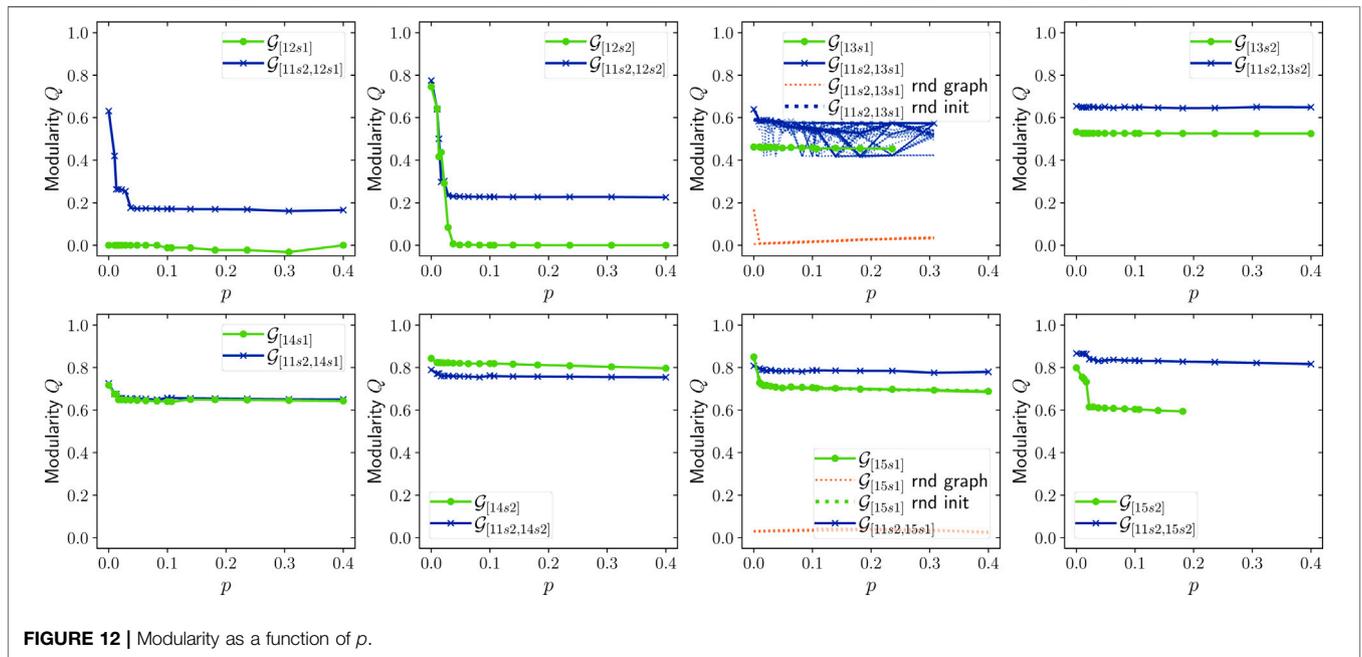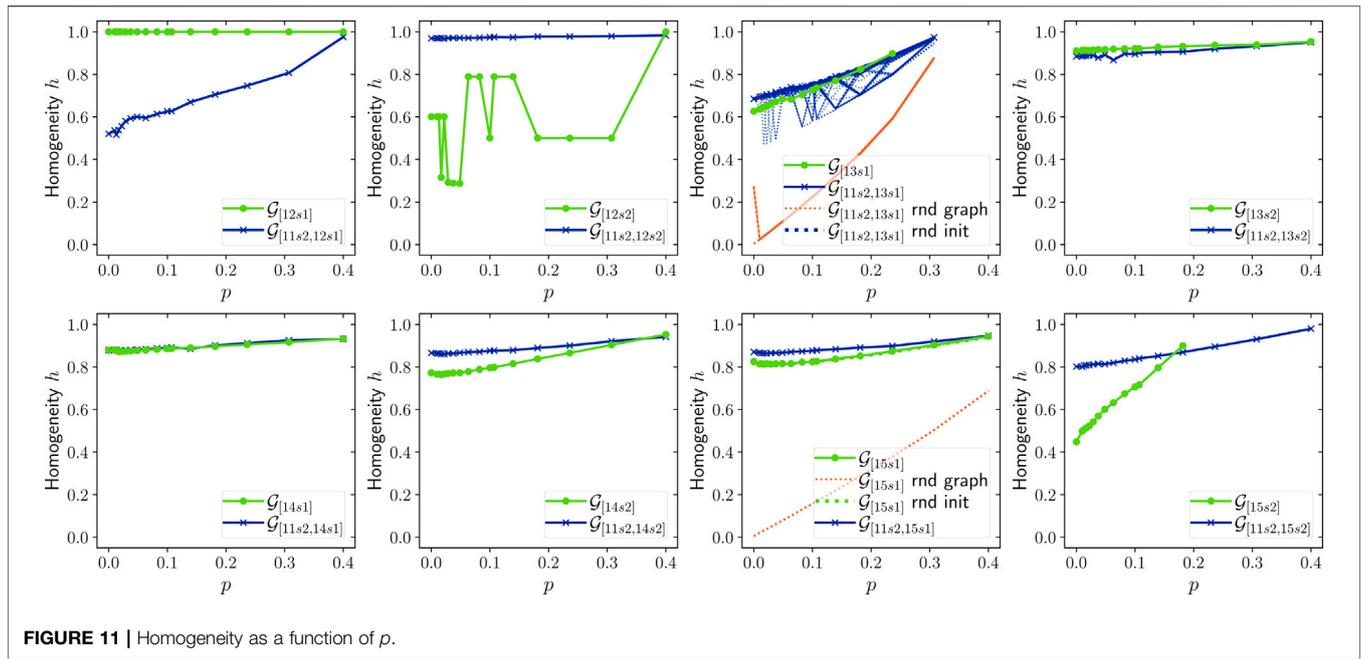
Furthermore, the mean levels of $\hat{H}(E_i)$ and $\hat{H}(C_j)$ for the partial networks are always less than or equal to the ones of the corresponding cumulative networks (comparing row 1–3 and row 2–4 in **Figure 13**). This suggests that partial networks allow a higher quality of interpretation regarding the community structure. A possible explanation for this is that Bitcoin entities have less time to obfuscate their activity: the longer the considered transaction history, the more the obfuscation

attempts accumulate and the more difficult it becomes to detect the true community structure.

Interestingly, the average $\hat{H}(E_i)$ and $\hat{H}(C_j)$ increase after initialising 10% of nodes. The increase in $\hat{H}(E_i)$ can be explained by the loss of small, homogeneous clusters with low $\hat{H}(E_i)$. For $\hat{H}(C_j)$, the increase is likely due to the decrease in the number of clusters, which in turn causes $H_{max}(E_i)$ to decrease.
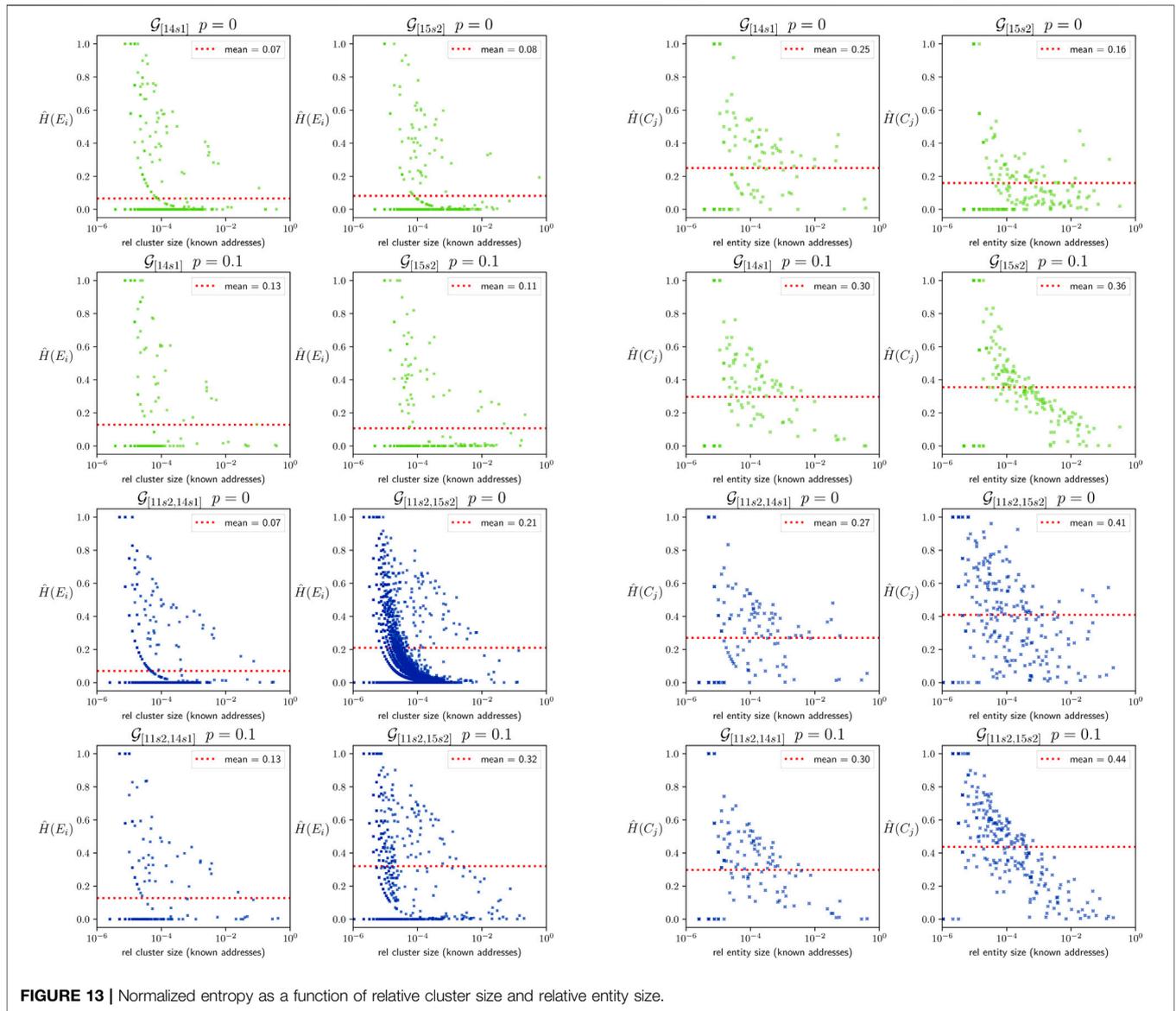
FIGURE 13 | Normalized entropy as a function of relative cluster size and relative entity size.

5 CONCLUSION AND FUTURE WORK

In this paper, we consider the application of a general-purpose community detection algorithm, LPA, to detect address clusters that are controlled by the same entity in the Bitcoin transaction history. Specifically, we apply LPA to Address Correspondence Networks, which incorporate information from a variety of simple address linking heuristics. We detect a strong community structure within these networks by inspecting their intra- and inter-cluster degrees. We find that the inter-cluster degree grows faster than the inter-cluster degree for cluster size increments. Address correspondence networks are therefore suitable for the application of general community detection methods from the broader field of network science—this

creates an entry point for future researchers to move far beyond the application of primitive heuristics.

Since LPA is able to exploit ground truth information, we find that clustering quality improves as the number of labeled addresses in the Address Correspondence Networks increases. However, under the assumption that the cost of labeling a Bitcoin address is constant, we find that the marginal gain in clustering quality per unit cost quickly declines. Under this assumption, we propose that address labeling is cost-effective until around $p = 0.1$ $p = 0.1$, i.e. until 10% of all addresses in the Address Correspondence Network are identified. Furthermore, we find that choosing which addresses to label does not have a significant effect on clustering quality. Finally, we find that the structure of communities in the Address Correspondence Network remains stable over time. Partial Address Correspondence Networks are,

therefore, reasonable proxies for their cumulative counterparts (and far less demanding from a computational point of view).

For future work, we plan to conduct experiments to test the robustness of the heuristics and specific combinations between them. For example, analyzing their likelihood and studying their contribution to the links between addresses. From a network reconstruction perspective, link prediction is an interesting approach to improve the correspondence network by validating current links and predicting missing ones. Additionally, different machine learning approaches can be implemented to graph analysis; supervised methods are suitable if more ground truth information is available in the future.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: https://bitcoin.org/en/download, https://www.walletexplorer.com/.

REFERENCES

1. Foley S, Karlsen JR, and Putniņš TJ. Sex, Drugs, and Bitcoin: How Much Illegal Activity Is Financed through Cryptocurrencies? *Rev Financial Stud* (2019) 32: 1798–853. doi:10.1093/rfs/hhz015

2. Gaihre A, Luo Y, and Liu H. Do Bitcoin Users Really Care about Anonymity? an Analysis of the Bitcoin Transaction Graph. In: *2018 IEEE International Conference on Big Data (Big Data)* (2018). p. 1198–207. doi:10.1109/BigData.2018.8622442

3. Meiklejohn S, Pomarole M, Jordan G, Levchenko K, McCoy D, Voelker GM, et al. A Fistful of Bitcoins. *Commun ACM* (2016) 59:86–93. doi:10.1145/2896384

4. Kondor D, Pósfai M, Csabai I, and Vattay G. Do the Rich Get Richer? an Empirical Analysis of the Bitcoin Transaction Network. *PLoS ONE* (2014) 9: e86197. doi:10.1371/journal.pone.0086197

5. Javarone MA, and Wright CS. From Bitcoin to Bitcoin Cash. *Proc 1st Workshop Cryptocurrencies Blockchains Distributed Syst* (2018):77–81. doi:10.1145/3211933.3211947

6. Vallarano N, Tessone CJ, and Squartini T. Bitcoin Transaction Networks: An Overview of Recent Results. *Front Phys* (2020) 8:286. doi:10.3389/fphy.2020.00286

7. Bovet A, Campajola C, Mottes F, Restocchi V, Vallarano N, Squartini T, et al. The Evolving Liaisons between the Transaction Networks of Bitcoin and its price Dynamics. *arXiv:1907.03577 [physics, q-fin] ArXiv* (2019) 1907:03577.

8. Nakamoto S. *Bitcoin: A Peer-To-Peer Electronic Cash System* (2008). Available at SSRN: https://ssrn.com/abstract=3440802

9. Kalodner H, Goldfeder S, Chator A, Möser M, and Narayanan A. BlockSci: Design and Applications of a Blockchain Analysis Platform. *arXiv:1709.02489 [cs] ArXiv* (2017) 1709:02489.

10. Nick JD. *Data-Driven De-anonymization in Bitcoin. Tech. Rep.* Zurich: ETH Zurich (2015).

11. Harrigan M, and Fretter C. "The Unreasonable Effectiveness of Address Clustering". In 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld) (2016), 368–373. doi:10.1109/UIC-ATC-ScalCom-CBDCom-IoP-SmartWorld.2016.0071

12. Fleder M, Kester MS, and Pillai S. Bitcoin Transaction Graph Analysis. *arXiv: 1502.01657 [cs] ArXiv* (2015) 1502:01657.

13. Zhang Y, Wang J, and Luo J. Heuristic-Based Address Clustering in Bitcoin. *IEEE Access* (2020) 8:210582–91. doi:10.1109/ACCESS.2020.3039570

AUTHOR CONTRIBUTIONS

JF and AP developed the software, curated the data, run the analyses, created the visualizations, and wrote the initial draft. DD and CT contributed with the conceptualization and methodology, supervised the study and reviewed and edited the text. AB supervised the study and reviewed and edited the text. All authors discussed the results. All authors worked and agreed on the final version.

FUNDING

14. Patel Y. *Deanonymizing Bitcoin Transactions an Investigative Study on Large-Scale Graph Clustering*. Princeton University Senior Theses, Princeton University (2018).

15. Ermilov D, Panov M, and Yanovich Y. Automatic Bitcoin Address Clustering. In: *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. Mexico: Cancun IEEE (2017). p. 461–6. doi:10.1109/ICMLA.2017.0-118

16. Biryukov A, and Tikhomirov S. Deanonymization and Linkability of Cryptocurrency Transactions Based on Network Analysis. In: *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. Stockholm, Sweden: IEEE (2019). p. 172–84. doi:10.1109/EuroSP.2019.00022

17. Harlev MA, Sun Yin H, Langenheldt KC, Mukkamala RR, and Vatrapu R. Breaking Bad: De-anonymising Entity Types on the Bitcoin Blockchain Using Supervised Machine Learning. In: *Proceedings of the 51st Hawaii International Conference on System Sciences 2018*. United States: Hawaii International Conference on System Sciences (HICSS) (2018). p. 3497–506. Proceedings of the Annual Hawaii International Conference on System Sciences.

18. [Dataset] Janda A. *WalletExplorer.com: Smart Bitcoin Block Explorer* (2017).

19. Barabási A-L, and Bonabeau E. Scale-Free Networks. *Sci Am* (2003) 288:60–9. doi:10.1038/scientificamerican0503-60

20. Raghavan UN, Albert R, and Kumara S. Near Linear Time Algorithm to Detect Community Structures in Large-Scale Networks. *Phys Rev E* (2007) 76:036106. doi:10.1103/PhysRevE.76.036106

21. Yang Z, Algesheimer R, and Tessone CJ. A Comparative Analysis of Community Detection Algorithms on Artificial Networks. *Sci Rep* (2016) 6: 30750. doi:10.1038/srep30750

22. Newman MEJ. Finding Community Structure in Networks Using the Eigenvectors of Matrices. *Phys Rev E* (2006) 74:036104. doi:10.1103/PhysRevE.74.036104

23. Pons P, and Latapy M. Computing Communities in Large Networks Using Random Walks. *Jgaa* (2006) 10:191–218. doi:10.7155/jgaa.00124

24. Blondel VD, Guillaume J-L, Lambiotte R, and Lefebvre E. Fast Unfolding of Communities in Large Networks. *J Stat Mech* (2008) 2008:P10008. doi:10.1088/1742-5468/2008/10/p10008

25. Newman MEJ, and Girvan M. Finding and Evaluating Community Structure in Networks. *Phys Rev E* (2004) 69. doi:10.1103/physreve.69.026113

26. Shannon CE. A Mathematical Theory of Communication. *Bell Syst Tech J* (1948) 27:379–423. doi:10.1002/j.1538-7305.1948.tb01338.x

27. Rosenberg A, and Hirschberg J. V-measure: A Conditional Entropy-Based External Cluster Evaluation Measure. In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational Linguistics (2007). p. 410–20.

28. Cover TM, and Thomas JA. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. USA: Wiley-Interscience (2006).

29. Vinh NX, Epps J, and Bailey J. Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. *J Machine Learn Res* (2010) 11:2837–54.

30. Rand WM. Objective Criteria for the Evaluation of Clustering Methods. *J Am Stat Assoc* (1971) 66:846–50. doi:10.1080/01621459.1971.10482356

31. Hubert L, and Arabie P. Comparing Partitions. *J Classification* (1985) 2: 193–218. doi:10.1007/BF01908075

32. Warrens MJ. On the Equivalence of Cohen's Kappa and the Hubert-Arabie Adjusted Rand Index. *J Classif* (2008) 25:177–83. doi:10.1007/s00357-008-9023-7

33. Cohen J. A Coefficient of Agreement for Nominal Scales. *Educ Psychol Meas* (1960) 20:37–46. doi:10.1177/001316446002000104

34. Liu X, Cheng H-M, and Zhang Z-Y. Evaluation of Community Detection Methods. *IEEE Trans Knowl Data Eng* (2019) 32:1. doi:10.1109/TKDE.2019.2911943

35. Clauset A, Shalizi CR, and Newman MEJ. Power-Law Distributions in Empirical Data. *SIAM Rev* (2009) 51:661–703. doi:10.1137/070710111

36. Maslov S. Specificity and Stability in Topology of Protein Networks. *Science* (2002) 296:910–3. doi:10.1126/science.1065103

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.