Bitcoin Transactions Network

Questions asked:

Is it possible to represent bitcoin transactions as a network?

Can you infer ownership of bitcoin addresses by analyzing transactions?

Is the bitcoin transactions network a scale-free network?

Gathering Data

The first task that was tackled in this project was data acquisition.

The bitcoin blockchain is by definition a huge database containing all transaction that were ever validated by the blockchain. Therefore, if we wish to gather data about transactions, we must query the blockchain for it.

Data Integrity

To guarantee the integrity of the data, I decided to install the reference bitcoin node implementation, known as bitcoin-core, and to run a full node, in order to be able to directly query other nodes for blockchain data.

Bitcoin-core is a open-source version of the bitcoin software, and currently it’s the most commonly used software by other nodes. If a node does not run bitcoin-core , it probably runs a modified version of it.

The software is available on github at

https://github.com/bitcoin/bitcoin

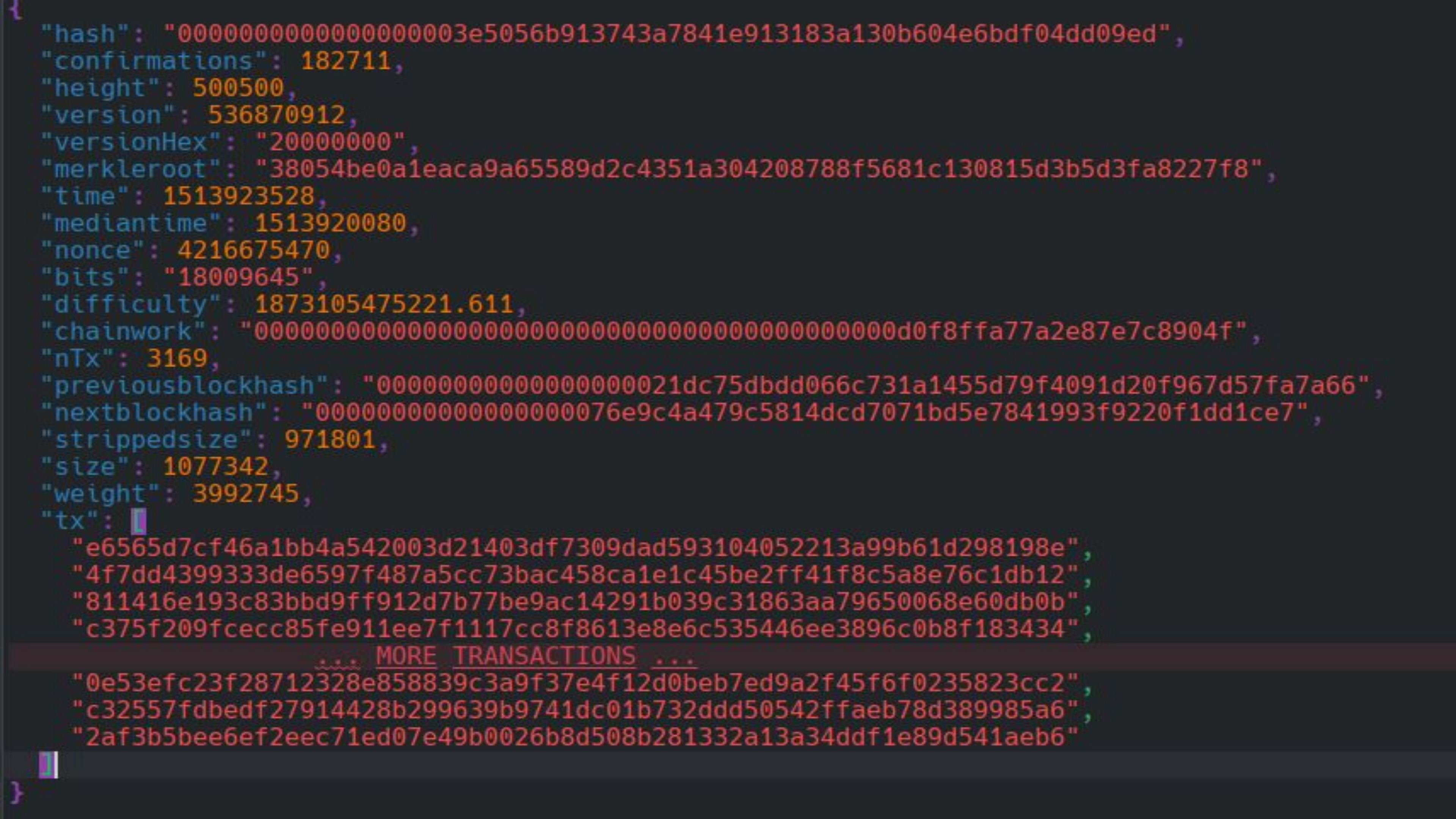
Once the software is installed and running, it will begin a (very slow) process of finding and querying other nodes for blockchain data. If one does not have a dedicated connection to a full node that is able to provide blockchain data, bitcoin-core will search for nodes at random.

This process might take a very long time.

This is crucial because in order for the blockchain to be accepted and for the data to be reliable, every node must process each block and validate it independently. Only by doing this we can be sure that the data we have is reliable and that it represents correctly the transactions that have occurred on the blockchain.

Data Acquisition

Once bitcoin-core is running, it’s possible to query other nodes for blockchain data. A block is returned by the blockchain as a JSON object containing information about the block, such as height, hash, time it was mined, and most importantly, a list of transactions contained in that block



```
{
  "hash": "00000000000000000003e5056b913743a7841e913183a130b604e6bdf04dd09ed",
  "confirmations": 182711,
  "height": 500500,
  "version": 536870912,
  "versionHex": "20000000",
  "merkleroot": "38054be0a1eaca9a65589d2c4351a304208788f5681c130815d3b5d3fa8227f8",
  "time": 1513923528,
  "mediantime": 1513920080,
  "nonce": 4216675470,
  "bits": "18009645",
  "difficulty": 1873105475221.611,
  "chainwork": "000000000000000000000000000000000000000000d0f8ffa77a2e87e7c8904f",
  "nTx": 3169,
  "previousblockhash": "00000000000000000021dc75dbdd066c731a1455d79f4091d20f967d57fa7a66",
  "nextblockhash": "00000000000000000076e9c4a479c5814dcd7071bd5e7841993f9220f1dd1ce7",
  "strippedsize": 971801,
  "size": 1077342,
  "weight": 3992745,
  "tx": [
    "e6565d7cf46a1bb4a542003d21403df7309dad593104052213a99b61d298198e",
    "4f7dd4399333de6597f487a5cc73bac458ca1e1c45be2ff41f8c5a8e76c1db12",
    "811416e193c83bbd9ff912d7b77be9ac14291b039c31863aa79650068e60db0b",
    "c375f209fcecc85fe911ee7f1117cc8f8613e8e6c535446ee3896c0b8f183434",
                      ... MORE TRANSACTIONS ...
    "0e53efc23f28712328e858839c3a9f37e4f12d0beb7ed9a2f45f6f0235823cc2",
    "c32557fdbedf27914428b299639b9741dc01b732ddd50542ffaeb78d389985a6",
    "2af3b5bee6ef2eec71ed07e49b0026b8d508b281332a13a34ddf1e89d541aeb6"
  ]
}
```

First big problem

While querying for blocks does not take a substantial amount of time, querying the blockchain for transactions is a lot slower. This happens if your node has not yet validated the entire blockchain. Moreover, each block contains thousands of transactions.
This is why i decided that querying for transaction data had to be done through a websocket API.

Gathering block data from the blockchain and transaction data through the API allowed me to develop a simple and moderately fast pipeline for data extraction.

Typical transaction hex-dump

Each transaction comes in a raw hex-like format, like so:

```
02000000013066b6fd51e0aea6e456aac1b894a076a50634a1517c3c81219acee3
41892a600f0000006a47304402200223ed42107279f13dc4e35af13a235557b1aa7
b8c0ba5125026d47310241b1d0220081bd0ff5ee2629cd1e51d8aef120170df2421
682e580a641efd7c6b26618d3a012102d46e0710b40c285aec6c9e3d0271d13be5
2acb216f8c5b81d8875abcd5e35860ffffffff012681d006000000001976a91442cdfb0
3341f83724902dc14f159f489d79beb0488ac00000000
```

This format encodes all the information stored in a transaction.

Edge extraction

From the hex dump we extract the information we need for our networks, such as:

- Transaction id
- Sender(s) address
- Receiver(s) address
- Amount sent to each receiver

After extracting edge information, we create csv files for each block. This way we are able to feed R with our data and plot our networks

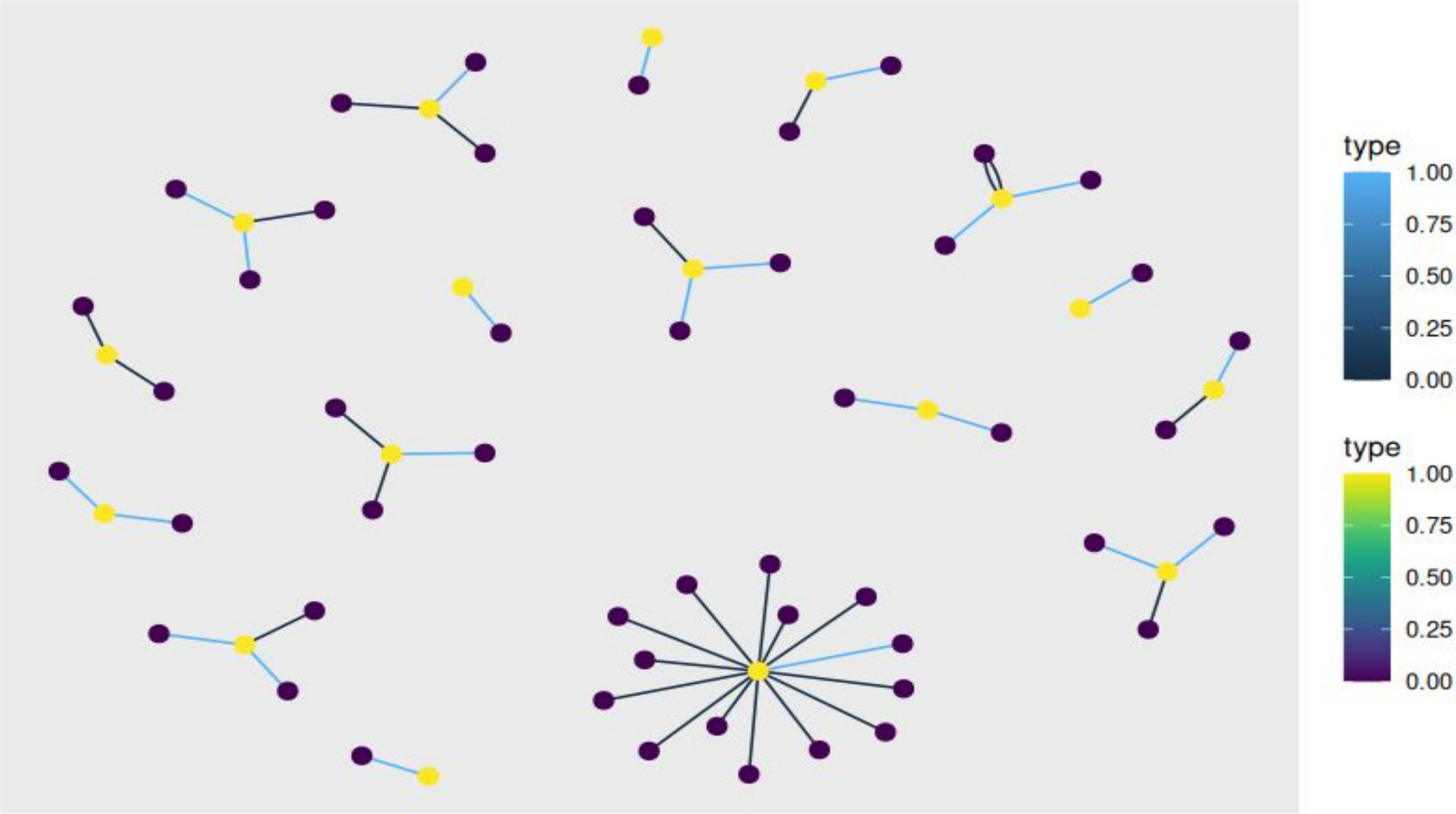
Transaction network

A transaction network is composed of two different kind of nodes:

Addresses and Transaction

Addresses can function both as input or output to a transaction. Yellow nodes are transactions, and purple nodes are addresses. There are two different kind of edges. From address to transaction (input, type 0) and from transaction to address (output, type 1).

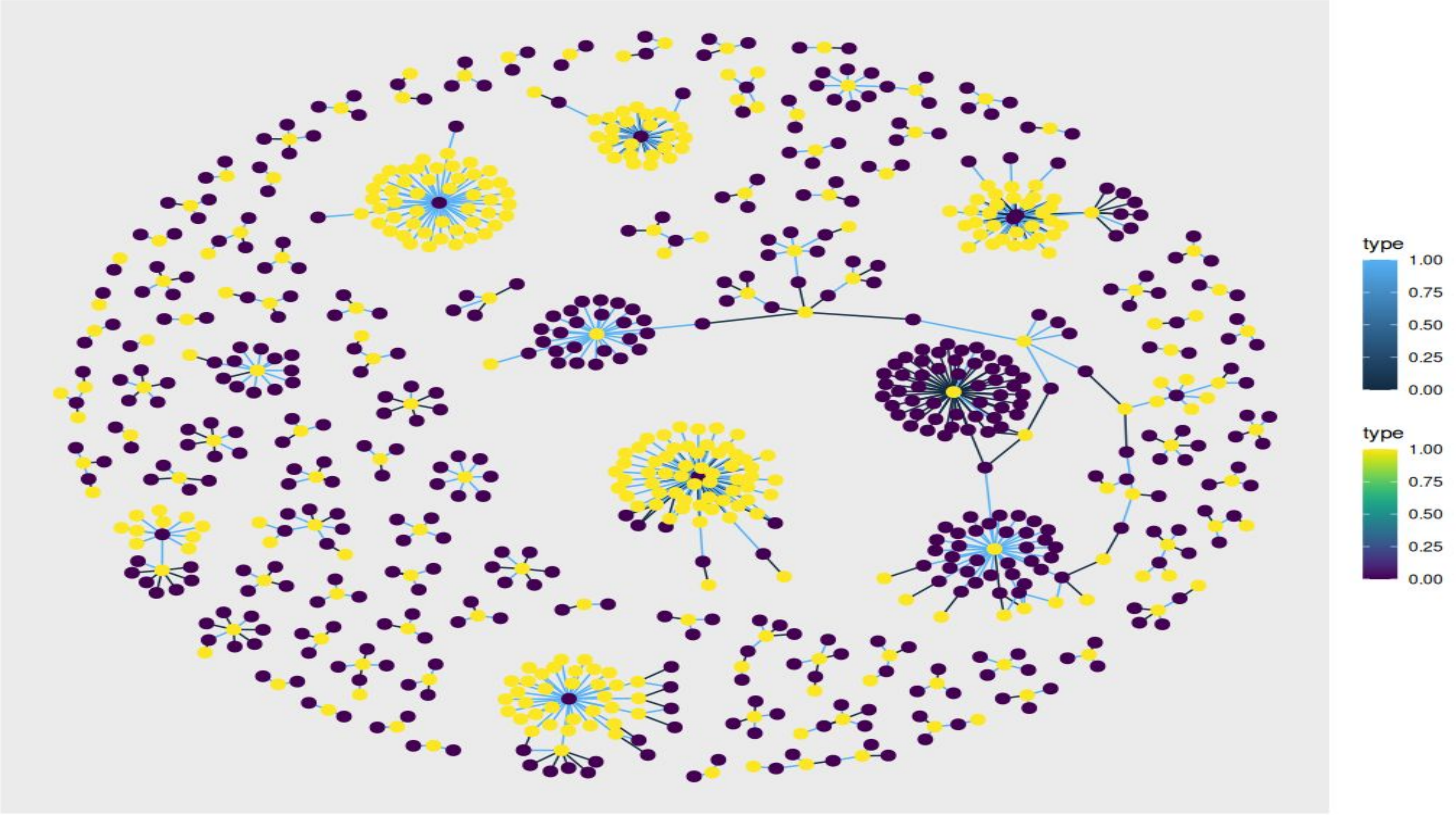
A common subset of transactions in a block might look like this

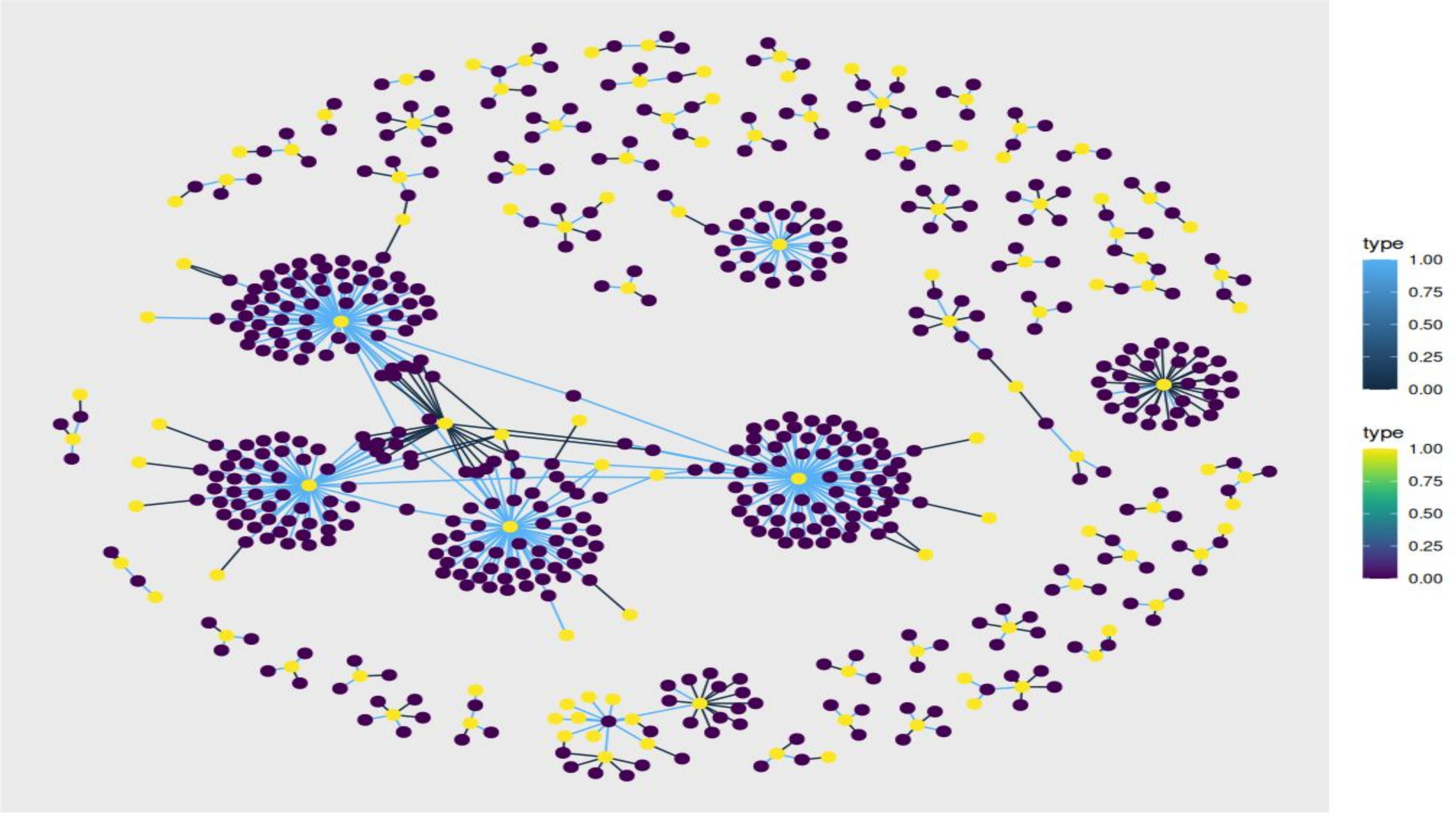

type
1.00
0.75
0.50
0.25
0.00
type
1.00
0.75
0.50
0.25
0.00

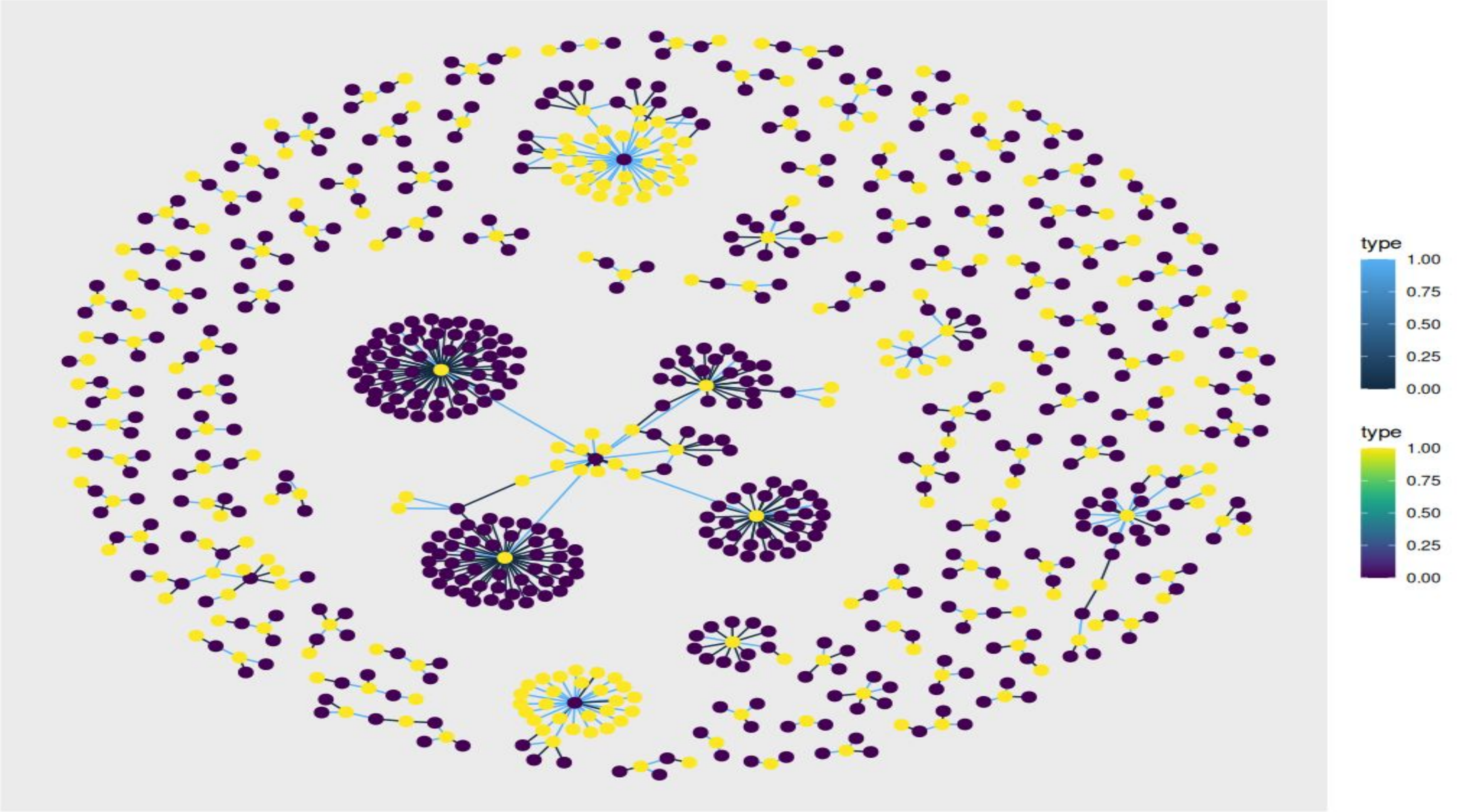
Searching the Connected Component

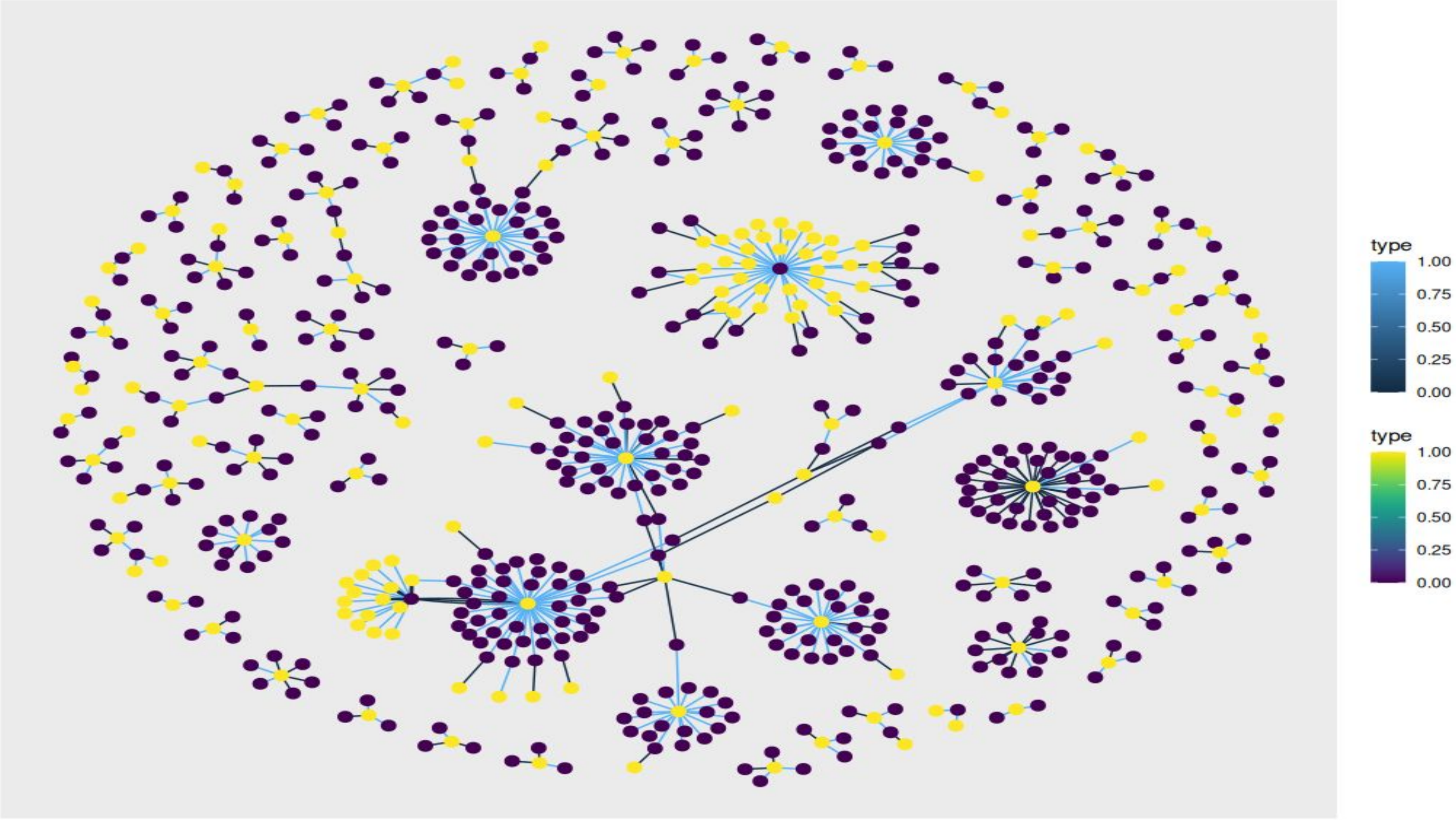
What we are looking for are large connected components inside a graph of a full block (or possibly more blocks). Once we try to plot more than 500 transactions, visualization becomes hard (and also meaningless), thus the approach i took was to extract a subset (or a slice) of unique nodes from the edgelist, and then extract again from the whole graph all edges containing at least one of the unique nodes from our “slice”. In this way we are able to determine if one node is involved in more than one transaction, or is part of a larger connected component of other nodes.

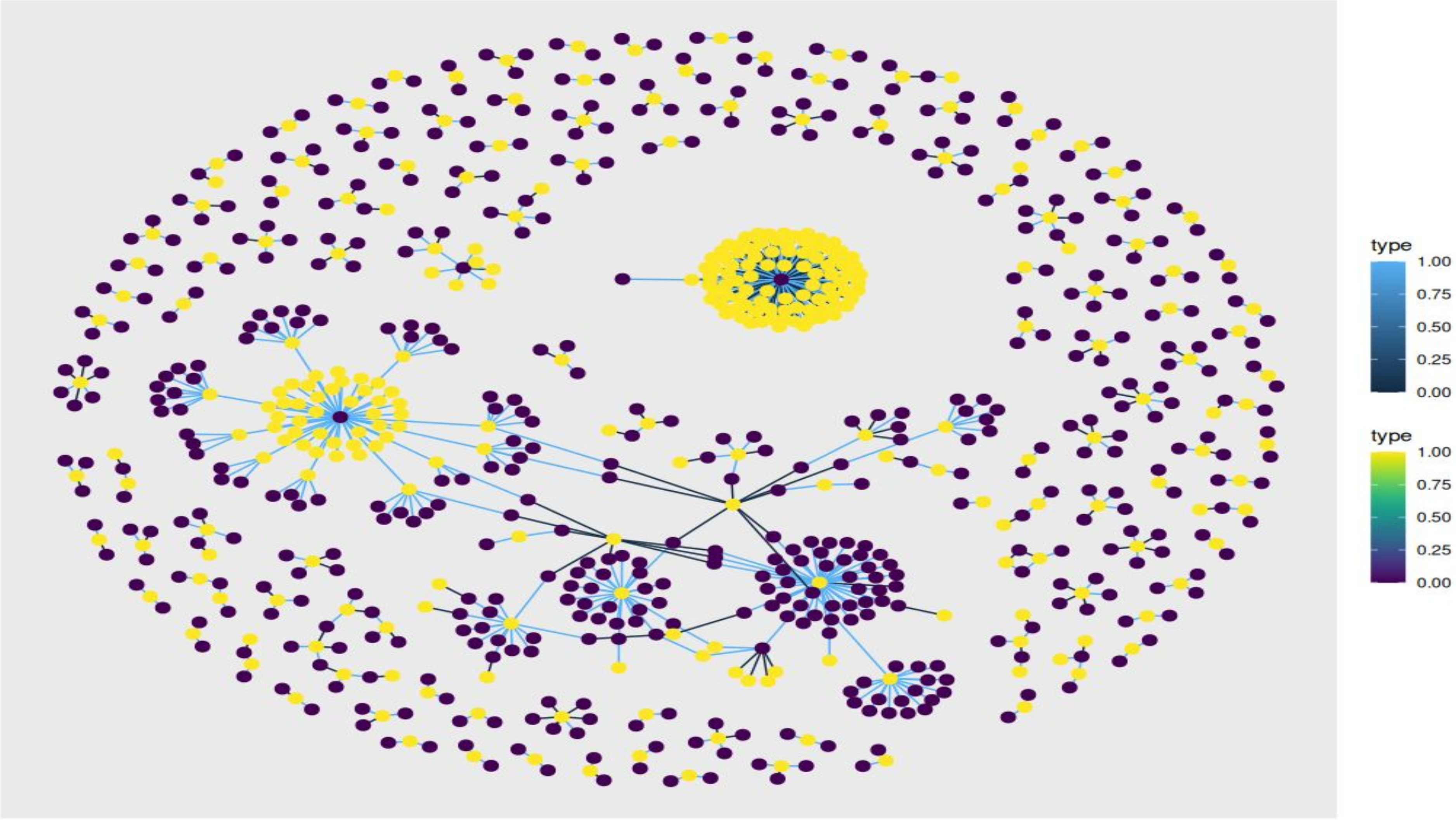
Here are some of the results achieved with this method. Here are some networks plotted with this method

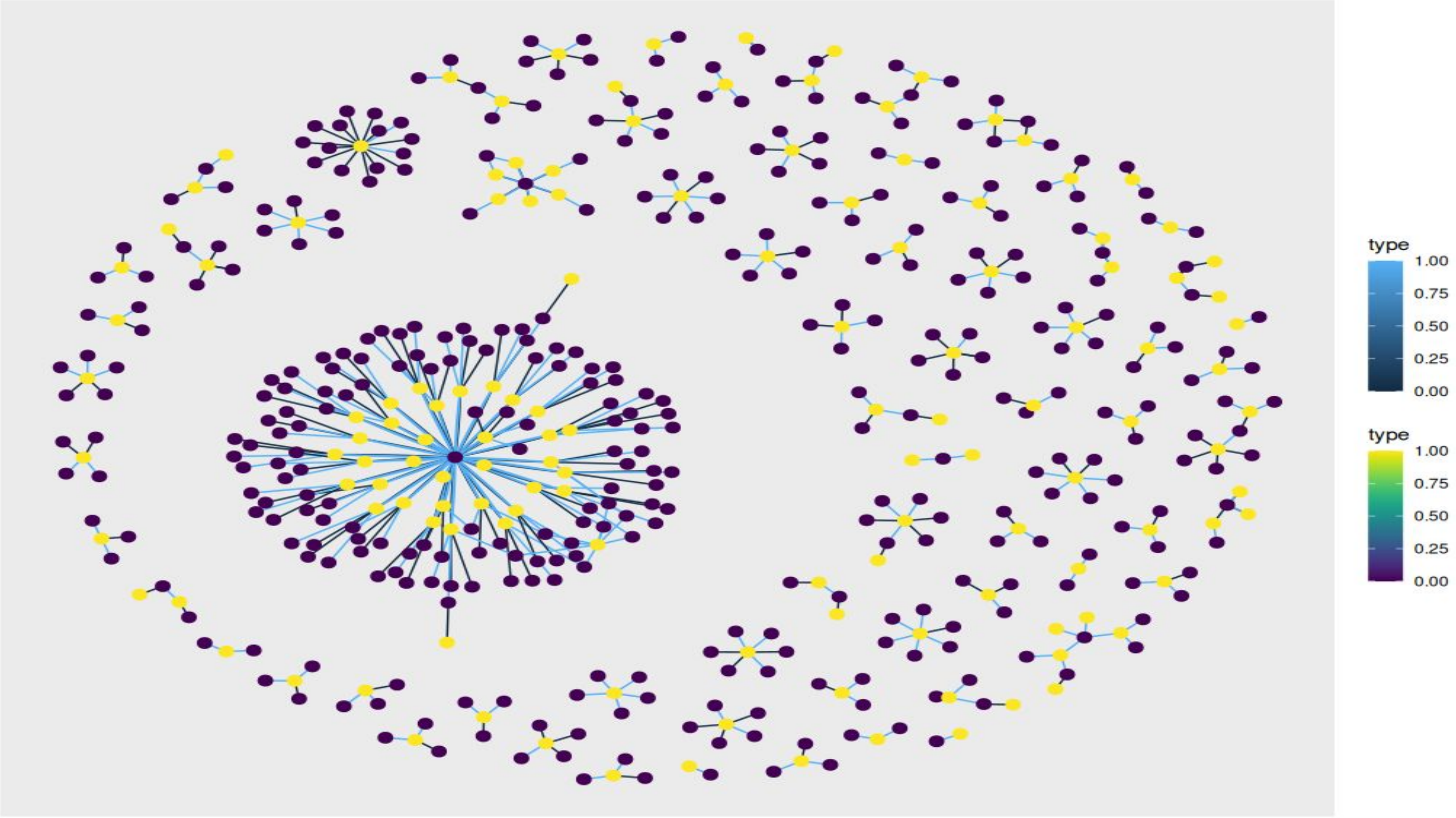

type
1.00
0.75
0.50
0.25
0.00
type
1.00
0.75
0.50
0.25
0.00

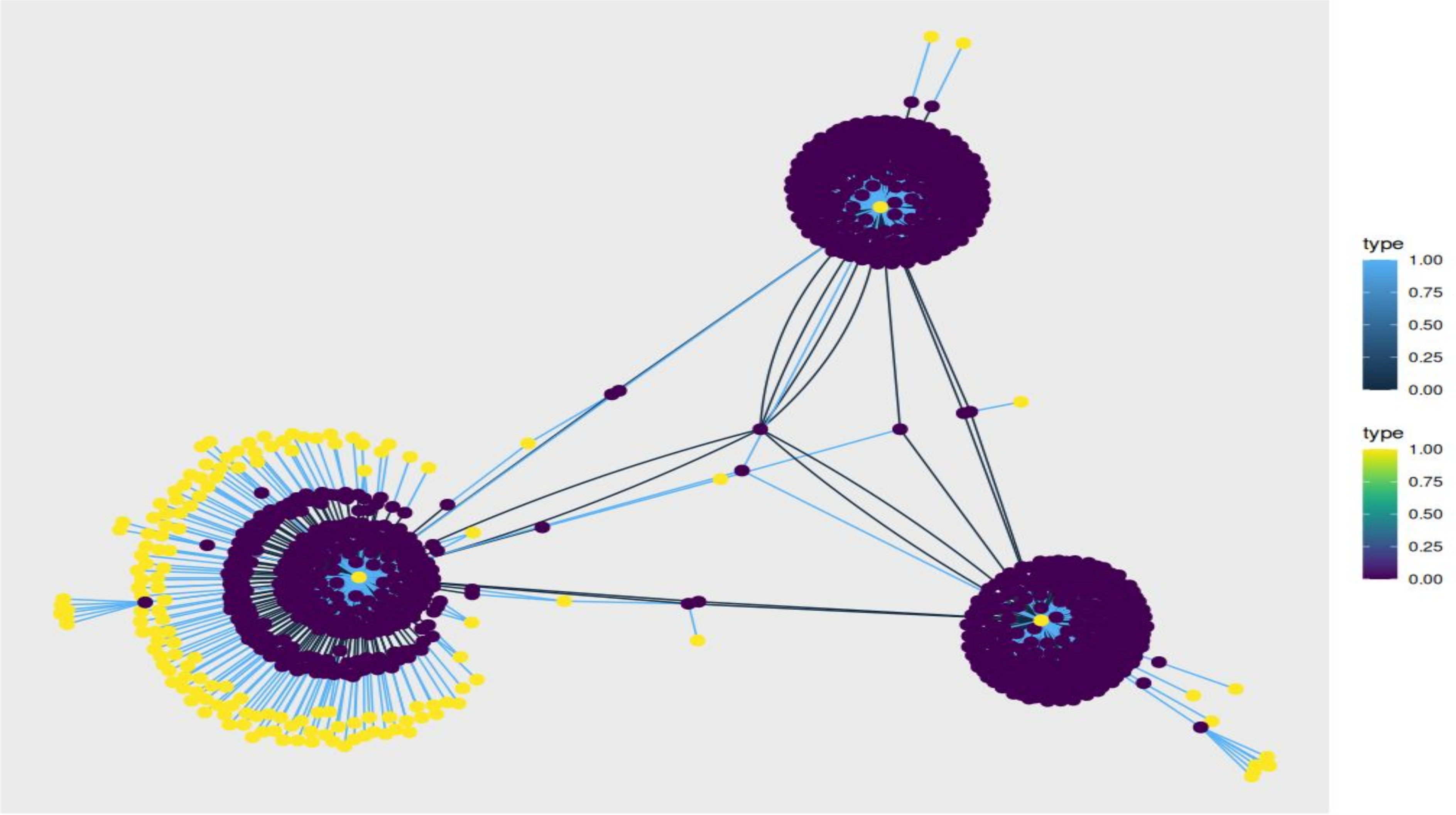

type
1.00
0.75
0.50
0.25
0.00
type
1.00
0.75
0.50
0.25
0.00


type
1.00
0.75
0.50
0.25
0.00
type
1.00
0.75
0.50
0.25
0.00


type
1.00
0.75
0.50
0.25
0.00
type
1.00
0.75
0.50
0.25
0.00


type
1.00
0.75
0.50
0.25
0.00
type
1.00
0.75
0.50
0.25
0.00


type
1.00
0.75
0.50
0.25
0.00
type
1.00
0.75
0.50
0.25
0.00


type
1.00
0.75
0.50
0.25
0.00
type
1.00
0.75
0.50
0.25
0.00

Clusters of addresses

These results come from analyzing and plotting just one single block. It is clear that while the majority of transactions follow a simple pattern of one address sending to two addresses (the receiver and possibly another address owned by the same sender), sometimes we are able to identify clusters of addresses being part of the same transaction. This is crucial information, especially in the input side of the transaction.

If a cluster of addresses all function as input to one transaction, it means that either these addresses belong to the same entity (remember that each wallet can generate a seemingly infinite amount of addresses), or the sender has access to the private keys of the addresses involved in the sending side of the transaction.

Dealing with bigger graphs

Upon realizing that the pipeline worked, i focused on trying to analyze a larger graph. The data i acquired consists of (almost) all the transactions in blocks between 500503 (07:36:28 AM 22-12-2017) and 500599 (11:10:12 PM 22-12-2017). This is almost 16 hours, in which around 100 blocks were mined. The network generated from these blocks contains over 1 Million edges.

Some calculations on this network are meaningless to try (such as diameter), but some of them are still doable (but still slow), such as computing the LCC for ever increasing subsets of this large graph.

Is the bitcoin transactions network a scale-free network?

The log-log degree distribution of samples of 100’000, 500’000, and the full network all suggest power-law properties for this network. A network whose degree distribution follows a power law, where most nodes have a low degree, while the number of nodes with high degree (hubs) decreases exponentially as the degree increases, is called a scale-free network.

Power laws and Scale Free Network

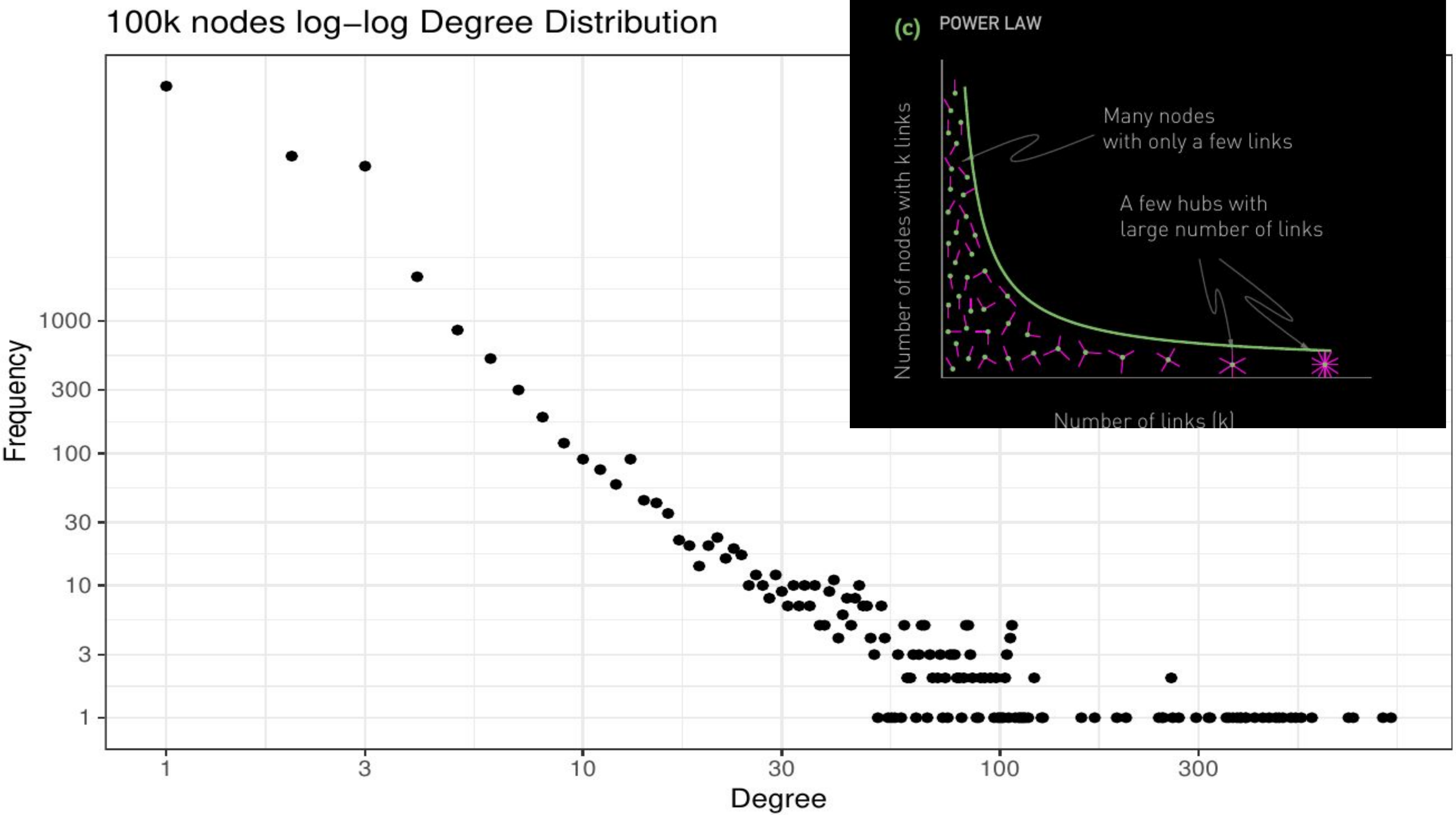
In a scale-free network, the log-log scale of the degree distribution of nodes is somewhat linear, suggesting that

$$p_k \sim k^{-\gamma}.$$

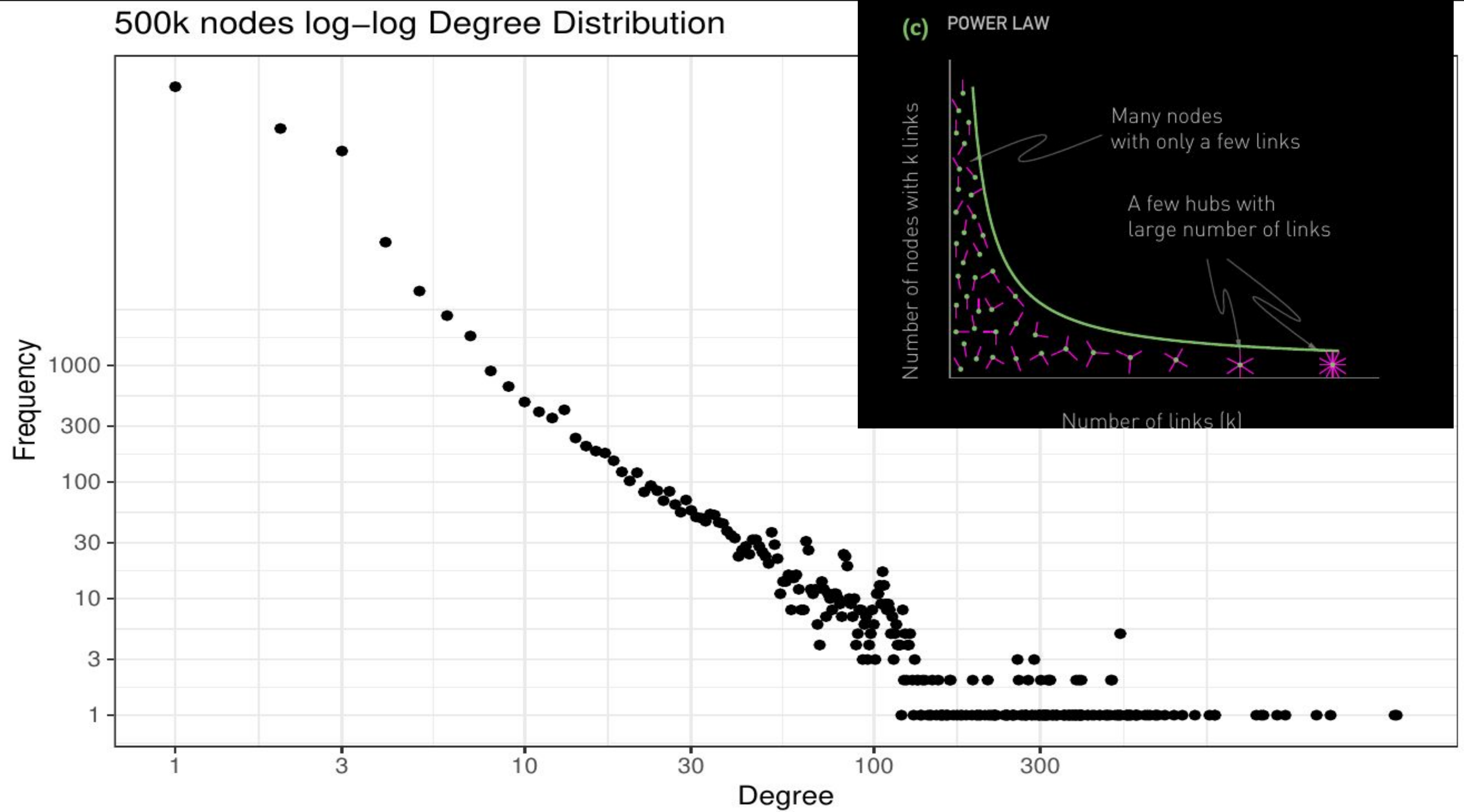
$$\log p_k \sim -\gamma \log k.$$

If this holds, log_pk is expected to depend linearly on log_k, the slope of this line being the factor gamma. The results on three sections of the graph suggest this somewhat linear relation on a log-log scale

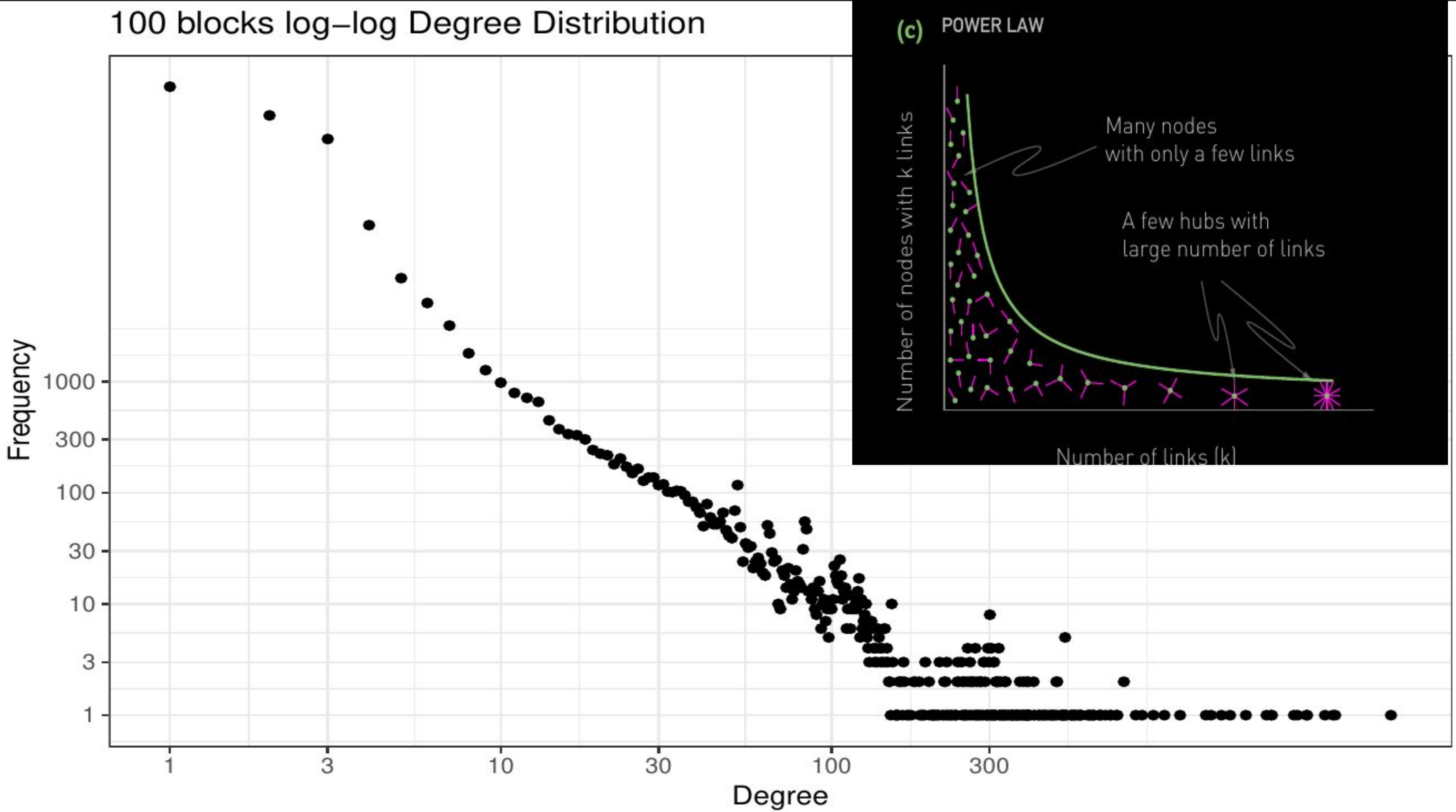
100k nodes log−log Degree Distribution



500k nodes log−log Degree Distribution



100 blocks log–log Degree Distribution



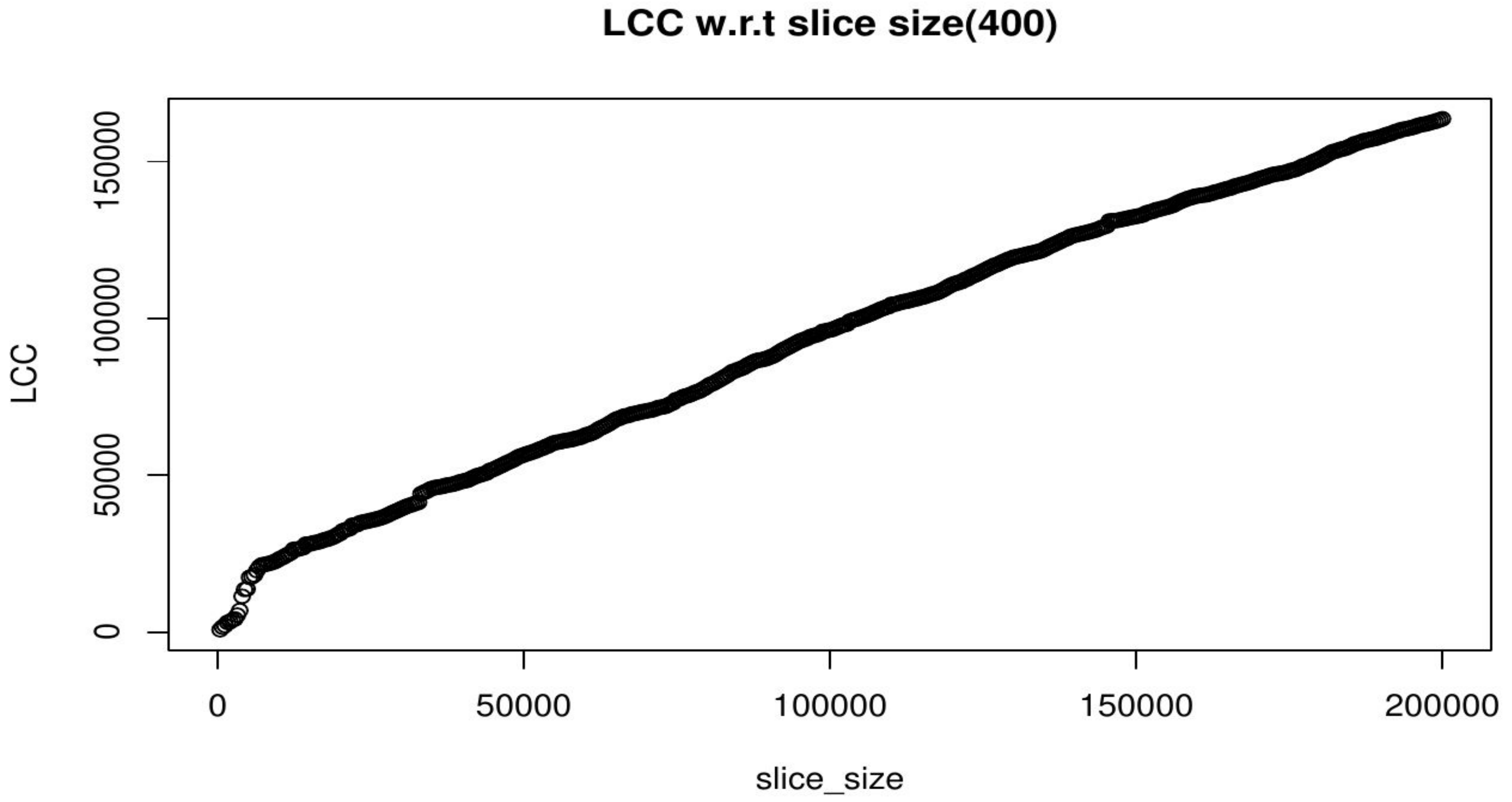
LCC on scale-free networks

In a scale-free network, the estimated degree of the largest node k_max is

$$k_{max} = k_{min} \; N^{\frac{1}{\gamma-1}}.$$

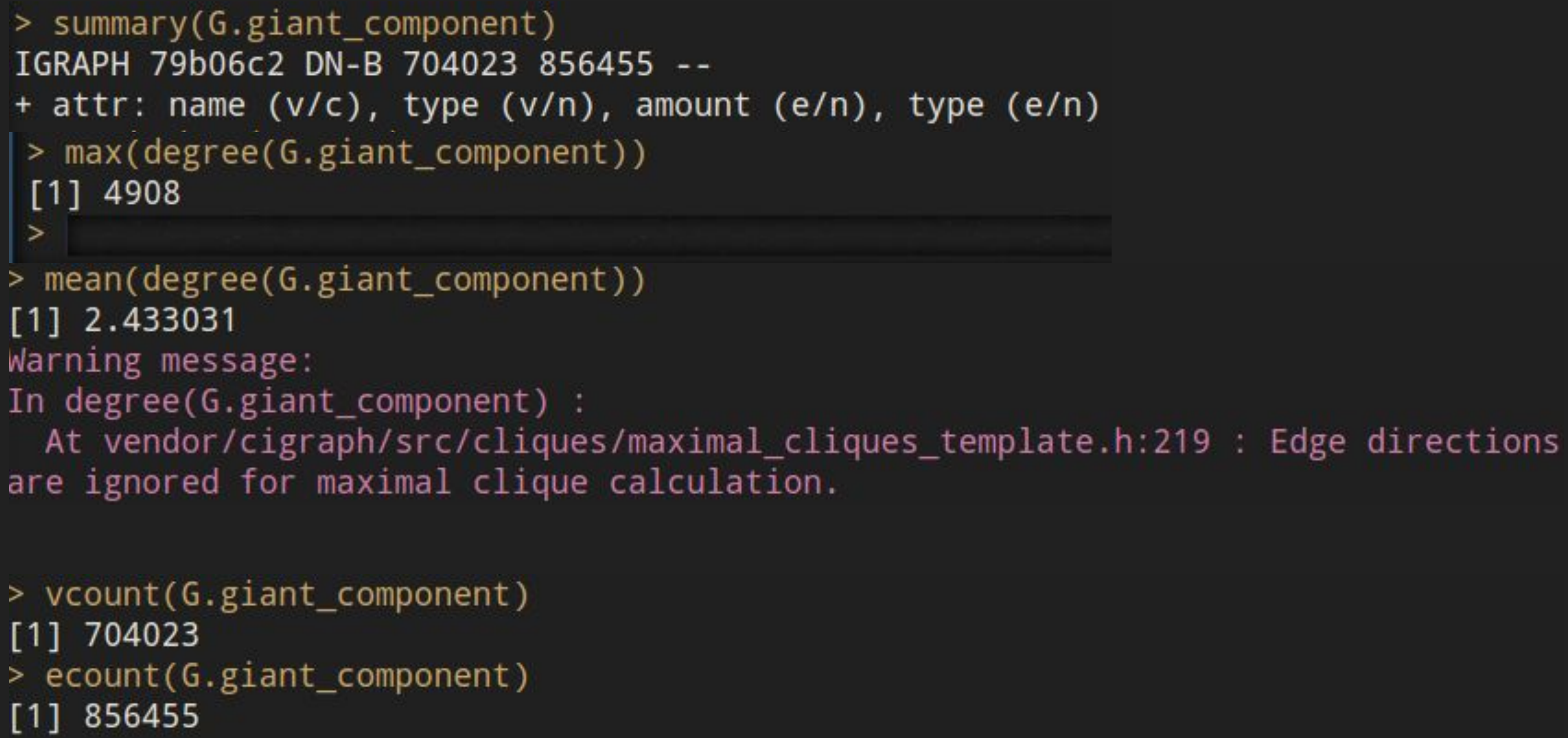
If this holds, the polynomial dependence of k_max on N would suggest that the larger the network, the larger the difference between the smallest node, k_min, and the biggest hub, k_max.

A linear behaviour of the size of the largest connected component in the network suggests that relation


LCC w.r.t slice size(400)
LCC
0
50000
100000
150000
slice_size
0
50000
100000
150000
200000

Giant Component

Giant component in this graph contains 704k nodes, 856k edges, with max degree equal to 4908.



```
> summary(G.giant_component)
IGRAPH 79b06c2 DN-B 704023 856455 --
+ attr: name (v/c), type (v/n), amount (e/n), type (e/n)
> max(degree(G.giant_component))
[1] 4908
> 
> mean(degree(G.giant_component))
[1] 2.433031
Warning message:
In degree(G.giant_component) :
  At vendor/cigraph/src/cliques/maximal_cliques_template.h:219 : Edge directions
are ignored for maximal clique calculation.


> vcount(G.giant_component)
[1] 704023
> ecount(G.giant_component)
[1] 856455
```

Conclusions

Is it possible to represent bitcoin transactions as a network? Yes

Can you infer ownership of bitcoin addresses by analyzing transactions? Possibly

Is the bitcoin transaction network a scale-free network? It’s likely

References

- Ron, D., & Shamir, A (2013). Quantitative Analysis of the Full Bitcoin Transaction Graph
- Network Science (Albert-László Barabási)
- Programming Bitcoin Learn How to Program Bitcoin from Scratch (Jimmy Song)
- Mastering Bitcoin Programming the Open Blockchain (Andreas M. Antonopoulos)
- https://www.lambertleong.com/projects/bitcoin_network_analysis

Code

Full project code at

https://github.com/ricvigi/DMAU2-public