



BitConduite: Exploratory Visual Analysis of Entity Activity on the Bitcoin Network

Christoph Kinkeldey, Jean-Daniel Fekete, Tanja Blascheck, Petra Isenberg

► To cite this version:

Christoph Kinkeldey, Jean-Daniel Fekete, Tanja Blascheck, Petra Isenberg. BitConduite: Exploratory Visual Analysis of Entity Activity on the Bitcoin Network. IEEE Computer Graphics and Applications, 2021, 10.1109/MCG.2021.3070303 . hal-03199547v1

HAL Id: hal-03199547

<https://inria.hal.science/hal-03199547v1>

Submitted on 15 Apr 2021 (v1), last revised 4 May 2021 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

BitConduite: Exploratory Visual Analysis of Entity Activity on the Bitcoin Network

Christoph Kinkeldey

Inria & Freie Universität Berlin

Jean-Daniel Fekete

Université Paris-Saclay, CNRS, Inria & LISN

Tanja Blascheck

Inria & University of Stuttgart

Petra Isenberg

Université Paris-Saclay, CNRS, Inria & LISN

Abstract—We present BitConduite, a visual analytics approach for explorative analysis of financial activity within the Bitcoin network, offering a view on transactions aggregated by entities, i.e. by individuals, companies or other groups actively using Bitcoin. BitConduite makes Bitcoin data accessible to non-technical experts through a guided workflow around entities analyzed according to several activity metrics. Analyses can be conducted at different scales, from large groups of entities down to single entities. BitConduite also enables analysts to cluster entities to identify groups of similar activities as well as to explore characteristics and temporal patterns of transactions. To assess the value of our approach, we collected feedback from domain experts.

■ **INTRODUCTION.** Bitcoin is a digital pseudo-currency and payment system based on strong public cryptography: a *cryptocurrency* [1], [2]. It challenges several notions of traditional banking as well as government-regulated currencies and transactions: using Bitcoin people can bypass traditional centrally governed payment systems. Bitcoin is legal to use virtually everywhere and a number of countries have officially accepted it as ‘private money’ [3]. Millions of people have directly transferred Bitcoin virtual money through its peer-to-peer network while building a large open data source called the Bitcoin *blockchain*: transactions bundled in blocks that form a chain.

Bitcoin, and in particular users’ transaction

activities, are an important data source to study because little is known about how Bitcoin compares to fiat currencies. Understanding behavior around the currency can help to explain certain Bitcoin phenomena such as its large volatility. In addition, a high level of technical expertise is required to extract, store and analyze Bitcoin transactions that domain experts who are interested in Bitcoin usually do not have. Only few approaches exist that lower the threshold of Bitcoin analysis and help with a deeper analysis.

We present BitConduite (Fig. 1), a visual analytics approach for the analysis of different types of activities and actor profiles in the Bitcoin network. It focuses on identifying and characterizing (but not de-anonymizing) *entities*: individuals,



Figure 1. BitConduite’s GUI contains five linked views: (A) filter view, (B) tree view, (C) cluster view, (D) entity browser and (E) transaction view. The teaser video shows the linking in action (<https://vimeo.com/317687395>).

commercial services or other groups using Bitcoin. BitConduite provides a workflow for systematic filtering and grouping of entities by their activity. Automatic clustering of entities helps analysts identify entity groups with similar activity. BitConduite’s purpose is to provide a first step in the analysis of Bitcoin: to derive overviews, questions and hypotheses about activities of entities. Our target group is analysts with a non-technical background, in particular researchers from the social sciences (e.g. economics) who may have technical expertise in statistical programming but usually not in preparing the raw Bitcoin data, performing address aggregation and conducting analyses with large amounts of data. Our approach is also in line with the notion of *data-first* design studies [4]. In addition, BitConduite is a means to collect questions and use cases from our analysts and help them formulate appropriate tasks.

1. Related Work

With the growing interest in Bitcoin as a financial and social phenomenon, methods and approaches to analyze Bitcoin data have emerged. In this section we present the most closely related past approaches for visual analysis of Bitcoin data.

Many websites offer simple visual analyses of Bitcoin blockchain data. For instance, *blockchain.info* [5] provides information such as the Bitcoin market value or the number of

transactions per block. Most of these websites provide information in the form of simple charts that resemble stock charts and presumably provide information for investors as target users. Only a small number of systems support more complex visual analyses of different Bitcoin characteristics. One example is *SuPoolVisor* [6] that supports surveillance of mining pools and de-anonymization of pool members. It visualizes information about mining pools (e.g. computing power) and their transactions. Similarly, *BitEx-Tract* [7] supports the analysis of Bitcoin exchanges, i.e. platforms to buy and sell Bitcoin. Transactions between exchanges can be analyzed over time as well as between exchanges and their clients. Both approaches focus on specific types of entities in Bitcoin (mining pools and exchange platforms) and are restricted to the respective subsets of transactions whereas in BitConduite we allow an exploratory analysis of transactions of all types of actors. On a more detailed level, *BitConeView* [8], displays the traces of specific transactions in a Gantt chart to support an analyst in detecting suspicious mixing of Bitcoins through the blocks (*taint analysis*). Other than BitConduite, it is tailored to one special task and provides insights on the transaction level only. Another visual approach of this kind is *BlockChainVis* [9] that shows node-link diagrams of transactions and enables analysts to filter by block, number of

transactions or the amount of the transaction. The basic approach is similar to BitConduite's but it is transaction-centered (not entity-centered), the filtering part is limited in comparison and advanced processing like clustering is not possible. McGinn et al. [10] present a dynamic node-link diagram of transactions between addresses in a visually appealing display. The authors identify structures in the graph that may indicate certain types of actors (e.g. commercial platforms). However, this approach only displays a short snapshot so no long-term insights are possible. In addition, it shows the raw address-based transaction data and no data processing is possible.

In summary, approaches for visually supported analysis of user activity on the Bitcoin blockchain are rare and only cover functionality for answering fixed questions [11]. The goal of BitConduite is to offer a generalized, long-term, entity-centered perspective for exploratory analysis of activities that no other approach provides yet.

2. BitConduite System

BitConduite consists of a back end for data preparation and management as well as for high performance data access and a front end with a graphical user interface (GUI) that comprises five linked views (Fig. 1). We collaborated with three economist researchers from a university and one cryptography researcher from our institution who regularly provided feedback to inform the system's development. BitConduite's design is based on a data-first methodology [12] triggered by real-world data and high-level exploratory analysis tasks supported by the data source. Next, we describe the components of the system in detail.

2.1. Activity Measures

To systematically describe an entity activity we defined eight simple statistical measures together with the experts who emphasized the need for measures that are simple and easy to understand. To ensure that the measures are sufficiently expressive we designed them to facilitate explanation of entity groups from the literature (such as Athey et al.'s [13] user model). The eight measures are listed in Table 1. With this set of measures we are able to describe an entity's activity related to number, time, amount and type of transactions. In all views of the GUI, the colors shown in Table 1

consistently represent the activity measures. In the future, BitConduite will also be extended to include additional activity measures when other types of activities are analyzed.

2.2. Data Acquisition and Preparation

We downloaded the raw data of blocks and transactions, imported them into a *MongoDB* database and then extracted the transaction data into a column-oriented *MonetDB* database. The latter enables fast aggregation of the data needed for computing the activity measures. Due to BitConduite's exploratory nature it requires computationally expensive re-computation of the measures for any time range on the fly. To accelerate this process, we further wrote the entity-related data into *HDF5* files that are loaded into memory where the server software can access them quickly. We opted for an in-memory solution that uses the *pandas* (Python data analysis) library for fast data processing.

We base all our analyses on high-level entities rather than on Bitcoin addresses. To do so, we aggregated all transaction addresses using the Reid and Harrigan [14] heuristic that has been shown to be effective [15]. First, we exported address pairs that appear together as inputs of a transaction. From them we constructed a graph with the addresses as nodes and their co-occurrence (being input address of the same transaction) as links using the *NetworkX* library in Python. A UnionFind algorithm yielded all the addresses that are linked to the entities, following the heuristic. The result is a list of addresses for each entity.

We downloaded and scraped lists of known addresses from public sources such as WalletExplorer [16]. With this information we were able to tag over 70,000 addresses that added context to the analysis with BitConduite.

2.3. Workflow

The workflow for exploratory analyses with BitConduite (Fig. 2) includes the following high level tasks:

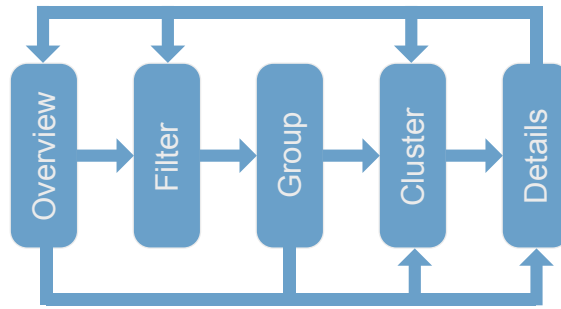
Overview. Inspecting an overview of all activity measures related to the whole dataset or to subsets.

Filter. Specifying dynamic queries to filter data and focus on regions of interest over time and activity measures.

Group. Defining and organizing groups of entities

Table 1. Measures describing entity activity. Abbreviation *s/a/l* stands for “smallest / average / largest”.

Measure	Color	Description	Definition
<i>num_txs</i>	■	Number of transactions (as sender / as receiver)	Overall number of transactions per entity.
<i>time_first</i>	■	Time of first transaction	Point in time, from which an entity was active.
<i>time_last</i>	■	Time of last transaction	Point in time, until which an entity was active.
<i>time_active</i>	■	Time active in days	Duration: last minus first transaction of an entity.
<i>amount_rec</i>	■	Amount received in BTC: <i>s/a/l</i>	Amount of Bitcoin an entity received.
<i>amount_sent</i>	■	Amount sent in BTC: <i>s/a/l</i>	Amount of Bitcoin an entity sent.
<i>num_inputs</i>	■	Number of inputs: <i>s/a/l</i>	Number of input addresses per transaction.
<i>num_outputs</i>	■	Number of outputs: <i>s/a/l</i>	Number of output addresses per transaction.

**Figure 2.** Workflow for exploratory Bitcoin activity analysis in BitConduite.

with similar activity.

Cluster. Automatic grouping of entities across activity measures to determine suitable value ranges for creating meaningful groups.

Details. Exploration of entities’ characteristics in detail.

2.4. Graphical User Interface

BitConduite’s GUI (Fig. 1) consists of five linked views: filter view, tree view, cluster view, entity browser and transaction view. They are integrated into a single page web application. All five views are dynamically updated with every change and can be manipulated independently for iterative exploration of the data. In the following, we describe the five views and provide more details in the use cases section.

2.4.1. Filter View. The filter view (Fig. 1-A) is a dashboard that provides an overview on the temporal distribution of transactions as well as histograms for the activity measures listed in Table 1. Initially, the time and value distributions

are displayed for all entities in the current data set. The analyst can filter entities on any of the histograms using brushing or a text input field. Pressing the filter button confirms the selection and filters the current set of entities. For some activity measures (those related to all transactions of an entity) the analyst can switch between smallest, average and largest value per entity.

2.4.2. Tree View. The tree view (Fig. 1-B) shows a hierarchy of partitions representing groups of entities, visualized as an icicle tree for its compactness. Initially, only the root node, representing the group of all entities, is visible and automatically marked as active. Every time a filter is executed in the filter view, it is applied to the currently selected node. A new row is added in the tree below its parent with two new sets of entities: those that fulfill the filter condition and the remainder set. In Fig. 3 we see the whole set of entities in the tree view (Fig. 1-B). The filter view histogram for “largest amount received” (Fig. 1-A) shows a range from 0–90,000 BTC. We apply a range filter of 0–10 BTC. A new row with two new groups of entities appears in the tree: those that fulfill the filter criterion (left) and the remaining ones (right). The small bars next to the labels signify the relative number of entities per class and tooltips provide detail-in-context. We show the nodes with equal widths and add a glyph to represent the number of entities inside each node. In an initial design, we scaled the size of each node by the number of entities but often groups were small and their nodes became hardly visible. Clicking on a node selects it as the current context and updates the visualizations. Labeling

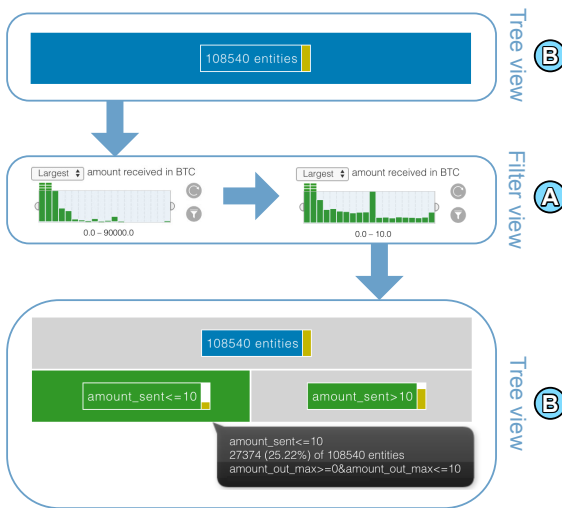


Figure 3. Principle of the entity group tree. Filtering (middle) the set of 108,540 entities (top) yields two subsets in the entity group tree (bottom): entities meeting the condition (left) and the remainder (right)

the nodes and deleting rows is possible as well. Using the tree view, the analyst can iteratively build up entity group trees and switch between the groups to compare their characteristics. An export function saves the entity group tree to a file for archiving and sharing.

2.4.3. Cluster View. This view (Fig. 1-C) provides entity clustering functionality to analyze groups of entities sharing similar characteristics with respect to activity measures of interest. This can help the analyst to discover groups of entities with similar activity without explicitly defining value ranges (as with filtering). Before starting the clustering it is necessary to select the desired number of clusters and one or more activity measures to be considered in the clustering. We use the *k-means* clustering algorithm from the *scikit-learn* machine learning library [17], a simple yet fast, flexible and scalable clustering algorithm. For example an analyst might be interested in entities that sent similar amounts of Bitcoin and were active in a similar time period. After the clustering is completed, 8-axes star glyphs represent the characteristics of each cluster across all activity measures. We chose the star glyph over other techniques such as parallel coordinates because star glyphs allow a representation using small multiples, make the cluster representation

relatively scalable without clutter and allow us to re-use the representation for the display of individual entities. The glyphs also serve as orientation for the choice of an appropriate number of clusters: if two or more clusters are visually similar and only differ in a measure that is not of interest, the analyst can decide to reduce the number of clusters and restart the clustering.

2.4.4. Entity Browser. The results of the filtering and clustering steps in BitConduite are groups of entities. For the members of a group, the entity browser (Fig. 1-D) shows detailed individual characteristics. It is not designed to show as many entities as possible but a sample of entities of interest. We chose a 20×20 grid of star glyphs showing up to 400 entities at the same time (with paging to see additional glyphs). A star glyph analogous to the one in the cluster view (Sect. 2.4.3) represents every entity and shows at a glance similarities and differences between them. We chose star glyphs over other glyph designs as they allow for discriminability of 10 or more dimensions due to their radial layout and because they use an effective position encoding for quantitative data [18]. We used a slight variation of the standard star glyph where we dual-encoded the axis category by color, always drew the 100% reference line and used a contour to indicate a shape for each data point that could be used for judging data point similarity. For each entity glyph, tooltips also show the exact name and values of each star glyph ray. The entity glyphs can be sorted by different attributes in ascending or descending order, for example to show the entities with the maximum number of transactions first. Or the analyst can determine and compare, e.g. the top ten entities with the maximum amounts received in a transaction.

2.4.5. Transaction View. When the analyst clicks on a glyph in the Entity Browser, it is shown magnified and its transactions are displayed in the transaction view (Fig. 1-E) on a timeline. Circles show transactions over time with their size representing the respective amount. The analyst can switch between the transactions an entity sent or received. The timeline reveals the temporal distribution of transactions and their amounts, answering questions related to an entity's activity

pattern over time as well as to single transactions.

2.5. Tasks Supported by Views

The views described above support the tasks of the workflow (Sect. 2.3) as follows:

Overview. The *filter view* visualizes histograms for all activity measures related to the whole dataset or to filtered subsets.

Filter. Interaction with the *filter view* allow the creation of dynamic queries to filter data and focus on regions of interest over time and activity measures.

Group. The *tree view* tracks filtering steps in a hierarchical way. The resulting groups of entities can be labeled, for example, as “short-term investors” vs. “long-term investors.”

Cluster. The *cluster view* supports interactive explorative grouping of entities across activity measures.

Details. The *entity browser* enables analysts to explore the attributes of entities in detail. The *transaction view* shows transactions of a single entity over time and the amounts of BTC transferred.

2.6. Temporal Analyses

BitConduite provides several ways to analyze temporal attributes of entities: in the filter view, the cluster view and the transaction view. In the filter view, it is possible to limit the group of entities under analysis to a specific time period (e. g. all entities active in 2009). In addition, three activity measures can be used to apply different types of temporal filtering: time of first and last transaction and the number of days an entity was active (e. g. to get the group of entities that have been active for at least a year). In the cluster view, entities can be clustered by any of those three time related measures to identify groups of entities with similar temporal behavior (e. g. separating entities into groups of short-term and long-term activity). The entity view allows to sort entities by time to compare when they were active (e. g. to identify the ten entities with the longest time active). When selecting a single entity, the transaction view gives an overview of temporal patterns in the transactions of this entity (e. g. to find out if the entity’s activity was temporally regular or not). Combining this functionality, complex analyses of temporal aspects of activity are possible.

3. Use Cases

The main goal of BitConduite is an easy-to-use and flexible, visually supported grouping of activities on the Bitcoin blockchain. Next, we present two use cases that demonstrate how an analyst can use the approach to conduct complex analyses in a simple way. The first use case is a comparison of activities in the first years of Bitcoin, with a closer look on how activity patterns changed over time. The second use case is about the effect of a Bitcoin-specific event, the second “halving day” in 2016, on the activities of miners.

3.1. Classifying Entities: The Early Years

Bitcoin evolved from a cryptocurrency without any real value to an asset for investment of huge amounts of money that has been running for over a decade now. Analyzing how it developed in the first years may help to predict if or which other cryptocurrencies might be successful.

For the years from 2009–2011, BitConduite shows 1.9 million entities that used 2.8 million addresses. The transaction view shows that the number of transactions has increased in 2011 with a peak in June representing 328,000 transactions in this month (Fig. 4). Looking at the first years, we would like to address two groups of entities in detail: “one-timers” and most active entities.

3.1.1. One-timers. One research question our domain experts raised was whether Bitcoin is generally used as an investment or is actively transferred, for example to make purchases. From the filter view we learn that the values for activity measures are highly skewed, especially *amount sent* and *amount received*. This is also true for the *number of transactions*: the majority of entities is involved in a low number of transactions. To identify all entities with just one transaction (*one-timers*) we define a filter, for which *number of transactions* (as receiver) equals 1. The result is a group of entities that only received Bitcoin value once (e. g. by mining or purchasing) and have remained inactive since. In the tree view we label this group as “one-timers” and the complementary group as “multi-timers”. Until the end of 2011, the majority of all entities (about 85%) were one-timers. There are several possible explanations for this behavior. Either people bought Bitcoins and forgot about them, they lost their private key

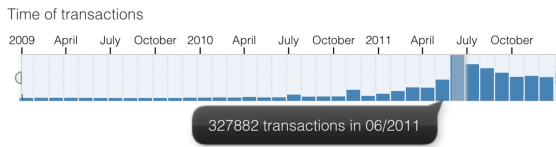


Figure 4. Monthly transaction count from 2009–2011.

or they invested and waited to sell at a better price. Further filter settings could help with these possible explanations by comparing whether early one-timers transferred their Bitcoins in later years when, for example the price increased.

Using the time filter we, next, determine that the fraction of one-timers dropped over the years from 95% (2009) to 87% (2010) to 85% (2011). In the first years of existence, most people did not spend their Bitcoins at all. The reason could be that no exchanges existed before 2010 and although New Liberty Standard defined a first exchange rate in October 2009, Bitcoin’s market value was still \$0 in practice and there was no real commercial opportunity to spend Bitcoins.

3.1.2. Most Active Entities. To identify entities with the highest activity in the early years, we switch to the “multi-timers” class by clicking on the associated node in the tree view. Looking at the filter view we notice that the *number of transactions* per entity ranges from 0 to 57,384 (as sender) and 1 to 189,951 (as recipient). We use the cluster view to cluster the multi-timers by four activity measures: *number of transactions*, *time active*, as well as *largest amount received* and *amount sent*. We obtain three clusters: one cluster with over 35,000 highly active entities and two clusters of entities with less activity differing in *number of transactions* and average *number of inputs*. We can tell from the glyphs that the two first clusters are similar so we reduce the number of clusters to two and restart clustering. The result is now a large cluster of 244,532 active entities and a small one with 39,043 entities of high activity (Fig. 5). By clicking on the glyph representing the first cluster we load the 39,043 active entities into the entity browser. Sorting by *number of transactions* yields four entities that are more active in relation to the others. The most active entity is tagged as “Mt. Gox”, the Bitcoin exchange platform that started in June

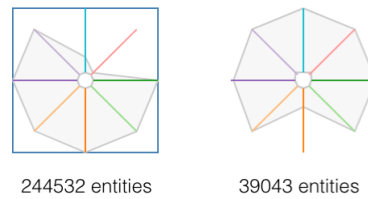


Figure 5. Glyphs representing clusters of multi-timers in search for the most active entities (left: large cluster of low activity, right: small cluster of high activity).

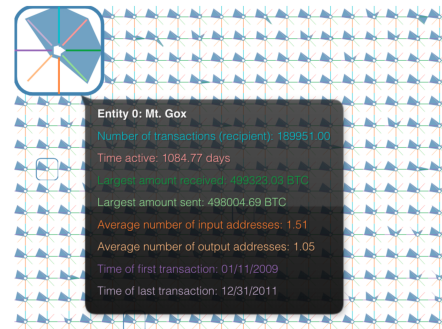


Figure 6. The most active entity is Mt. Gox, an early exchange platform.

2010 (Fig. 6). This simple analysis shows that from their first appearance, big platforms have dominated the activity on the Bitcoin blockchain. This is a phenomenon that is still valid nowadays.

3.2. Analyzing the Halving Day 2016

This second use case concerns Bitcoin mining. The mining market has changed over the years and has become more centralized with large mining pools dominating the market. This is of interest for researchers in economics who examine the mining market and its impacts on the Bitcoin system. *Halving days* (when mining rewards are cut by 50%) are deemed important because of their impact on the Bitcoin price and mining activities.

To see effects of the halving day on July 9, 2016, we take a closer look at mining activities during the months before and after this day. It is known that the Bitcoin price remained relatively stable after but not much is known about changes in mining activity. The economics experts we collaborate with were interested in whether the same entities successfully mined Bitcoin after the halving day or if some potentially gave up due to decreasing profit.

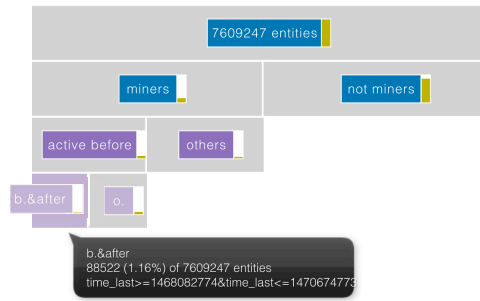


Figure 7. The group of entities (“b.&after”) that were active as miners before and after the halving day.

To answer this question we focus on transactions from 30 days before and 30 days after the halving day. Looking at the time chooser and the tree view, we learn that during this time period, 7.6 million entities took part in about 13.8 million transactions (as sender, recipient or both). The histograms in the filter view reveal that although the *number of transactions* ranges from 0 to 1,075,326 (as sender) and 1 to 3,074,401 (as receiver) the vast majority of entities took part in a small number of transactions.

3.2.1. Identify Miners. Miners can be identified as receivers of the “coinbase” transaction, which is the first transaction in each block that rewards them for mining the block [2]. It does not have a regular input address like other transactions. That is why we can identify entities that were involved in mining activities by applying a “zero input” filter (smallest recorded *number of inputs* = 0). In the entity group tree we label them as “miners” and the complementary group as “not miners” (Fig. 7 second row). We learn that 1,050,441 (roughly 13.8% of all) entities in this time period belong to the “miners” group. Using the time chooser we find out that the *number of transactions* after the halving day is about 16% lower than in the month before. Does this indicate less mining activity after the halving day?

3.2.2. Activity of Miners around the Halving Day. To explore mining activities we cluster miners by number of transactions as recipient and the time they were active in the selected time period. Starting with three clusters, the cluster view shows a large group of entities with low activity (905,732 / 86.2%), a smaller group of

high activity (115,373 / 11.0%) and a tiny one in between (29,336 / 0.03%). We again decide to cluster with only two groups to merge the two similar clusters. The result is a large cluster (922,245 / 87.8%) with low activity and a much smaller one with high activity (128,196 / 12.2%). Looking at the glyphs and their details in the cluster view reveals that the two groups also differ in terms of the range of amounts received and sent and the number of input addresses per transaction.

To examine whether miners continued or stopped their activity after the halving day, we first find miners that were active in the weeks before the halving day. We apply a filter by *time of first transaction* before the halving day yielding a group of 7.9% of all entities we name “active before.” We filter this subgroup setting the *time of last transaction* filter to the time period after the halving day. This yields the group of miners that were active before the halving day and remained active afterwards (1.2% of all entities) (Fig. 7). As a complementary group we get those that “gave up”, i. e. they were active before but not in the 30 days afterwards (6.7% of all entities).

This analysis shows that a large fraction (roughly 85%) of the miners that were active before the halving day did not receive mining rewards in the following 30 days. The reason could either be that they stopped mining after the halving day because the lower reward made their work no longer profitable or that they still took part in the mining competition but were simply not successful. The blockchain does not provide information about this and without further information we can only speculate about the reasons. However, indeed, the number of miners seems to have decreased significantly after the halving day. The transaction view shows that one of the most active miners from before the mining day was inactive during the 30 days after the halving day. From the observation above we can conclude it is likely that the halving day had a relevant impact on mining activities.

All in all, the use cases shows that—while keeping in mind the uncertainty inherent to our method due to the entity clustering heuristics—BitConduite can support a range of complex explorations and insights.

4. Expert Workshop

We conducted a half-day workshop with Bitcoin experts to receive feedback on BitConduite’s support for exploratory analyses based on real Bitcoin-related analysis questions.

4.1. Setup, Procedure and Data Collection

We set up six workstations in a large shared office in our lab. Participants interacted with BitConduite running in a Chrome/Chromium web browser showing data about the early years of Bitcoin from 2009–2011.

The experiment included a 30 minute training phase, during which we presented the system using a slideshow and gave participants three hands-on exercises to complete. We answered participants’ questions throughout the training. We stopped the training after participants confirmed that they understood the BitConduite workflow. Next, participants began a 60 minute free exploration phase, during which they used BitConduite to answer their own questions about the Bitcoin blockchain. During the free exploration phase the three experimenters present answered questions about BitConduite if help was needed.

Throughout the workshop the participants filled out three questionnaires, one about demographics and background, one to record their questions regarding Bitcoin (see Table 2) and a final one to collect structured feedback on BitConduite’s usability. During the free exploration, we recorded participants’ actions using the tool.

4.2. Background Information

We invited 6 participants (5 male, 1 female). Their age ranged from 25 to 51 years (average: 34.7 years). Two participants were students and the other participants were professional researchers: a senior researcher, research engineer, assistant professor and associate lecturer. The professional researchers had been involved in the development process of BitConduite as external voluntary Bitcoin experts. Thus, they were familiar with the approach but had not seen the complete version of the system before. All participants confirmed to have experience with Bitcoin, ranging between one month and five years (average: 1.9 years, one participant did not provide this information).

Table 2. Example questions participants wanted to explore with BitConduite during the free exploration phase. Questions that can be answered with BitConduite are marked with an asterisk (*).

Category	Questions
Exploring specific entities (known entities)	Does Kraken have liquidity issues? (P2) How many [B]itcoins does Satoshi own? (P2)
Exploring specific entities (other entities)	*Do [entities] [exchange] money with the same people? (P1) Who are the 10 main owners of Bitcoin and how much [do] they own? (P3)
Linking and relationships between entities	*Do [entities] send small amounts to 1 person (or the opposite = large amount[s] to multiple)? (P1) Do mining pools interact with each other directly? (P1) *Explore and link multiple entities to a single one based on its behavior (P1)
Exploring trends (behavior)	*Which factors affect pools of miners dynamics? (P1) *What happened to BTC exchange platforms during trouble periods? (P6)
Exploring trends (temporal)	*Are there daily users that are non-miners / non-professionals? (P1) Is there any seasonality in the use of BTC? (P5)

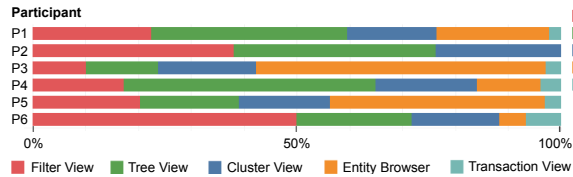


Figure 8. Relative proportion of mouse-click interactions of each participant on the main views.

4.3. Results

In this section we report on the results from the interaction logs, user experience questionnaires and participants’ research questions.

Interaction Logs. Interaction logs for each participant provide a first approximation of which system features they did or did not use. We extracted the number of times participants executed a logged event (through a mouse click), compared to the overall number of clicks and compiled these proportions in an overview (Fig. 8).

Given that participants were interested in a large variety of different research questions it is not surprising that Fig. 8 shows a large variety in the proportions of interactions for each participant. Use of the filter view was extremely varied (10%–50%); similarly for the entity browser (0%–55%).

Overall, the transaction view was interacted with the least. There are several possible explanations for its diminished use: research questions may not require studying individual entities and their transactions; the view was at the bottom of the page; and we did not log all interactions with the timeline (e. g. changing the time range).

The interaction patterns show that all participants executed repeated sequences of switching between the filter and tree view (*filter* → *tree* → *filter* → *tree* → *filter*), which shows that our choice to place them visually next to each other worked well to reinforce their connection. Looking at our intended workflow for BitConduite (*filter* → *tree* → *cluster*), we find that all participants used this sequence except for P4. This participant faced technical difficulties and, therefore, his/her workflow was interrupted. Another typical sequence participants P3, P4 and P6 applied was *filter* → *cluster* → *entity browser*, meaning that they skipped modifying the tree view and just used the automatically selected group. From this data, we conclude that participants used BitConduite the way we intended and participants used similar interaction strategies to analyze the data.

User Experience. We asked participants to rate the usability of BitConduite. No participant found the system unnecessarily complex. One participant agreed and one was neutral that they would need further support to use the approach. Similarly, one participant agreed that they needed to learn many things before its use. All but one participant were confident with using the system, found it easy to use, its functionality well integrated and would use BitConduite more frequently.

Answering Research Questions. We received eleven answers in the exit questionnaire on whether BitConduite helped participants to answer their questions; five positive and six negative. The most common problem was that additional data was needed that was not included in the data we had loaded for the study. For example several participants wanted to see newer data than we provided or additional data we had not extracted (yet) from the blockchain.

We asked the participants for new discoveries using the approach. P1 had two insights related to the amount of BTC transferred and generated new questions regarding one time users. P6 made discoveries about payouts from mining pools. P3

found that the clustering offered new and surprising results. Despite the inability of BitConduite to help with several questions, we collected many positive responses to the user experience-related questions reported above. It is likely the ability to see data in new ways and discover new aspects of the Bitcoin blockchain that led to positive responses about the system.

5. Discussion

BitConduite’s preliminary evaluation revealed the usefulness of its entity-based analysis and the importance of approaches for exploratory analyses. Still, we identified two major limitations related to our approach and the entity clustering we use. **Approach.** Our exploratory approach requires flexible computation of activity measures, which causes limitations with respect to scalability. While we tried to keep data processing times in the range of a few seconds, the main bottleneck is the on-the-fly clustering that can take minutes in the worst case (if many entities are clustered at once). Here, a progressive clustering strategy would help [19]. **Entity Aggregation.** Entity based analysis is insightful, however, uncertainty exists regarding entities as there are no methods to measure entity aggregation quality. *Mixing services* (also called *tumblers*) obfuscate the links between addresses for privacy reasons and make address aggregation more difficult. In addition, entity aggregation is computationally expensive and has a large memory footprint in our implementation.

6. Conclusion and Future Work

The main contribution of BitConduite is to make in-depth exploratory analysis of Bitcoin entity activity possible, lowering the threshold for analysts without the technical background to prepare and handle the data. An important part of its workflow involves systematic and reproducible grouping of entity activities using filtering coupled with a tree representation. Clustering can be used to reveal new groups of entities with similar activity. Starting with large scale analyses it is possible to drill down and retrieve detailed information on single entities and display their transactions on a timeline. In two use cases we demonstrated how BitConduite can help characterize entity activity to answer questions relevant to Bitcoin experts.

During a workshop with Bitcoin experts we

learned that several research questions could easily be answered using BitConduite (e. g. about trends and outliers in activities or mining behavior). Questions regarding temporal trends could not be answered (e. g. seasonality in the use) and pointed out a limitation of the approach. Ratings concerning BitConduite’s usability (confidence, ease-of-use, learnability) were predominantly positive. Overall, five out of six experts said they would like to use BitConduite more frequently.

Limitations of our approach stem from the fact that aggregation of addresses to entities provides a new perspective but adds uncertainty that cannot be reliably quantified. To decrease uncertainty one could include more external information such as tagging of entities, requiring de-anonymization, which was not our goal in this project.

The workshop showed that the most important extension of our work would be a more convenient comparison of temporal patterns. An additional view could be integrated, e. g. a radial chart or similar to facilitate comparison of activity patterns over time. Another useful extension would be similarity search, i. e. suggestion of entities similar to a specific entity of interest or search for anomalies, i. e. entities with abnormal activity. Lastly, future work will be to add the capability to track addresses and individual entities by integrating the functionality we demonstrated in a separate approach called the *Blockchain Entity Explorer* [20].

7. Acknowledgements

We thank the anonymous reviewers for their feedback. This research was partially supported by Labex DigiCosme (project ANR-11- LABEX-0045-DIGICOSME) operated by ANR as part of the program “Investissement d’Avenir” Idex Paris-Saclay (ANR-11-IDEX-0003-02). Special thanks are due to Daniel Augot from Inria Saclay / LIX for constantly supporting our work this project with expertise and ideas.

■ REFERENCES

1. S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008. [Online]. Available: <http://bitcoin.org/bitcoin.pdf>
2. A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, 2016.
3. “Wikipedia: Legality of bitcoin by country or territory.” [Online]. Available: https://en.wikipedia.org/wiki/Legality_of_bitcoin_by_country_or_territory
4. M. Oppermann and T. Munzner, “Data-first visualization design studies,” in *Workshop on Evaluation and Beyond - Methodological Approaches to Visualization (BELIV)*, 2020, pp. 74–80.
5. “Blockchain.info.” [Online]. Available: <https://blockchain.info>
6. J.-z. Xia, Y.-h. Zhang, H. Ye, Y. Wang, G. Jiang, Y. Zhao, C. Xie, X.-y. Kui, S.-h. Liao, and W.-p. Wang, “SuPoolVisor: a visual analytics system for mining pool surveillance,” *Frontiers of Information Technology & Electronic Engineering*, vol. 21, no. 4, pp. 507–523, Apr. 2020. [Online]. Available: <http://link.springer.com/10.1631/FITEE.1900532>
7. X. Yue, X. Shu, X. Zhu, X. Du, Z. Yu, D. Papadopoulos, and S. Liu, “Bitextract: Interactive visualization for extracting bitcoin exchange intelligence,” *IEEE TVCG*, vol. 25, no. 1, pp. 162–171, 2018.
8. G. Di Battista, V. Di Donato, M. Patrignani, M. Pizzonia, V. Roselli, and R. Tamassia, “Bitconeview: Visualization of flows in the bitcoin transaction graph,” in *Symposium on Visualization for Cyber Security*. IEEE, 2015.
9. S. Bistarelli and F. Santini, “Go with the—bitcoin—flow, with visual analytics,” in *Conference on Availability, Reliability and Security*. ACM, 2017, pp. 38:1–38:6.
10. D. McGinn, D. Birch, D. Akroyd, M. Molina-Solana, Y. Guo, and W. Knottenbelt, “Visualizing dynamic bitcoin transaction patterns,” *Big Data*, vol. 4, no. 2, pp. 109–119, 2016.
11. N. Tovanich, N. Heulot, J.-D. Fekete, and P. Isenberg, “A Systematic Review of Online Bitcoin Visualizations,” *EuroVis 2019 - Posters*, 2019, poster.
12. M. Oppermann and T. Munzner, “Data-first visualization design studies,” *CoRR*, vol. abs/2009.01785, 2020. [Online]. Available: <https://arxiv.org/abs/2009.01785>
13. S. Athey, I. Parashkevov, V. Sarukkai, and J. Xia, “Bitcoin pricing, adoption, and usage: Theory and evidence,” Stanford University Graduate School of Business, Tech. Rep. Research Paper No. 16-42, 2016. [Online]. Available: <https://ssrn.com/abstract=2826674>
14. F. Reid and M. Harrigan, “An analysis of anonymity in the bitcoin system,” in *Security and Privacy in Social Networks*. Springer, 2013.
15. M. Harrigan and C. Fretter, “The unreasonable effectiveness of address clustering,” in *Conference on Advanced and Trusted Computing*. IEEE, 2016, pp. 368–373.

16. “WalletExplorer: Bitcoin block explorer.” [Online]. Available: <https://www.walletexplorer.com/>
17. “Scikit-learn machine learning in python.” [Online]. Available: <http://scikit-learn.org/stable/>
18. J. Fuchs, P. Isenberg, A. Bezerianos, F. Fischer, and E. Bertini, “The influence of contour on similarity perception of star glyphs,” *IEEE TVCG*, vol. 20, no. 12, pp. 2251–2260, Dec. 2014.
19. J.-D. Fekete and R. Primet, “Progressive analytics: A computation paradigm for exploratory data analysis,” *CoRR*, vol. abs/1607.05162, 2016.
20. P. Isenberg, C. Kinkeldey, and J.-D. Fekete, “Exploring entity behavior on the bitcoin blockchain,” *IEEE Conference on Visualization - Posters*, 2017.

Christoph Kinkeldey is a Post Doctoral Researcher at Freie Universität Berlin. He specializes in visual analytics and geovisualization and is particularly interested in how visualization of uncertainty can support people to gain a better understanding of complex information. Contact him at christoph@kinkeldey.de.

Jean-Daniel Fekete is the Scientific Leader of the Inria team Aviz that he founded in 2007. He received his

PhD in Computer Science in 1996 from University of Paris-Sud, France, joined INRIA in 2002 and became Senior Research Scientist in 2006. His main research areas are Visual Analytics, Information Visualization and Human Computer Interaction. Contact him at jean-daniel.fekete@inria.fr.

Tanja Blascheck is a Post Doctoral Researcher at the University of Stuttgart. Her main research areas are information visualization and visual analytics with a focus on evaluation, eye tracking and interaction. She is interested in exploring how to effectively analyze eye tracking data with visualizations and the pervasive use of visualization on novel display technology like smartwatches. Contact her at research@blascheck.eu.

Petra Isenberg is a research scientist at Inria, France in the Aviz team. Her main research areas are visualization and visual analytics. She is interested in exploring how people can most effectively work when analyzing large and complex data sets—often on novel display technology such as small touchscreens, wall displays or tabletops. Contact her at petra.isenberg@inria.fr.