Cardiff **U**niversity **Bi**tcoin **D**atabase (CUB*i*D) User's Manual

Dr. Hossein Jahanshahloo

Cardiff University Business School

Autumn 2020

1. Introduction

CUBiD was developed in Cardiff Business School by Dr H. Jahanshahloo to remove the complex technological barriers for academics, regulators, and practitioners, who are interested in utilising the wealth of information in the Bitcoin network data. CUBiD is an easily accessible and user-friendly platform that can be used by anyone independent of their IT skill or programming and database expertise. CUBiD's aim is to make the Bitcoin network data accessible to everyone.

Although the Bitcoin network data can be accessed via different websites, APIs, and is publicly available, collecting and utilising it requires to overcome a significant technological barrier. Collection of the Bitcoin network data and organising it in a structured format requires introduces complex problems in collecting, cleaning, checking, validating, and organising the data from an unstructured format to a structured format. Due to these reasons, the full potential of Bitcoin network data is underutilised by academics, regulators, and practitioners. CUBiD presents Bitcoin data in a structured format and paves the way for full utilisation of the wealth of Bitcoin network data.

CUBiD consists of two data layers: the first layer (layer 1) is the core Bitcoin network data and the second layer (layer 2) is a post-processed data based on the data in the layer 1. The Bitcoin network data (layer 1) consists of three tables, Block Header, Transactions, and Transaction Details. To further assist its users and to reduce their computation time and requirement, CUBiD offers the layer 2 data. Layer 2, currently, includes 4 tables that provide further information on blocks, addresses, and wallets activities.

The point that should be emphasized here is that, should a user require help with their data analysis, CUBiD offers specialised in-house advice and tailor-made solutions to all its users. Thus, CUBiD users can confidently focus on their application of the database knowing that the technical side of the database will be taken care of, should they require further assistance.

2. The Data

Please note that all the values in CUBiD are in units of Satoshi. 1 Bitcoin is equal to 100,000,000 Satoshi.

2.1. Layer 1 – The Bitcoin Network Data

The first layer of the database consists of 3 tables Block Header, Transactions, and Transaction Details.

2.1.1. The Block Header Table

This table includes the following fields on all the blocks' headers.

- *Block_No*
  - The block's height (block number).
- *Block_Hash*
  - The hash of the block.
- *Date*
  - The date that the block is mined.
- *Time*
  - The time that the block is mined.
- *Number_Transactions*
  - The number of transactions that are included in the block.
- *Block_Reward*
  - The reward that the miner should have received for mining this block. This field is in units of Satoshi.
- *Block_Reward_Claimed*[1]
  - The amount of Bitcoin that the miner received for mining this block. This number is different from *Block_Reward* in some cases because sometimes miner do not claim the mining reward or claim less than what they should have claimed. This field is in units of Satoshi.
  - This amount is calculated as the difference between the block's coinbase transaction output value and the block reward.
  - For instance, the reward for mining block 526591 should have been 1250,000,000 Satoshi, while the miner only claimed 625,000,000 Satoshi. Other two examples are blocks 124724 and 501726 that the miners should have claimed 5,000,000,000 and 1,250,000,000 Satoshi while they claimed 0 and 4,999,999,999 satoshi, respectively for these blocks.
- *Total_Input*
  - The total input amount for all the transactions included in the block. This field is in units of Satoshi.
- *Total_Output*
  - The total output amount for all the transactions included in the block. This field is in units of Satoshi.
- *Trans_Fees*
  - Amount of transaction fees that should have been rewarded to the miner for calculating the hash for this block. This amount is also equal to the *Total_Input* minus *Total_Output*. This field is in units of Satoshi.
- *Trans_Fees_Claimed*[2]
  - The amount of transaction fee that the miner has claimed for mining this block. This number is different from *Trans_Fees* in some cases because sometimes the miners claim less than what they should have claimed. This field is in units of Satoshi.

[1] The difference between Block_Reward and Block_Reward_Claimed is the amount of Bitcoin that have gone out of circulation.

[2] The difference between Trans_Fees and Trans_Fees_Claimed is the amount of Bitcoin that have gone out of circulation.

  - This number is calculated as the difference between the block's coinbase transaction output and the block reward. The miner receives the block reward and transaction fees for the block in the coinbase transaction. Therefore, in case that the output value of the coinbase transaction is less than the summation of block reward and transaction fees, the assumption is that the miner has not claimed the full transaction fee that they should have claimed.
  - One can argue that the miner might have not claimed the full block reward. This is a valid argument but unlikely[3]. If one wants to construct this amount in this manner and assume i, they can find the total output of the coinbase transaction for the block by adding *Block_Reward_Claimed* and *Trans_Fee_Claimed*[4].
  - For instance, in block 626205, the total amount of coinbase transaction is 1,276,843,204 satoshi. The reward for this block is 1,250,000,000 Satoshi and the amount of fees for the transactions in this block is 26,843,272. The summation of the latter two numbers is 1,276,843,272 Satoshi that is 68 Satoshi more than the coinbase transaction output.
- *Prev_Block*
  - The hash of the previous block.
- *Merkle_Root*
  - The Merkle tree creates a short key that allows verification of transactions included in the block and provides a way to verify the entire blockchain (and therefore proof of work) on every transaction. A Merkle root is created by hashing together pairs of transaction IDs, which provides a short yet unique fingerprint for all the transactions in a block. This Merkle root is then used as a field in a block header, which means that every block header will have a short representation of every transaction inside the block.
- *Nonce*
  - The nonce in a bitcoin block is a 32-bit (4-byte) field whose value is adjusted by miners so that the hash of the block will be less than or equal to the current target of the network. The rest of the fields may not be changed, as they have a defined meaning.
- *Total_Transaction_Weight*
  - This is the sum of all the transactions' weights in the block.
- *Block_Size*
  - Total size of the block in bytes.
- *Version*
  - Block version related to protocol proposals underway
- *Bits*

2.1.2. The Transactions Table

This table includes the following fields on all the transactions.

- *Block_No*
  - The block number that the transaction is included in.
- *Trans_Hash*
  - The unique identifier that is used to identify a particular transaction.
- *Date*
  - Date of the Transaction.
- *Time*
  - Time of the Transaction.

[3] Majority of the differences between the coinbase transaction output and the amount that it should have been, are very small. This is more likely because of a rounding error. Since block rewards still do not have enough decimal digits (in units of Bitcoin) to induce any rounding error, the assumption here is that the difference is due to unclaimed transaction fee.

[4] Clearly, this can be also found from the transaction table as well, however, this way is faster and more convenient regarding the search speed.

- *Total_input*
  - The total amount that was sent in the transaction. This field is in units of Satoshi.
- *Total_Output*
  - The total amount that was received in the transaction. This field is in units of Satoshi.
- *Fee*
  - Total fees paid to process this transaction. This field is in units of Satoshi.
- *Reward_For_Block*
  - This field is equal to the block number if this transaction was a coinbase transaction and empty otherwise.
- *Number_Input_Addr*
  - Number of addresses from which the bitcoins were sent.
  - The number of inputs in the transaction.
- *Number_Output_Addr*
  - Number of addresses that received bitcoin.
  - The number of outputs in the transaction.
- *Replaced_By_Fee (RBF)*
  - Refers to a method that allows a sender to replace a "stuck" or unconfirmed transaction with a new one that uses a higher fee. This is done to make sure a transaction confirms as quickly as possible. The "replacement" transaction uses the same inputs as the original one. This is not considered a double spend, as the receiving address(es) typically remain the same.
- *Lock_Time*
  - The transaction lock time is the time at which a particular transaction can be added to the blockchain. This is the earliest time that miners can include the transaction in their hashing of the Merkle root to attach it in the latest block to the blockchain.
- *Trans_Size*
  - Total size of this transaction in bytes.
- *Trans_Weight*
  - A measurement to compare the size of different transactions to each other in proportion to the block size limit.
- *Trans_Index*
- *Version*

2.1.3. Transaction's Detail Table

This table includes the following fields on all the transactions.

- *Block_No*
  - The block number that the transaction is included in.
- *Trans_Hash*
  - The unique identifier that is used to identify a particular transaction.
- *Trans_Row*
  - A sequential integer number identifying each row in the transaction.
- *Input_Address*
  - The addresses from which the money was sent.
- *Input_Value*
  - The amount of Bitcoin that was sent from the address. This field is in units of Satoshi.
- *Output_Address*
  - The addresses(es) to which the money was sent.
- *Output_Value*
  - The amount of Bitcoin that was received by the address. This field is in units of Satoshi.

2.2. Layer 2 – The Calculated Data

CUBiD layer 2 data consist of tables of post-processed data based on layer 1 data. While the Bitcoin network structure and thus the tables in layer 1 do not change, the layer 2 data tables are constantly updating, and new tables are and will be added based on the requests and feedbacks from CUBiD's users.

2.2.1. Address Information Table

This table provides an overview on all the existing addresses in the Bitcoin network. When a new block is mined, the new addresses are added to this table and the existing addresses will be updated. This table includes the following fields on all the addresses.

- *Address*
- *Number of Active Blocks*
  - Number of blocks that this address was either a receiver or sender of Bitcoin.
- *First Active Block*
  - The block number of the first block that this address has been involved in a transaction.
- *Last Active Block*
  - The block number of the last block that this address has been involved in a transaction.
- *Number of Transactions*
  - Number of transactions that this address was either a receiver or sender of Bitcoin.
- *Number of Sent Transactions*
  - The number of transactions that the address was the sender.
- *Number of Sent Transaction Blocks*
  - The number of blocks that this address has been involved as a sender of Bitcoin.
- *First Sent Transaction Block*
  - The block number of the first block that this address has been involved as a sender of Bitcoin.
- *Last Sent Transaction Block*
  - The block number of the last block that this address has been involved as a sender of Bitcoin.
- *Number of Received Transactions*
  - The number of transactions that the address was the receive.
- *Number of Received Transaction Blocks*
  - The number of blocks that this address has been involved as a receiver of Bitcoin.
- *First Received Transaction Block*
  - The block number of the first block that this address has been involved as a receiver of Bitcoin.
- *Last Received Transaction block*
  - The block number of the last block that this address has been involved as a receiver of Bitcoin.
- *Final Balance*
  - Final balance of the address. This field is in units of Satoshi.

- *Total Sent*
  - Total amount of Bitcoin that has been sent from the address. This field is in units of Satoshi.
- *Total Received*
  - Total amount of Bitcoin that has been received by the address. This field is in units of Satoshi.

2.2.2. Address Balance by Block Table

This table reports each address's balance after its activities (i.e., sending and/or receiving bitcoin) in each block. When a new block is mined, the new addresses are added to this table and the existing addresses' transaction history will be updated. This table is calculated in an ascending order of blocks. This table includes the following fields on all the addresses.

- *Address*
- *Block_Number*
  - The block number that the address was active in.
- *Total_Sent_Received*
  - Total amount that has been sent from this address (shown as a negative number) or received by the address (shown as a positive number) in this block. This field is in units of Satoshi.
- *Balance*
  - The balance of the address after all the activities up to and including this block. This field is in units of Satoshi.

2.2.3. Address Activity

This table report the number of addresses involved in each block. This table gets updated whenever there is a new block. When a new block is mined, the statistics regarding active, zero balance, and new addresses are added to this table. This table includes the following fields for each block.

- *Block_Number*
- *Active_Addresses*
  - The distinct number of addresses that have been involved in the block.
- *Zero_Balance_Addresses*
  - The distinct number of addresses that have spent all their Bitcoins in this block and have balance of zero[5].
- *New_Addresses*
  - The distinct number of addresses that did not exist before this block and have been involved in a transaction for the first time in this block.

2.2.4. Wallet Identification (Address Linkage) – Available in the Next Release

This table provides a linkage between different addresses that are assumed to be part of the same wallet. For identifying different wallets, Union-Find algorithm is used. When a new block is mined, the new

[5] These addresses might receive more bitcoin in the next blocks.

addresses are added to this table and new wallets will be created and existing wallets will be updated. This table includes the following fields on all the addresses.

- *Wallet_Number*
  - A unique identifier for the wallet
- *Number_Addresses*
  - Number of Addressed that are part of the wallet.
- *Number of Active Blocks*
  - Number of blocks that this wallet was either a receiver or sender of Bitcoin.
- *First Active Block*
  - The block number of the first block that this wallet has been involved in a transaction.
- *Last Active Block*
  - The block number of the last block that this wallet has been involved in a transaction.
- *First Sent Transaction Block*
  - The block number of the first block that this wallet has been involved as a sender of Bitcoin.
- *Last Sent Transaction Block*
  - The block number of the last block that this wallet has been involved as a sender of Bitcoin.
- *First Received Transaction Block*
  - The block number of the first block that this wallet has been involved as a receiver of Bitcoin.
- *Last Received Transaction block*
  - The block number of the last block that this wallet has been involved as a receiver of Bitcoin.
- *Balance*
  - The final balance of the wallet. This field is in units of Satoshi.
- *Wallet_Sent*
  - Total amount that has been sent from this wallet. This field is in units of Satoshi.
- *Wallet _Received*
  - Total amount that has been received by this wallet. This field is in units of Satoshi.