Entity Flow Data Generation Process

Team CryptoQuant
team@cryptoquant.com

June 2021

Abstract

CryptoQuant aims to provide various on-chain data in large blockchain networks, which allows us to build an in depth understanding of markets that can lead to robust trading strategies. We provide our on-chain data as charts via our platform CryptoQuant.com for technical analyst (retail traders) but also their granular data via REST API in real time. Entity flow data is one of our premium data that allow traders to view comprehensive aspects of entire blockchain ecosystem like financial statements in stock market. One can wonder that how we collect addresses to label as entities to compute entity flows and validate it. This process is made up of distinct subprocesses including seed address collection, clustering, and etc. We show below how we do the process and carefully manage entire process to enhance data quality.

1 Background

CryptoQuant analyzes every single transaction on the Bitcoin and Ethereum blockchain networks evoking $16.5 billion transaction-value every month. Outputs of our analysis can be one of the followings: 1) network data that relates blockchain network itself such as active address count, 2) market data related to price data evoked by cryptocurrency exchanges, 3) entity flow data that summarize money flow movement among the most important players in the network such as exchanges and miners. We emphasize that entity flow data are the data which should be looked at by professional traders since they may unveil mysterious price movement (volatility) of cryptocurrencies by looking at major players' money flow. Making these concepts into meaningful metrics is the one our clients will wonder and we in this article show how we build the process to output meaningful entity flow data and manage entire process to keep enhancing data quality.

To be more specific of our entity flow data, CryptoQuant currently tracks on-chain flow metrics for Bitcoin (BTC), Ethereum (ETH), and Stablecoins (ERC20-based). For example, knowledge of exchange addresses allows us to determine the supply held by exchanges and to calculate on-chain flows—i.e., the transfer of native units—into or out of addresses associated with exchanges. Measuring flows and supply held is an imperfect science so caution must be applied in their application to trading or other uses cases, particularly in the early periods of their release.

We organize this article into the following parts:

1. Data collection: how we collect labeled addresses to compute metrics
2. Data serving: how we serve our data efficiently and securely
3. Data validation: how we ensure that we have valid labeled addresses

Before we dive into the main sections, we briefly explain key concepts to understand how we compute entity flow metrics, as followings.

Table 1: Key concepts and terms that are related to the process of entity flow calculation.

Concept	Description
Address (Wallet)	A pseudo-anonymous identifier that is used to receive and send funds on the blockchain network
Entity	A set of addresses that are controlled by a single user
Clustering	A method to identify addresses that belong to the same entity from seed addresses
Reserve	An amount of coins (or tokens) that an address (or entity) has

2 Data Collection

Our data collection procedure is consisted of three parts: seed address collection, basic clustering, and advanced clustering. First we collect seed addresses that belong to certain entities such as exchanges and miners. Then we do basic clustering based on the seed addresses we collected to identify other addresses that belong to the same entities of the seed addresses. Clustering should be done differently for coin (or token) by coin based on unique properties of each blockchain. More than that, we do advanced clustering based on the clusters we found by analyzing interactions among known clusters and unknown clusters and each cluster itself, where the techniques are based on machine learning and graph analysis.

2.1 Seed Address Collection (Dusting)

2.1.1 Exchange

Dusting In the language of cryptocurrencies, the term dust refers to a tiny amount of coins or tokens - an amount that is so small that most users don't even notice. Taking Bitcoin as an example, the smallest unit of the BTC currency is 1 satoshi (0.00000001 BTC), so we may use the term dust to refer to a couple of hundreds of satoshis. Within cryptocurrency exchanges, dust is also the name given to tiny amounts of coins that "get stuck" on users' accounts after trading orders are executed. We use this term as depositing a small amount of money(about $100 value) into exchanges to find out starting point for clustering.

Labeling Our wallet labeling procedure for exchanges is basically based on dusting and dusting is periodically done as time goes by because exchanges might change their wallet addresses intentionally for some reasons such as security issue. We collect useful data which can be used for tagging by exploiting exchanges as a service. These data include deposit addresses or transaction hashes. We systemically manage tagging information as periodic dusting is done.

Auto-dusting The following items are automatically, periodically executed on our system through exchange API keys:

1. Deposit cryptocurrency into given exchange
2. Trade part of the cryptocurrency deposited for another token
3. Withdraw cryptocurrency from given exchange
4. Trace the flow of funds on the blockchain from wallet to wallet while performing each of the above steps and record
5. Repeat all of the above steps multiple times during different points of the day to validate if similar graph patterns emerge

2.1.2 Mining Pool

We get labeled seed addresses from mining pools by parsing coinbase scripts in coinbase transactions. Mining pools typically write down their identities to coinbase transactions to show their mining power in their input scripts. We parse those identities and specify coinbase addresses to the identities (entities). In this way, as a result (after clustering applied), from our analysis we can cover almost 98% of all addresses in coinbase transactions.

2.2 Basic Clustering

BTC We collect other addresses that belong to the same entity by basic clustering based on dusted addresses. We basically collect all the addresses that are considered to be owned by the entity through heuristic clustering. Specifically, We use only co-spent heuristics, which means input addresses in transactions are belong to the same entity. This assumption may fail when *CoinJoin* is applied to the transactions and we carefully ignore those transactions by exploiting patterns of *CoinJoin* transactions. The reason why we only use this algorithm is that other heuristics such as change address heuristics are far less accurate than co-spent heuristic is, which possibly will create super-clusters. And the existence of super-clusters will reduce the accuracy of the data since they could ignore possible in and out flows from entity to entity.

ETH and ERC20-based stable coins Based on seed addresses we collected from dusting, we investigate all addresses that had any transaction associated with the seed addresses. In case for exchanges, one clear pattern that we discovered is that the amount of coins that a candidate address is given would transfer to the seed address after certain amount of time, which means that the candidate address is controlled by the same entity of the seed address. This pattern can be applied as an algorithm to make robust cluster for Ethereum. It is rather clearer to decide an address should be included to a cluster or not by looking at the transaction pattern of the address in the case of Ethereum's transactions, compared to Bitcoin's. This is because an Ether transaction shows who is the sender and is the recipient and there is only one of each. However, as policies of managing its own addresses will vary with entities, we keep on research and investigating patterns of transactions of entities.

2.3 Advanced Clustering

Based on basic clustering results, we do advanced clustering by analyzing interactions among known clusters and unknown clusters and each cluster itself, where the technologies are involved as machine learning and graph analysis. In advanced clustering, we basically tag clusters with high probability of belonging to the same entity through graphical meaningful statistics (e.g. distance) or analysis of transaction patterns among clusters. We take careful steps to make a decision right for tagging unknown clusters since false positive clusters could be included in our data.

3 Data Serving

Data serving is about how to serve our data to client. This is consisted of three parts: building infrastructure, security, and time delay to calculate data. First we get raw data from each node or 3rd party platform. Then we extract meaningful data from it, and insert data to database.

3.1 Building Infrastructure

3.1.1 Server

AWS There are a lot of different type of servers to serve data as a service. Among them, we use cloud platform that is easily scalable, fast, and low-cost. AWS(Amazon Web Services) is one of the most stable cloud platform service in the world, and we use below services in AWS.

- Elastic Compute Cloud(EC2)
- Elastic Beanstalk(EB)
- Relational Database Service(RDS)

Mostly, our server is located on California to deliver data as fast as we can to customers, where our customers are mainly located in United Stated. We will spread our data center to global. Our service takes auto-scaling system for stable data center. Elastic Beanstalk and RDS has 2 servers at minimum in case of the servers fail, and we take snapshots every single day to restore the data and service immediately.

3.1.2 Node

Bitcoin We built our multiple bitcoin nodes in United States distributed intensively to make sync node as fast as we can. Also, we get bitcoin on-chain data through *JSON-RPC* which is connected to node directly. Some nodes are used to calculate network, market data, and others are used to calculate entity flows. Network, market data just call *JSON-RPC* because the data is simple, but for entity data, we use *RocksDB* for speed. Entity flows data need much more resources to check its clustering, so we reorganize bitcoin on-chain data for clustered one.

Ethereum and ERC20-based stable coin We built our multiple Ethereum nodes in United States distributed intensively to make sync node as fast as we can.

4 Security

4.1 Hacking and DDoS Attack

We basically use CloudFlare to protect our service from hacking and DDoS attack. More than that, we are developing advanced DDoS protection script, and deploying very soon. For authorization, we are using Bearer token method.

4.2 Time Delay to Calculate Data.

Before describing each blockchain coin's data, we want to notice that most of the data need to be combined with price to calculate it. We get price data from most influential exchanges, and calculate average price of those. This procedure takes two or three minutes because we use close price data for accurate one. Also, it can be possible not to get data through API because of shutdown or maintenance, so we check price data multiple times. Also, for every data from our service, we give head time as time. For example, if there is hour data with **2020-01-01 00:00:00** time, it means the data from **2020-01-01 00:00:00** to **2020-01-01 00:59:59**.

4.2.1 Bitcoin

Bitcoin data have three kinds: network, market, and entity flow data. Each of it has different time-window units (e.g. block, hour, day). We describe time delay to serve data for our customers for each time-window below.

Time delay for block data

- **Network/Market Data**: We check bitcoin node every 10 seconds, and the data are calculated and combined with the price data. For calculating on-chain data, it takes 40 seconds at most. Total amount of time combined with price takes 2.5 minutes to 5 minutes.

- **Entity Flow Data**: We check bitcoin node every 5 minutes, and the data are calculated and combined with the price data. For calculating on-chain data, it takes 5 minutes at most. Total amount of time combined with price takes 6 minutes to 15 minutes.

Time delay for day and hour data

- **Network/Market Data**: We check Bitcoin block data every 30 seconds, and aggregate data from the block data. This calculation takes 10 minutes at most. Total amount of time combined with price takes 5 minutes to 10.5 minutes.
- **Entity Flow Data**: We check Bitcoin block data every 5 minutes, and only aggregate data from the block data. Total amount of time combined with price takes 7 minutes to 15 minutes.

We are about to provide day data to client at exact midnight without delay. However, the data cannot be precise because we cannot be sure that the block data is completely fetched before midnight due to randomness of arrival of block data. Thus, it will be continuously updated if the block data is inserted. Of course, if finished to calculate precise day data, then it is not changed after it. We will announce this new feature as soon as finishing development.

4.2.2 Ethereum and ERC20-based stable coins

Ethereum data now has only entity flow data. It has different units like bitcoin. For block data, delay time is like below one.

Time delay for block data

- **Entity Flow Data**: For calculating on-chain data, it takes 1 minute at most. Total amount of time combined with price takes 4 minutes to 6 minutes.

Time delay for day and hour data

- **Entity Flow Data**: We check Ethereum block data every 1 minute, and aggregate the data. This calculation takes 5 minutes at most. Total amount of time combined with price takes 4 minutes to 6 minutes.

5 Data Validation

DISCLAIMER. You never know which addresses that the entity (exchange/miner) has correctly unless you ask to the entity directly. However we try our best to validate our data in indirect ways.

To validate how accurate our entity flow data are, we indirectly evaluate our date in several ways. Followings are the ones we evaluate frequently and periodically and each analysis covers its own evaluation aspect so that we comprehensively validate how accurate our data are.

5.1 Reserve Analysis

For some exchanges, there is a place to disclose the balance (reserve) of cryptocurrency every cycle (for example, Korean exchanges). Using this information, we can verify the accuracy of the data indirectly by comparing it with our balance. In Korea, cryptocurrency services are being institutionalized, and this verification method is expected to increase further. The example analysis for the exchange GOPAX is shown in Figure 1.

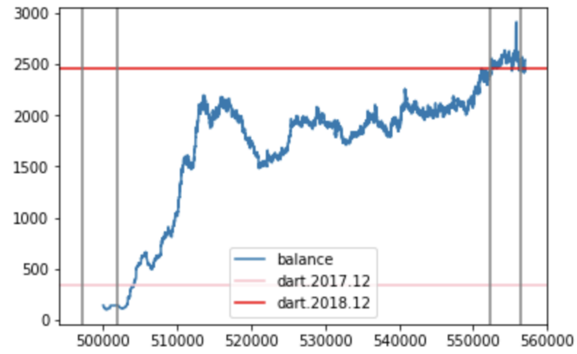

Figure 1: Result of comparing the public reserve of GOPAX with the total reserve of the addresses we collected.

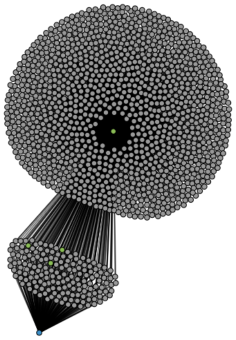

Figure 2: Bitfinex's two-month Tx results

5.2 Trend Analysis

We indirectly measure how complete our addresses for entities are by seeing trends of inflow, outflow, and reserve in time. This analysis let us to find abnormal flows caused by incomplete set of addresses, which we call noise in data, so that we can add missing links to the set to make complete labeled addresses. This process is periodically done for sustainable data quality.

5.3 Graph Analysis

Blockchain can be understood as a single large network like any other usual network such as social network service. Thus general graph analysis methods including machine learning could be the good way to understand the association between the collected addresses. Moreover there could be any chance to label unknown clusters by linking from existing labeled clusters. One example of our analysis can be visualizing such entity's cluster in address level to get structural intuition of how the entity manages wallets, where Bitfinex case is shown in Figure 2.

6 About CryptoQuant

CryptoQuant is an on-chain data provider for cryptocurrency focused funds and investors. Utilizing clustering and AI, we identify the addresses of the most important players including exchanges and miners in blockchain networks. This allows our clients to optimally navigate and profit from cryptocurrency markets.

- Website: https://cryptoquant.com
- API Documentation: https://cryptoquant.com/docs
- Live Charts: https://cryptoquant.com