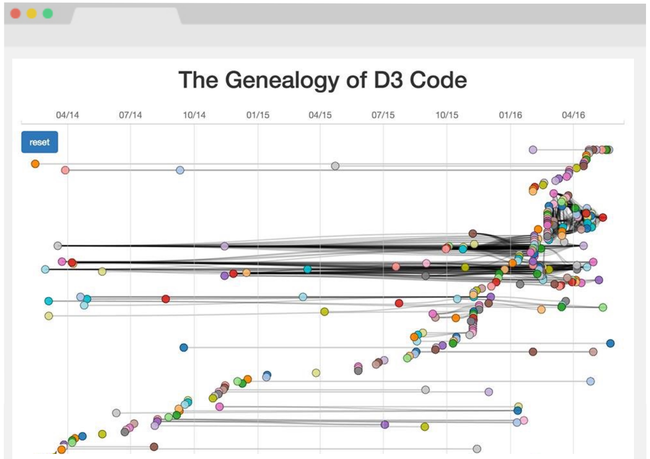
Visualizing D3 Code Ancestry

Tracing the Genealogy of D3 Code

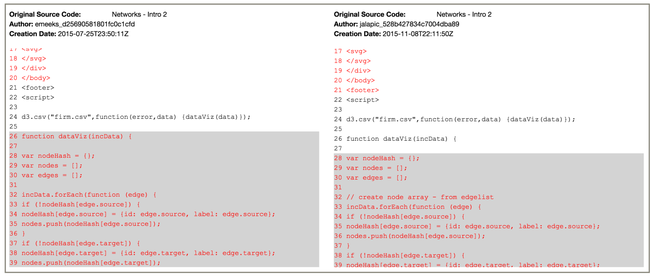
Lucas Throckmorton & Cathy Zhu
CS448b: Data Visualization

Problem / Motivation

Within the D3 community, code sharing is widely encouraged via the form of github gists, publicly available on bl.ocks.org. We would like to investigate the impact of code sharing by identifying primary source examples and tracing how these examples are modified to generate new examples.



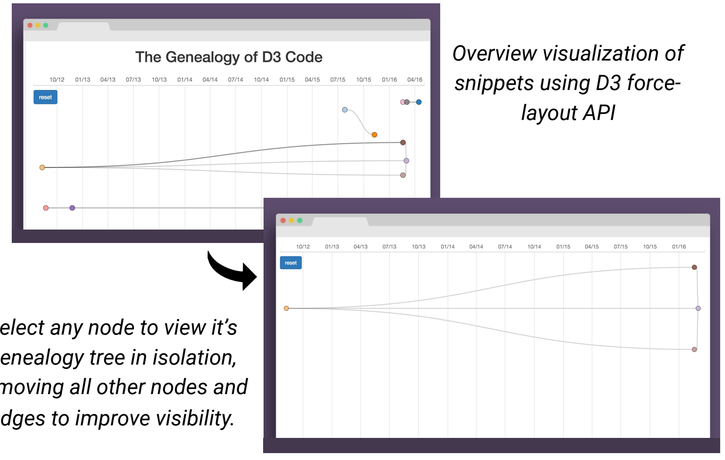
Visualization of 6k randomly sampled D3 code snippets



Selecting an edge displays a meta card, providing information on both of the edge's nodes. This includes a comparison of the code from each snippet, highlighting any identified relationships.

Approach

Data Source: We scraped approximately 6800 snippets of D3 code from github gists and ran them through MOSS, a plagiarism detection tool, identifying links between snippets with more than 10 lines of similar code. We determine ancestry by timestamp, meaning that given two pieces of similar code, we say the code created at an earlier time "influenced" the code created at a later date.



Overview visualization of snippets using D3 force-layout API

Select any node to view it's genealogy tree in isolation, removing all other nodes and edges to improve visibility.

Visualization: The final product is an exploration tool for our dataset, presenting a big picture view in the form of a timeline tree and metadata cards below our main visualization that shows pertinent metadata for each node (e.g. source code, API calls, author, creation time) and link (specific lines copied). We also enable manipulation of the main view via zoom, drag, and selection.

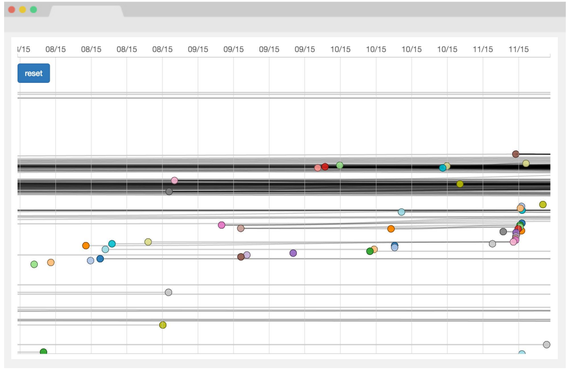
As D3 does not comfortably deal with so much data, we showcase the different functionality on two different datasets. A small dataset on the Force Layout API is implemented with detailed metacards, and a large dataset containing the most recent 6000+ nodes demonstrate zooming, selection, and node manipulation.

Future Work

More Data: Our data is currently stored in a single JSON file imported via JavaScript. Unfortunately, D3 currently cannot handle more than ~800 links worth of data containing raw code at a suitable latency. Hosting this on a server with an actual database would facilitate information hiding, since it would be possible to live query for detailed information rather than front-loading JSON with raw code snippets. It would also enable calculation of per-thread metadata. There are approximately 30k gists available on blocksplorer — an ideal dataset.

Selection Metadata: As mentioned above, we currently support metadata for nodes and links only. We are interested in other selections of the data that may have summary metadata. For example, a single thread of ancestry, or all code snippets associated with a particular author or API call.

Queries and Filters: Another work in progress is implementation of additional selection/filtering criteria (author, timeframe, API calls, or ancestral tree).



Scrolling allows the user to zoom into the visualization to better view busy node clusters.

Source Code: github.com/lthrock/d3-genealogy
Demo: lthrock.github.io/d3-genealogy/