I have tried to analyze the data I have been given in the API, I have visualized part 1 in Task 2, but I did not understand the meaning of *"Chromosome Overview: A visualization that shows the distribution of mutations across the **22 human autosome pairs plus the two sex chromosomes**."* And how should I visualize it. However I can provide the idea how we can add filters and make the visualizations synchronized and interactive.

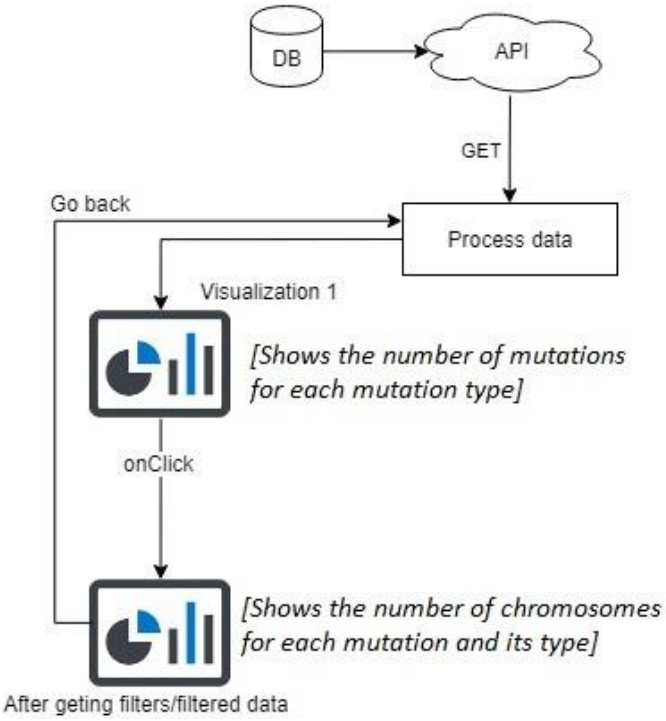
In D3.js, after the basic visualizations are plotted, then we need to append the onclick events to our visualizations and we just need to pass the filtered data or filters which should be analyzed as per requirements to the function(s) inside the onclick events; i.e. after a visualization 1 is clicked the appended function (in onclick event) will be having the logic and the data along with the filters which can be passed to visualization 2 and visualization 2 will show the filtered data based on the filters provided by visualization 1.

Due to short period of time I used highcharts, however I also plotted the data using D3.js, but for 1 level only. Then I shifted my logic into highcharts.

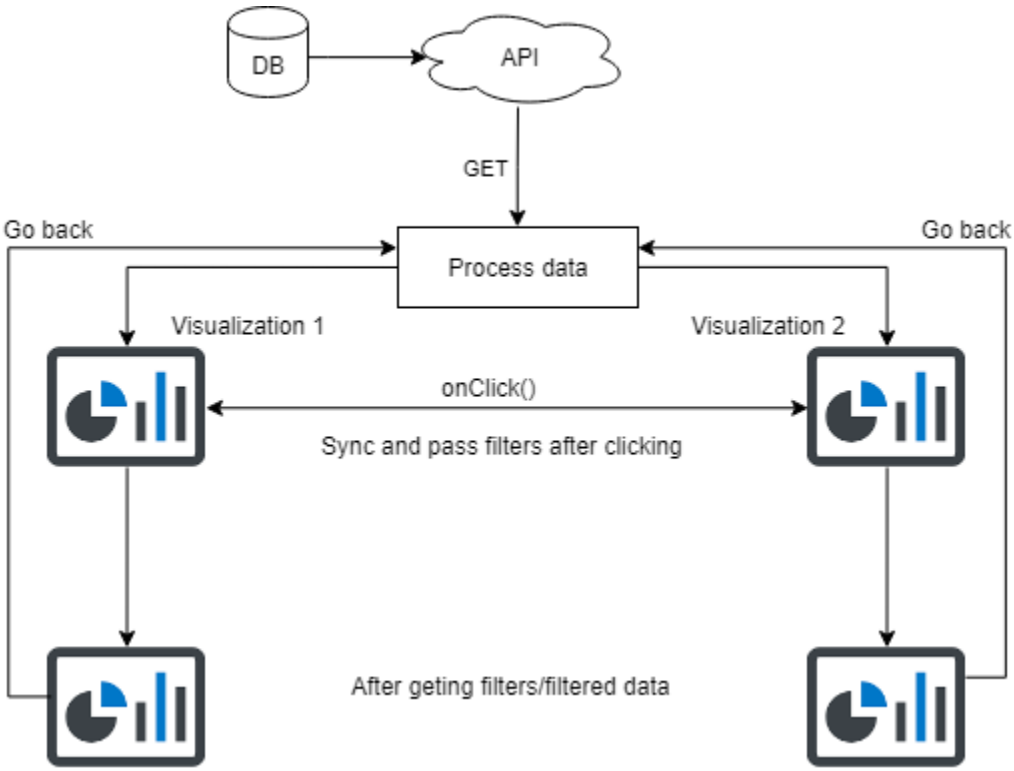
To work and understand more deeply in the powerful library of D3.js, I have also bought the license to the Lynda tutorials, because I love to play with the data and to visualize it. I have worked a lot with different visualization libraries but I have less experience with interactive visualizations, but in a very short time span I will have a very good hold on it, after I am provided with a platform to work on. As the level of understanding increases very fast and easily if there is motivation and interest involved in it.

To test the system we can provide unit tests that will give us a report generated by the JavaScript unit testing frameworks like (Jasmine, Mocha, Tape, Jest etc.)

The application I have developed is working as follows:



I am not able to complete the task but I would be able to approach it the way I understood is as follows:



Task 3

After the research now I am able to propose a design and some set of rules for tabular data that would be able to manage every type of data and also able to manage huge amount of datasets.

However, data is meaningless without the ability to visualize and act upon it. The companies that survive the next decade will not only have superior data; they will have a superior user experience.

Good data tables allow users to scan, analyze, compare, filter, sort, and manipulate information to derive insights and commit actions. A list of design structures, interaction patterns, and techniques are as follows:

Fixed Header and Horizontal Scroll

Fixing the row header as a user scrolls provides context on what column the user is on. Horizontal scrolling is inevitable when presenting large datasets. It is good practice to place identifier data in the first column. As an advanced feature, enable individual locking of columns so users can compare data with multiple anchoring identifiers.

Resizable columns

It is one of the important features which helps by resizing columns allows users to see abbreviated data in full. It also helps the user incase user want to hide a particular column(s) that the user don't want to see.

Row Style *(Zebra Stripes, Line Divisions, Free Form)*

The row style helps users scan data. Reducing visual noise by removing row lines or zebra stripes works well for small datasets. Users may lose their place when parsing larger datasets. Line divisions help users keep their place. Alternating rows (*aka zebra stripes*) help users keep their place when scanning long horizontal datasets. Although they cause usability problems when there is a small number of rows because users may ascribe meaning to the highlighted rows.

Visual Table Summary

A visual data summary provides an overview of the accompanying table. It allows the user to spot patterns and issues in aggregate before taking any action or insights. It gives the overview of the current situation or available data.

Pagination

Pagination works by presenting a set number of rows in a view, with the ability to navigate to another set.

Hover Actions

Presenting additional action when a user hovers reduces visual clutter. However, it can cause discoverability issues because the user needs to interact with the table to expose the presentation of actions.

Inline Editing

Inline editing allows the user to change data without navigating to a separate details view. That should update instantly and the effect should be reflected in the visualizations.

Expandable Rows and Quick View

Expandable rows allow the user to evaluate additional information without losing their context. Quick View is much like expandable rows, it enables a user to view additional information while staying in-context.

Row to Details

Clicking on a row link transforms the table into a view with list items on the left and additional details on the right. It enables a user to parse large datasets, as well as reference many items without losing their place.

Sortable Columns

Column sorting allows users to organize rows alphabetically and numerically. Which enables a user to easily get to the data what he desires for.

Basic Filtering

Basic filtering allows users to manipulate the data presented in the table.

Filter Columns and Customizable Columns

Filter columns design pattern allows users to assign filtering parameters to specific columns. While customizable columns feature enables users to pick the columns they want to see and sort accordingly. The feature may include the ability to save presets for later use.

Searchable Columns

This design pattern allows a user to search specific values within each column.

Add Columns

This pattern allows users to add columns from a dataset. It is a way to keep the table's data limited to essential information and enables the user to add additional columns based on their use case.

Therefore, the above proposed design and interaction patterns shows us that how can we manage and visualize any type of data that is available. No matter how big the table is the design patterns above will

handle the problem. If we provide the opportunity to interact with the visualized data, by filtering columns, customizable columns, add/remove columns, searchable columns, Inline editing, sortable columns, and visual table summary, it will surely lower/remove the issues and changes that occurs over time.