WebGen (VIP 496) Log - Spring 2023

Amara Im | keoamara.im@stonybrook.edu
B.S. in Computer Science, Applied Mathematics and Statistics
Stony Brook University, Class of 2024

Team Members:

Richard Moffitt, PhD
Jonas Almeida, PhD
Adit Anand
Amara Im
Alisa Yurovsky
Moinak Bhattacharya
Jiaolingzi (Lindsay) Kong
Nimisha Papineni
Rekha Sathian
Pallavi Surana

Week 2 - 2/1

- Subteam meeting with Adit (Saturday 2/4)
  - Caching for clinical data
  - main.js in smartcache
  - Data structures
- Any Clinical data

```
{       cohort: string
        clinical_data:[ {barcode : string
                         xxx : unknown
                         yyy : unknown} ]
}
```

- Use cases:
  - Fetch all clinical data for a [cohort]
  - Generate a pie chart or histogram from [cohort][feature] -> x.filter
  - Get list of features that are available for a [cohort] -> keys(x.filter)
  - Generate a vector of {barcode,feature} for heatmap/violins [cohort][barcodes][feature] -> x.filter[cohort].filter[barcodes].map{barcode,feature}
  - Given a pie chart selection [feature][feature_selected_value], return a list of barcodes -> x.filter[cohorts].filter[feature,feature_selected_value]

Week 3 - 2/8

- Adit:

Week 4 - 2/15

- Amara did not attend.

Week 7 - 3/8

- Questions:
  - does clinical data need gene parameter or it is general to the entire cohort
  - Indexed by cohort and barcode
  - Make caching function more generalized and passed in data types (i.e. mutation, clinical, barcode) so that we don't have to make 3 different caching functions

Week 8 - 3/22

- Questions:

- does clinical data need gene parameter or it is general to the entire cohort
- Indexed by cohort and barcode
- Make caching function more generalized and passed in data types (i.e. mutation, clinical, barcode) so that we don't have to make 3 different caching functions

getCacheXYZ()

if a cache exists, return it; otherwise create it and return it

this.fetchWrapperXYZ(lists)

Operates on a cache. The primary function to be used in order to retrieve data, either from the cache or from a new query.

this.constructQueriesXYZ(lists , cache)

Determines which (of the listed) data are already in the cache

Or else need to be retrieved from a query

Returns two respective lists (without data in them)

this.executeQueriesXYZ(list of missing)

Fetches any missing data via API helper function:

fetchXYZ(list of missing)

Chops up the requests in to smaller api calls

Executes the API fetches

Concatenates the just-fetched-data together

Saves the just-fetched-data to the cache

Retrieves previously existing data from the cache

Appends found data and just-fetched-data for return