Mr. Hoyin Chu

NIAID Data Scientist Fellow

Civic Digital Fellowship, Fall 2020

Mentor: Dr. Christopher Steven Marcum, Office of Data Science and Emerging Technologies

The database of Genotypes and Phenotypes (dbGaP) is a popular platform for hosting genomic data from studies supported by the NIH. To access the data contained in these studies, PIs need to submit data access requests (DARs) to the appropriate data access committee (DACs) for review. Currently, there are few standards on how each DAC should report its activity summary statistics, and the lack of common metrics makes it difficult to know what are the areas each DAC could improve upon in its performance.

Under the guidance of Dr. Christopher Marcum, we developed an R package named “DACReportingTool” which automatically generates a DAC activities summary report, complete with summary statistics, figures, and draft interpretations for any given time interval. The package can be divided into four components: a web scraping component, a data storage and update component, a data analysis component, and a reporting component. The web scraping component consist of functions in files with names that start with “Request”. The main role of this component is to either scrape content from a specific webpage or make API calls, the results of which are parsed into an R dataframe. The data storage and update components are responsible for storing the scraped data in an internal proprietary format (i.e. rda files), and to incrementally update these locally stored data or otherwise overwriting them should the user so desire. The data analysis and visualization component refers to the functions that start with the names “Get”, and they convert the DAC data into summary statistic tables that are easily interpretable and accessible by the user. These tables are subsequently interrogated in the reporting component. Visualizations, summary statistics, and draft interpretations are hard-coded within rmarkdown template that can be compile by the user to generate a shareable report in MS ord or HTML format. When the user calls the appropriate function, the package will produce a document with rich data visualization that can be customized to their needs.

In addition to the components described above, some experimental work was also done to expand on the use cases for the package as a research tool. For example, the development branch of package contains functionality that allows the user to construct a network where the nodes represent studies, and two studies are connected by an edge if they have been requested by the same PI. By constructing these networks, one could identify clusters of studies that are unexpected, and this information can be used to identify research gaps that the NIH can fill.

During the development of the package, open-source practices such as version control, documentation and continuous integration were followed. The package is available as a github repository (https://github.com/codingitforward) and development effort can be picked up by cloning the repository and making new pull requests.

In conclusion, we developed a package which provides a programmatic interface to retrieve, store, and analyze data access requests using a variety of sources. Furthermore, the package can be used for research purposes, and new functionalities can be added to the package with ease. We expect the package to become a new standard for data access reports, and ultimately be adopted by DACs across the NIH to automate components of the reporting workflow.