Week of June 6-10

1. Changed the serialization method used for communications between R and the server from serializeJSON() to toJSON(). This results in much cleaner accessing of result values on the server side since we now use actual string properties to access result values as opposed to numbers.
2. Created the initialization service for the front end controllers. This greatly reduced the amount of code in each controller, and it prevents up from having to write duplicate code when editing or adding features that will be shared across controllers.
3. Moved the neighbor explorer into its own tab in order to make it less confusing for the users.
4. Changed the layout and styling of the sidebar so that buttons are now colored and easily distinguishable from other text. I also added dividers so that users can see that certain sections of the sidebar are responsible for completely differnet things than other sections. Lastly, I added a material design color palette to be used by the whole app in order to make the styling look more consistent.
5. I made a feature that allows users to see all the paths that exist between two genes with a maximum of one hop in between. This resulting table of paths is also downloadable.
6. Fixed a major parsing issue that resulted in command line arguments to Rscript not being parsed properly on unix based systems. It turns out that all I had to do was surround the arugments being passed in quotes.
7. I added a route to enhance security by capturing any request that doesn’t match the specified paths. Additionally, I changed the start location of the http server since it used to start in the directory of the entire application thus giving it access to every file. Now it only has access to the necessary client-side files.
8. Researched several ways of caching correlation matrices in memory so that R doesn’t have to load them from the disk each time a request is made. I first looked at opencpu and spent a decent amount of time creating and R package for it only to find out that each call to opencpu results in a totally isolated R session. Secondly, I considered using R instead of Rscript and I converted all of my scripts into R functions. It took a while for me to realize that this would not be a feasible solution for several reasons. First of all, it would require a separate server running for every user otherwise other users would be waixting while one user’s request finishes. Additionally, it would require me creating a framework to keep track of which request created which output, and also which handler I would have to use to handle that output. Finally, I decided to use sparse matrices are recommended by Benjamin and that solved all of the issues I was having. The amount of memory used by the server is now orders of magnitude less, so we need not worry about running out of memory.