Web App To Do For Weekend of Jun 3-5

# Graph Layouts and Styling

* We should probably add the concentric layout to the neighbor explorer
* We’ll have to see if it makes sense to do so though since right now it is built specifically for showing first and second neighbours.

# R Scripts and Data Retrieval

* Need to see if it is worth it to change the data we are returning from R scripts in an attempt to make things easier to style. For example, creating the random layout for the neighbor-genreal method was a tough since we’re always adding only one source node manually to the graph, and the rest of the source nodes are sort of re-discovered by the script. IT might be useful to instead return objects from R that have certain properties such as which panel a gene should belong to and whether or not a gene is a source node
* It could also be worth it to go from using lists to using data frames. The reason is that jsonlite tends to do the following:

> y <- list(weight = 5, parent = "epi")

> toJSON(y)

{"weight":[5],"parent":["epi"]}

* Notice how the values are for some reason wrapped in arrays. When working with data frames, this phenomnenon no longer occurs

> x = data.frame(weight = 5, parent = "epi")

> toJSON(x)

[{"weight":5,"parent":"epi"}]

* So now the entire objects is encompassed in an array as opposed to its properties. This will make things cleaner on the server side code when it comes to indexing, though not by much.

# Additional Features

* Need to create a script and caching mechanism to be able to have the pre processing stats such as significant interactions and self loops for the entire network like Ventaka suggested. We’ll have to think whether or not caching is worth it. If we add it, it will make our server startup time take at least a minute which I suppose is fine once deployed, but is too slow for testing and dev.

# Front End Angular Changes

* Need to create a service that will initialize all controllers with the variables and methods that they have in common

# Front End Layout and Styling

* We need to make the interface more user friendly by perhaps coloring the buttons, repositioning some elements, adding the legend back in, etc.

# Neighbour Explorer Repositioning

* We should keep in mind that filtering for the neighbor explorer doesn’t really make sense since we would need a filter for every single level that we explore. So we will omit this until later on.
* We need to reposition things over to the Neighbour Explorer tab and get rid of the watch functions
* We need to fix the issue where all of the neighbours are nodes that have previously appeared

\*Need to extract a function for when there are no neighbours of a selected node. We always return the node back on its own.

For the bipartite view, we need to extract a function for the loop that is repoinsible for adding parent nodes.

\*Switch to using OpenCpu instead of R script and see if that has any impact in performance. I am hoping that it will allow us to keep object sin memory so that we don’t have to keep reading in correlation matrices.

Before we create a remote repo on Mordor, let’s first transition to using OpenCpu. Unfortunately, OpenCpu is actually a server. We don’t seem to have much control over where that server exists and which directory is its root directory.

\*\*\*Need to add some kind og security check for when a user uploads files

\*\*\*Add server side check to see if a file has been specified. If not, it crashes our server.

\*\*\*Add some more validation to the R scripts such as checking of lengths and such. I.e. the 1:n syntax in R is stupid if n = 0.

\*We need to add a function that will redraw the graph when a user switches from the graph view to the table view.

\*Made the highlighting of edges when a node is clicked much faster by using the cy.batch() method.

\*Need to add client side validation to check whether or not

\*\*\*Decouple different layouts from having to run R scripts every time. The R scripts should be used only the first time we get the data. Afterwards, applying different layouts should not result in running an R script again and again.