# Week of May 24

Currently working on improving the performance of the app on the client side. Some animations used by angular materials are slowing things down

Furthermore, some of the caculations we are doing for the dictionary of interactions needs to be moved to the server side in order to improve performance.

Making the concentric layout more dynamic and based on the number of nodes. We are now determining at each level the ideal radius. We should probably make the source nodes twice the size of the neighbours so that they are visible even with all of the edges coming out of them.

So we moved the self loop logic to the server, and we moved the function that builds the edgeDictionary to the server as well.

Added a popup for examining edges and seeing their source, target, and weight.

Made changes to the gene locator so that the zooming works more smoothly. Also added an animation to change nodes to green. Added a button to remove the highlighting of nodes if wanted.

Need to make the random layout more dynamic in terms of volume needed to display nodes

Need to make the nodes with a larger degree have a bigger size in the random layout

Need to add a legend for the graph showing what color corresponds with what tissue

\*\*\*mapData

IT looks like we can’t override the same property using a different selector in cytoscape js. Say we have a class selector setting the color of a node, if we had an event more specific selector, or another selector with !important, we would still not be able to override the initial property. One way we have overcome this is by setting the pie property instead of the background-color property. Keep in mind the limitations of this method. First of all, if there isn’t another property that has a similar behavior as the one that we are trying to overwrite, then we are out of luck. Second of all, if we move to using nodes of a different shape, the pie property will no longer make any sense. Therefore, we will eventually have to structure our styles in such a way as to be able to use the toggleClass method. Currently, for the bipartite view at least, the position of the label and the color of the node are all controlled by the same class. Therefore disabling this class when trying to locate a node within the graph will have an impact that is beyond just node color. We need to make a dynamic framework that names things in a clever manner so that we can select specifics via class selectors.

Here is how we are going to have a more rigorous framework when it comes to styling our components with classes on the server side:

1. We will have a list of class suffixes such as color, font-size, label-placement, etc.
2. When we process and create the nodes, we would prepend the apporprite prefix to all of these. For example, all epi nodes will end up having the classes: epi-color, epi-font-size, epi-label-placement, etc.
3. Our styles would then have to work with the appropriate class selectors.
4. This prevents the issue we were having before where one selector was in charge of doing too many things at the same time.