Okay so for the filtering of edges, we need to come up with an intuitive front end to show the user what stage in the process they are at.

1. The user types in some genes of interest and they can click a button called Get First Neighbours. Filters are disabled at this point.
2. Once the graph is returned, they will be presented with the option of filtering based on the correlation. The Get First Neighbours button will turn into the filter button.
3. The new filtered down graph will be returned to the user.
4. The interface will change so that the filters are disabled and the Filter button now turns into the Get Second Neighbours button.
5. So the second neighbours of those filtered down first neighbours are returned, and the filters are now enabled for the second neighbours. The Get Second Neighbours button becomes the Filter button.

The point is that our R scrip needs to be able to filter on first neighbours and on second neighbours separately. This means that when we want to get the second neighbours unfiltered, but with filtered first neighbours, we need to probably use a different copy of the correlation matrix that is not filtered. Otherwise, if we use the same copy, we would end up getting a limited number of second neighbours since we would be using a filtered correlation matrix. The logic goes as follows

We need to fix the layout to be more dynamic by figuring out the sizes of the radii based on the number of nodes in the graph.

We need to create a table out of the interactions. This table will have to be exportable so we’ll consider some existing frameworks that do this.

We need to fix the way in which we obtain degrees for the R scripts. Right now, we are not returning the degrees for any of the selected genes. This can prove to be very dangerous when trying to filter by degrees on the client side. We only need to make some small modifications to the script to return an additional vector of degrees for the selected genes.