So here is why our clustering position method wasn’t setting the properties properly. The firstNodes array is an array of arrays of nodes. So if we call a method on it and try to modify a node, it won’t work because

The point is that allNodes is a flattened array of nodes.

firstNodes is an array of arrays of nodes.

Modifying

TPB

ALAS

CDC

NUP88

HEBP1-E is a first neighbor of CTSL-S

It has duplicate edges with CYTH1-S which is a first neighbor of TERT-E

So we have duplicate edges between genes that are first neighbours of different genes.

Fixed this, but let’s put a little more thought into the exclusions just to make sure that we aren’t missing anything

We might have to reorganize what we return from R. We’ll leave it for now. Looks like our exclusions for nodes are also messed up.

# Verification of Exclusions

## First Neighbours

### Nodes

Node exclusions should be the simplest ones to do since we are basically excluding any genes that we have encountered thus far. This means that we shouldn’t be adding any of the selected genes to the first or second neighbours. Therefore, we initialize the exclusions to be the selected genes.

### Edges

When creating the edges that are going from the selected nodes and to the first neighbours, we only want to ignore the previously seen genes. For example, say the selected genes are Gene A and Gene B. The first neighbours of Gene A are: Gene B, Gene C, Gene E. The first neighbours of Gene B are: Gene A, Gene R, Gene S. We process Gene A first and create edges AB, AC and AE. Then when we create the edges from B, we don’t want to create the edge BA. We only want to create the edges BR and BS. Thus, we need to have A as an exclusion. In fact, if we were to generalize this to more than two genes, here are the requirements:

Suppose a user selects genes: gene-1 to gene-k. Suppose that we are creating the edges for gene-m when 1<m<k. We want to exclude gene-1 to gene-(m-1) since when we were processing either one of these genes, it is possible that gene-m is one of its neighbours and that we would have created the possible edges 1m, 2m,…, (m-1)m. Therefore, we exclude gene-1 to gene-(m-1) so that we don’t end up creating edges: m1, m2, …, m(m-1).

## Second Neighbours

### Edges

When creating edges that are going from the first neighbours of the selected genes, to the first neighbours of those first neighbours, things get a little more complicated. Continuing our previous example, we now want to find the second neighbours of Gene A, which are the first neighbours of: Gene B, Gene C, and Gene E. We don’t wanna create edges from these 3 genes back to Gene A since those edges already exist. Now what happens if Gene C is a neighbor of Gene B? Well, then we have to make that edge. However, when we later on go to process Gene C and find its neighbours, we want to exclude Gene B from edge creation. This is done by the line

edgeExclusions <- c(edgeExclusions, firstNeighbours[[i]][j])

in our script.

These are the cases I can think of for now and they are covered in our code.

# Clustered Layout

WE need to have some calculations that determine the ideal radii for the clustered layout option.

Venkata has requested that we give the user the option to first only look at the first neighbours of the selected nodes, then look at the second neighbours if desired. Therefore we will have to change our R script slightly so that instead of returning a list of all edges, it returns 2 lists of edges, one for the first neighbours, and one for the second neighbours. In fact, we can also change some of the other return values so that we make things more generalizable.