# Inverse Comorbidities

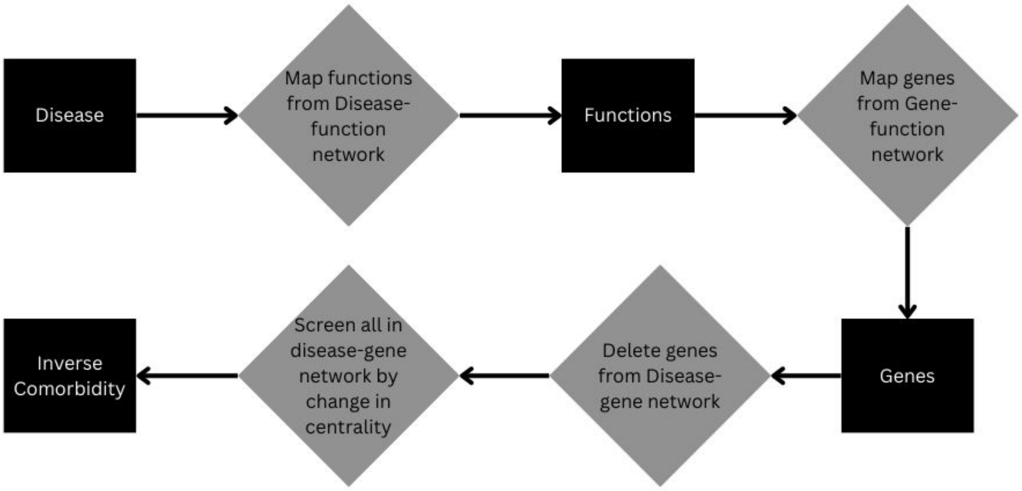
**BT5240 Course Project**

*Shreya Rajagopalan (BE21B038)*
*Aadit Mahajan (BS21B001)*

## Work done so far:

Progress done so far in the project has mainly been along the lines of understanding the intricacies that entail the problem statement on hand. On the implementation side of the project, this involves work done on procuring the right data sets, understanding the pattern of data, sorting out cross compatibility issues between data sets being used and implementing a proof-of-concept code for elementary analysis purposes. On the knowledge side of the project, we needed to learn about the different databases used for storing data regarding diseases, genes, and gene ontologies. We also had to perform elementary research on gene ontologies and disease ontologies to understand the pipeline completely.

### Logic flow and *Proof-of-concept* code


Disease
Map functions from Disease-function network
Functions
Map genes from Gene-function network
Genes
Delete genes from Disease-gene network
Screen all in disease-gene network by change in centrality
Inverse Comorbidity

We wanted to write a proof-of-concept code to confirm that the logic flow we had designed for obtaining the inverse comorbidities was working correctly. Although we did manage to come up with a strategy to find such relations, the answers we got for inputs were not very convincing. The exact example referred to is that of Alzheimer's disease. According to literature, patients with neurodegenerative disorders such as Alzheimer's or Parkinson's disease have an innate low probability of developing breast cancer. When the above logic flow was used for finding inverse comorbidities of Alzheimer's disease, the output involved a few types of cancer (indicating that the search was not completely off its target) but it was not able to identify breast cancer as one of the least probable comorbidities. Two potential areas where we could have made an error:

1. The logic flow itself
2. The final screening process for identification of inverse comorbidities.

Pertaining to the first possible problem mentioned in the previous paragraph, we realised that the above logic flow would give a list of comorbidities that only had a functional relation associating them to each other, and that this was the reason why we were not exactly able to pinpoint to the examples mentioned in literature.

Regarding the screening process we have currently implemented, identification of the nodes that can be considered as inverse comorbidities is through the difference in the initial and final degree centralities of the nodes after the deletion of genes in the disease gene network. If the change in the degree centrality of a node is higher than a threshold (which means that the node has lost a significant bunch of its neighbours), the node can be considered partially orphaned and is an inverse comorbidity. This section of the implementation has been completed. However, as mentioned before, we have been able to identify a few drawbacks and are planning to implement a slightly different approach.

## Plan for further progress

We aim to implement an alternate strategy to identifying the two categories of inverse comorbidities separately. This will involve performing clustering and community detection on the disease-gene network dataset as well as the disease-function dataset separately. Analysing the communities obtained will give a fair idea of the genes that are commonly affected by certain sets of diseases. It will also provide a good understanding for the exact screening process for identifying the correct pairs of inverse comorbidities, in terms of centrality measures / neighbourhood parameters to be considered in the screening process.