Module 3 Day 3 Report

Motivating Problem From Domain

Fanconi anemia (FA) is a rare disease that affects how DNA is translated and repaired. This leads to abnormalities in bone marrow and skeletal structure as well as an increased risk for cancer. Due to causing low red and white blood cells, patients are at an increased risk for anemia, infections, and excessive bleeding.[1] There are 22 known genes that will lead to FA if mutated. By evaluating the contribution of connections of each gene in 5000 random subnetworks, genes can be assigned a score in which a higher score correlates to a higher contribution.

Computational Problem Formation

Given 5,000 subnetworks of FA genes, assign each gene a score based on contribution to connections within the subnetwork.

Specific Approach

Given 5,000 subnetworks of FA genes, and STRING network of gene connections, score each FA gene in each loci based on its contribution to connections in each subnetwork, while all other loci remain the same.

Specific Implementation of Approach

For each loci in each 5,000 FA gene subnetworks, replace the gene from the specific loci with every other gene from that loci while keeping all other loci the same and get the densities of connections. Get the density again of the subnetwork but with the target loci removed from the subnetwork, creating an empty loci case. The difference between the density of the subnetwork with the replaced gene and the empty loci case gives the gene score. This is done for each of the 5,000 subnetworks and the average of the gene scores from each subnetwork is the genes score. Gene scores can be visualized in a network graph where color indicates loci and size is proportional to the genes score.

Pseudo Code

```
FA_all_genes = [all FA genes from input file]
FA_loci = {dictionary of each loci and its genes}
Gene_scores = {}

For gene_connection in STRING.txt:
        Sort gene connection
        Add to set to get rid of duplicates

For i, subnetwork in enumerate(FA_subnetwork.keys()):
        Genes_done = []
        Default_density = average density of subnetwork
```

```
For gene in subnetwork:
        gene_done.append(gene)
        Loci = reverse lookup which loci gene is in
        Empty_subnetwork = subnetwork with gene removed
        Get density of empty_subnetwork
        default _genes_score = unchanged_subnetwork_density - empty_subnetwork
        Add gene and score to gene_scores

        For og in other_genes_in_loci:
                If og not in genes_done:
                        Append to genes_done
                        Replace original_gene with og
                        Get density of replaced_subnetwork
                        Score = replaced_subnetwork - empty subnetwork
                        Append score and gene to gene_scores
Visualize gene_scores in network where node_size = score and color = loci
```

Discussion

Results

Each gene was scored based on its contribution to the density in each 5000 subnetwork. The visualization of the scores are shown below in figure 1 in which the different colors represent the 12 loci and the size of the nodes are representative to the gene score.

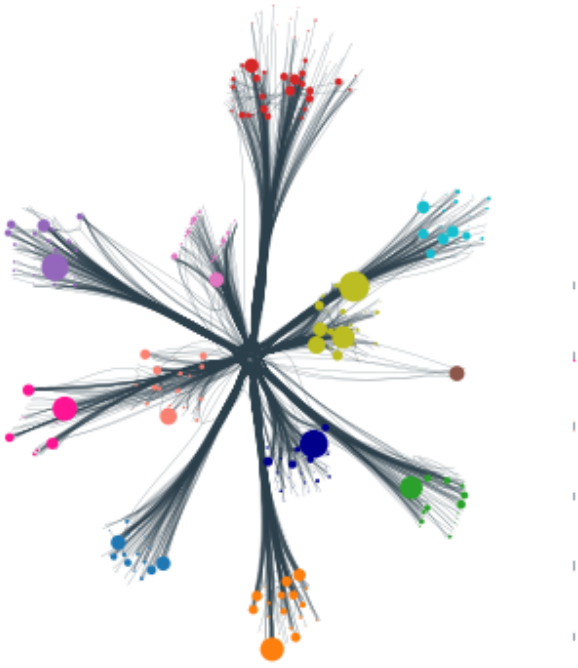

Figure 1. Visualization of FA gene scores across loci.[2]

Densities of subnetworks were based on the edge weight as opposed to just the count to further include significance of connections compared to just amount. Gene scores overall were very small, ranging from 0 to 0.02, which makes sense as the edge weights were used.

Limitations, assumptions, improvements

The main assumption made was that all FA genes should contribute to the density of at least 1 subnetwork, resulting in a score. If a gene was scored 0, it was assumed this is because that gene was not included in the STRING of all connections. However, it is also possible this is due to the gene not having any connections.

The main limitation of this code is the run time. To permute each loci for each gene, it took a few seconds per subnetwork to run and to run each 5,000 subnetwork the runtime was ~5 hours. To speed up the process, the code could instead be run in clusters on a server.

Sources

1. National Center for Advancing Translational Sciences |. rarediseasesinfonihgov. https://rarediseases.info.nih.gov/diseases/6425/fanconi-anemia.

2. Holtz, Yan. “Python Graph Gallery.” *The Python Graph Gallery*, python-graph-gallery.com/. Accessed 17 Nov. 2023.