

Lab Session 11

29.10.2024

BT3051 - DSA Biology Lab

1. Gene regulatory networks (GRNs) are networks of genes that either activate or suppress each other. They can be represented as graphs where the nodes are the genes and the edges are the regulatory relationships between the genes.

Given such a gene regulatory network, Can you use the Breadth-First Search algorithm to find the shortest path between two genes?

Write a simple Python implementation of performing a BFS on a Gene Regulatory Network that outputs the shortest path between a start-gene and a target-gene. Include a checkpoint in the code to verify if the start and target genes are present in the GRN.

Input: An unweighted directed graph.

Example -

```
gene_regulatory_network = {'gene1A': ['gene2A', 'gene1B'],
                           'gene2A': ['gene3A', 'gene2B'],
                           'gene3A': ['gene3B'],
                           'gene1B': ['gene1C', 'gene2C'],
                           'gene2B': ['gene1C', 'gene2C', 'gene3B'],
                           'gene3B': [],
                           'gene1C': ['gene2B', 'gene2C'],
                           'gene2C': [] }
```

```
start_gene = 'gene1A'
```

```
target_gene = 'gene2B'
```

Output:

Example -

The shortest path in order of start_gene to target_gene -

```
'gene1A', 'gene2A', 'gene2B'
```

2. Genetic mapping is the representation of relative distances between genes on chromosomes. Minimum Spanning Tree algorithm can be used to find the shortest path between genes which can be used to understand the organization of genetic data giving insights into mutation, migration and evolution of the genes.

Using a priority queue to track edge weights, write a Python program that implements the Prim's algorithm for determining Minimum Cost Spanning Tree.

Input: undirected weighted graph that can be represented as an adjacency list
Example -

```
gene_graph = {'geneA': [('geneB', 5), ('geneC', 3)],
              'geneB': [('geneA', 5), ('geneC', 2), ('geneD', 6)],
              'geneC': [('geneA', 3), ('geneB', 2), ('geneD', 7)],
              'geneD': [('geneB', 6), ('geneC', 7)]}
```

Output: List of edges of the MST & total cost of the edge weights in the MST

Example -

MST Edge weights = 10

MST Edges:

```
[('geneA', 'geneC', 3), ('geneC', 'geneB', 2), ('geneB', 'geneD', 5)]
```