Yisheng Zhou (Joseph)
Wayne Hayes
June 11, 2015

Bio-Net Alignment Project Report

CS 199 Final Report, Spring 2015

Object:

Using Dijkstra's Algorithm to align two graphs: yeast gene and human gene.

Material:

1. "yeast2.txt"

   This file saves edges in yeast graph. The structure of each line is "node_name1 space node_name2".

2. "human1.txt"

   This file saves edges in human graph. The structure of each line is "node_name1 space node_name2".

3. "Sims.txt"

   This file saves the similarity between one yeast node and one human node. The structure of each line is "human_node_name space yeast_node_name space similarity".

Background:

Coverage calculation: n1 and n2 are in yeast while n3 and n4 are in human; n1 aligns to n3 while n2 aligns to n4; there is an edge between n1 and n2, and an edge between n3 and n4. Then, n1-n2 and n3-n4 can be recorded as well-covered edges. Our final result is the percentage of the number of the well-covered edges in the number of all edges in the given graph .

Introduction:

Dijkstra's original variant found the shortest path between two nodes. It regards a single node as the "source" node and finds shortest paths from the source to all other nodes in the graph, producing a shortest path tree. For this material, we need to produce two shortest path trees (may be several trees, if the graphs are not connected graph), one in human graph and one

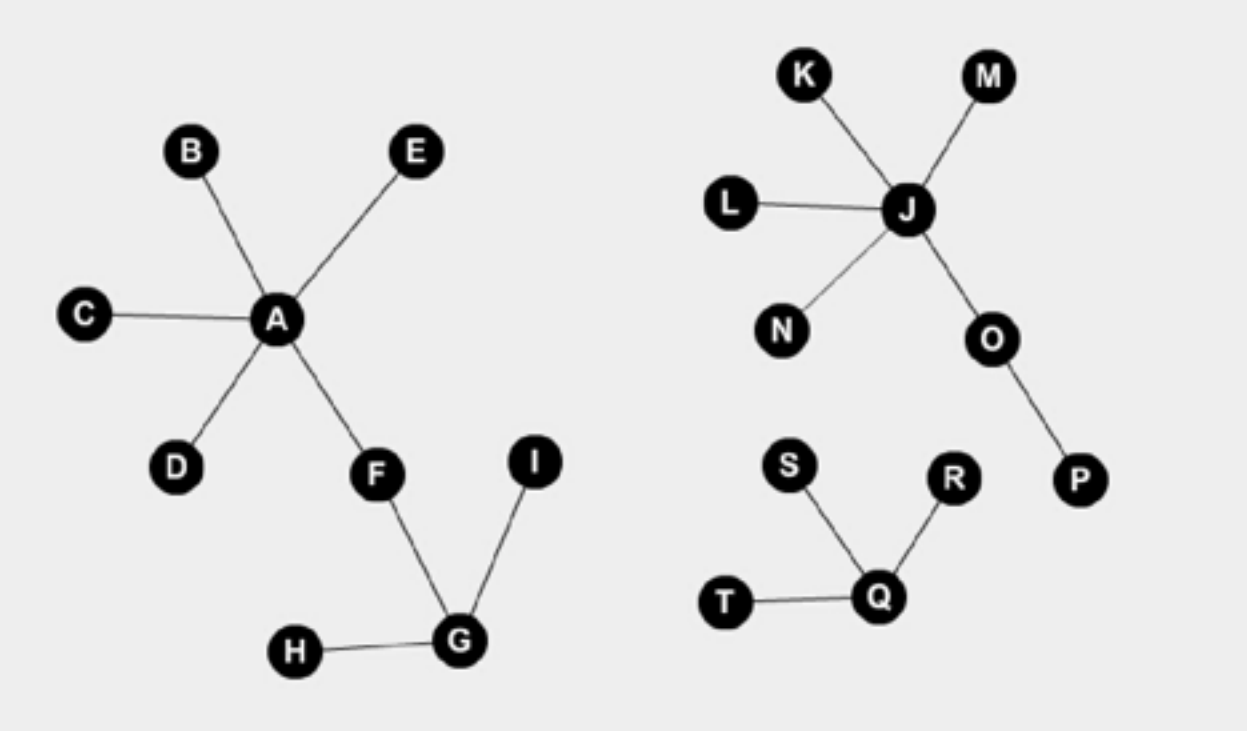
in yeast graph. The nodes in two trees are aligned. The shortest paths in this project are the paths that can reach the pair of nodes with the highest similarities.

Part I: Dual Graph Idea

The result of the program is based on edge coverage, while the expansion of the graph is based on nodes. If the program regards edges as nodes, every expansion can directly increase the number of the aligned edges, which means the edge coverage increases. There is already an algorithm for doubly connected edge list (DCEL) that can translate faces into nodes and edges between faces into edges between nodes, which is called Dual Graph Algorithm. The program can also apply the idea to a regular graph.

However, Dijkstra algorithm is a greedy algorithm. The new expansion in the program requires similarity values. If the program simply translates old graph into new graph, it requires a new formula to define nodes' value in the new graph based on the old graph values of nodes the edge connects. Now, the program does not have a reliable formula, which makes the dual graph transfer not work.

Part II: Break Point Idea



Graph A to G is a connected component in human. Graph J to P and Graph Q to S are two connected components in yeast. A and J have the highest similarity. Let [A, B, C, D, E] align to [J, K, L, M, N]. Then, O will align to F and P will align to G if the expansion keeps working. However, from this graph, we can see if we let [Q, R, S, T] in another connected component align to [G, I, F, H], the program can get a higher coverage.

Therefore, if the similarity is low enough, the program should find another pair of nodes with the highest similarity and start a new expansion. This assumption will be achieved in the following version because the programmers do not have a clear threshold of low similarity.

Part III: New Definition of Coverage in Human Graph

The previous definition of edge coverage is number_of_aligned_edges/total_yeast_edges and number_of_aligned_edges/total_human_edges. However, it may not make sense because total_yeast_edges and total_human_edges are constant numbers. Using total aligned edges to divide these two values can only give two results with a new constant ratio. Nothing new can be shown. Because not all nodes in human_graph are aligned, the program can build new subgraph of human_graph that only contains aligned nodes and their edges. If using the edge number of the subgraph, the denominator will be influenced by the alignment, which can show how our alignment works from some other points of view.

Part IV: Timestamp as Filename

The program applies random seeds, which means the program gives different result every time. In order to prevent new file cover old file, the program needs to check whether old file exists. If old file is already in the folder, the program needs to find a new name. File check in Python is slow.

There is a small improvement: use timestamp as file name. So, every time the program run, it will make a new file. All files can also be sorted in order.

Conclusion:

After trying 5 times, the edges coverage values (number_of_aligned_edges/ total_yeast_edges) gained from Dijkstra algorithm is between 10.2% to 11.5%.