Adding Color-Coding algorithm as an extension to Cytoscape

G Sanjay(CS19B019), Sriram Goutam P(CS19B073)

1 Introduction

1.1 Motivation

Without knowledge of what is going on outside of it, a cell cannot carry out its fundamental task or even choose whether to survive, develop, or die! Signaling pathways are a complex network of molecular interactions that allow cells to communicate with one another. Through these pathways, cells get information from their surroundings as well as from other cells, enabling them to react properly to changes in their outside environment. Signaling pathways are critical for cell and organism function because they regulate a wide range of biological functions.

A signalling pathway is a series of chemical reactions and molecular interactions that take place within a cell in response to a particular signal. A ligand, such as a hormone or neurotransmitter, binds to a particular receptor protein on the cell surface or within the cell to activate the route. This binding event sets off a chain of events that result in a particular reaction within the cell.

An example of Canonical Signaling Pathway

Some uses of signaling pathways in cells:

- To sense the location of harmful microbes by our immune system, so that it can go and kill it.
- To sense when to release energy vs when to conserve energy by our fat cells. Molecules like Insulin are used for this purpose.
- Whether a cell should grow or die depends upon signaling molecules. If the outside environment is conducive for the growth of cells then it will grow.

Protein-protein interactions research is critical for determining the pathways a cell uses for sensing the outer environment. One such approach for locating signalling pathways in a protein-protein interaction network is **color-coding**.

Cytoscape is a prominent open-source software platform for visualising and researching biological networks. There are currently just a few tools available in Cytoscape for researching protein-protein interactions. We propose developing a Cytoscape plugin for color-coding algorithm that would allow users to study complicated protein interaction networks and identify relevant connections and pathways.

1.2 Paper critique

We used the papers Efficient Algorithms for Detecting Signaling Pathways in Protein Interaction Networks and Algorithm Engineering for Color-Coding with Applications to Signaling Pathway Detection as our reference.

This first paper provides a method for finding signaling pathways in protein-protein interaction networks. This is done in two stages: first values are assigned to interactions of proteins based on some features (like how many times were the proteins experimentally observed to interact), and hence a graph is made out of protein interactions(with the values indicating the probability that the two proteins interact). Next the first paper finds the longest simple path in the graph of some specific length(say k), this is done after taking the logarithm of all the edges in the graph since in the original graph the product of the edges is to be maximized.

Two algorithms were proposed by the authors for finding the longest simple path: the first one is an exact algorithm based on Dynamic Programming, this is a brute force method and inefficient. The time complexity of this algorithms is $O(kn^k)$ and the space complexity of the algorithm $O(n^k)$. Here n denotes the number of vertices.

The second algorithm is a randomized algorithm(which also uses Dynamic Programming) known as the *color coding* algorithm. This was proposed by Alon *et al* in 1995. This algorithm assigns colors to each vertex and instead of finding a simple path, a colorful path is found(colorful path means where each vertex in the path has a different color). This algorithm doesn't guarantee to give the longest path, hence the algorithm is repeated many times. The time complexity of this algorithm is $O(2^k km)$ and the space complexity is $O(2^k n)$. Here m denotes the number of edges.

One of the important contributions of the paper is that it extends the color coding algorithm in many ways:

- Suppose we want to make sure that a specific protein occurs in a path then we can simply assign a color *only* to that protein.
- The algorithm is also extended for finding other data structures from simple paths. These structures occur naturally while analyzing real life biological situations. An example is *rooted trees*

The paper then applies the color coding algorithm to the *Yeast Protein Network* and compares it with the brute force method. For length 7 paths the brute force method performs better by 1.5 times, for paths of length 8 and 9 color coding performs better by 2 times and 7 times respectively.

To conclude, the paper uses color coding in an excellent manner to improve the results for finding signaling pathways.

1.3 Problem statement

Biological problem: Finding signaling pathways by which a cell senses its environment efficiently.

Computational problem: A protein-protein interaction network in Cytoscape is given where edges represnt the logarithm of the probability that the two proteins interact. Then implement a plugin for the color coding algorithm which outputs the maximum weight path, the value of the corresponding path and the time taken by the algorithm to calculate the path.

2 Methodology

There are mainly two parts in the project: Implementation of Color Coding and making an interface on Cytoscape for uploading input and printing output. The whole app is written in Java.

2.1 Implementation of Color Coding Algorithm

Setting: Let V be the set of vertices of the graph, I be the set of starting vertices, T be the set of ending vertices. We are to find maximum weight paths of length k.

Brute force implementation: Let $W(v, S)$ be the weight of the best path ending at v involving all the vertices of set S. Here $|S| = k$. The Dynamic Programming relation is as follows:

$$W(v, S) = \max_{u \in S \setminus \{v\}} W(u, S \setminus \{v\}) + W(u, v), |S| > 1$$

$W(v, \{v\}) = 0$ if $v \in I$ otherwise $W(v, \{v\}) = \infty$

Applying this relation in the bottom up fashion we can calculate maximum weight path of length k ending at v.

For a vertex v the number of sets containing v of the size k is n^{k-1}. There are $O(n)$ number of destination vertices, hence the space taken by the algorithm is $O(n^k)$.The total number of paths of length k is n^k(n choices for each vertex), for each path it will take $O(k)$ time to find the value of the path, hence it will take $O(kn^k)$ time to find the maximum value path.

Since n is usually very large in biological systems this method takes a lot of time and is not suitable for practical purposes. Hence the need for color-coding algorithm.

Color Coding Implementation: We have k colors and we assign each vertex a color at random. Then we find the maximum weight colorful path(i.e. path with distinct colors).

We repeat this process multiple times because we may not get the maximum weight simple path the first time. As this is a randomized algorithm, a path of length k may not be colorful in the given coloring and because of this it may not be identified.

The probability that a k length simple path is colorful is $p = \frac{k!}{k^k}$, since there are $k!$ possible permutations of k colors and k choices of color for each vertex. Let t be the number of trials required to ensure that the probability of failure is at max ϵ. Then

$$1 - (1 - p)^t \geq 1 - \epsilon$$

$$\implies t \geq \frac{\ln\epsilon}{\ln(1 - p)}$$

$$t \geq \frac{\ln\epsilon}{-O(e^{-k})}$$

since $\frac{k!}{k^k} \geq \sqrt{2\pi k}e^{-k}$

$$t \geq e^k \ln(\frac{1}{\epsilon})$$

In our algorithm we have taken 99% as the minimum success rate(i.e. $\epsilon = 0.01$ is used to calculate t).

The Dynamic Programming relation of color coding is similar to that of the brute force approach except in the place of set of vertices, we use set of colors. Hence now instead of number of vertices for space we count number of subsets of the set of colors which is 2^k instead of n^{k-1}. Hence the space complexity will be $O(2^k n)$ instead of $O(n^k)$ and time complexity will be $O(2^k km)$.

The recursive relation is of the color coding algorithm is:

$$W(v, S) = \min_{u:c(u)\in(S\setminus\{c(v)\})} W(u, S\setminus\{c(v)\}) + w(u, v), |S| > 1$$

where $W(v, \{c(v)\}) = 0$ if $v \in I$ otherwise $W(v, \{c(v)\}) = \infty$.

Subsets of colors are represented using bitmaps. Let $a_k a_{k-1} \dots a_2 a_1$ be the bitmap of the set S. Let $S = \{1, 2\}$ then it will be represented as $1100 \dots 00$, if $S = \{1, 3\}$ then it will be represented as $101 \dots 0$. These bitmaps are inturn represented as numbers, for example $a_k a_{k-1} \dots a_2 a_1$ is represented as $\sum_{l=1}^{k} a_l 2^l$. For each bitmap the number is unique. Hence the set S is represented as a number $N(S) = \sum_{l=1}^{k} a_l 2^l$.

Let $\alpha(v)$ be the number corresponding to the vertex v. Then $W(v, S)$ is represented as $\text{arr}[\alpha(v)][N(S)]$ in the code. Remaining part of the algorithm is just executing the DP relation from bottom up.

2.2 Adding Color Coding implementation as an extension to Cytoscape

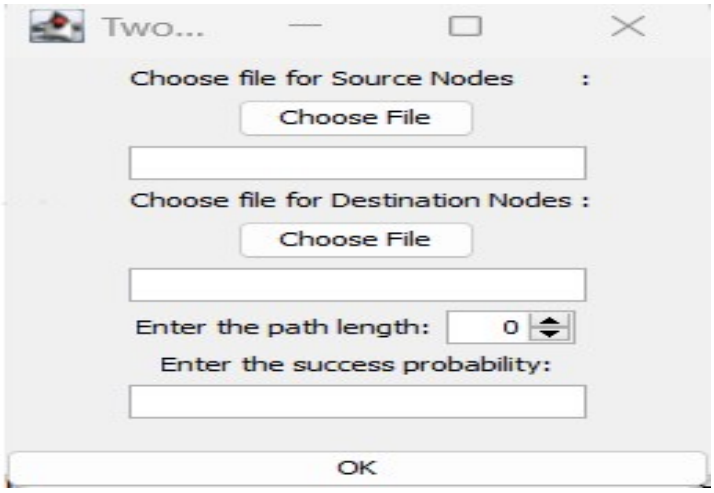
Cytoscape is an open-source software platform that is widely used for visualising, analysing, and modelling large networks. It has an easy-to-use interface and a variety of sophisticated capabilities for network research and visualisation.

A Tutorial provided by Cytoscape for new plugin development is used for this extension development. Java is the language used for Cytoscape development. Three steps for adding color coding as Cytoscape plugin

- Taking Protein-Protein interaction network as input from user.
  - This is handled by Cytoscape itself.
- Taking source nodes, destination nodes, path length k and success probability from user.
  - This is handled by creating a pop-up for input and prompting user.
- Running the color coding on the uploaded graph and input. Finally generating the output.

2.2.1 Creating a pop-up for taking input from user

- Once the protein-protein interaction graph is uploaded to cytoscape, user can be able to find the cytoscape plugin for Color Coding in Apps/Samples. Once the plugin is clicked, following popup is rendered on screen.
- The above pop is created using Java features like $JFrames$, $Jpanels$, $JLabels$, $JButton$ etc.
- Once the OK button is clicked. Color Coding in run in the backgrounfd for the given uploaded input and output is rendered.



2.2.2 Running the Color coding and Rendering the output

- The input given by user by uploading is read and is converted into format suitable for implementation of Color Coding.
- Adjacency list, Adjacency matrix are build from input graph using

  $$cyApplicationManager.getCurrentNetwork()$$

  and lists for source nodes and destination nodes are build by reading the input files.
- This adjacency list, adjacency matrix and all other input are passed to implementation of Color Coding.
- The final output from Color Coding is rendered to user.

3 Results obtained

- Sample output is as follows

- Source nodes are colored Green and Destination nodes are colored Red in the graph.

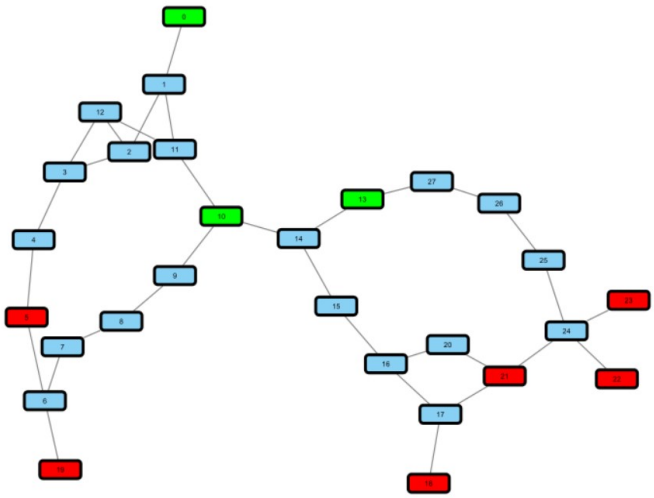


Figure 1: Rendering graph in Cytoscape

We tested our app against a graph of $n = 2407$ and $m = 6511$ (this is only part of the dataset as Cytoscape wasn't loading the complete dataset) and for path lengths 4, 5, 6, 7 and 8. The results are as follows:

Path length	Time taken(ms)
4	1119.47
5	5589.49
6	46402.59
7	101199.81
8	502951.27

Table 1: Times from our implementation.

The paper that we cited also tested its algorithm on the Yeast Protein Network. The number of nodes in the graph is around 4500 and number of edges is 14319. The results in the paper are as follows:

Path length	Time taken(s)
6	32
7	97
8	498

Table 2: Times from paper.

4 Conclusions and Future work

4.1 Conclusions

We have made a Cytoscape app using Java which provides maximum weight paths of specific length using Color Coding algorithm. The time taken by our algorithm for large graphs was comparable to the results given in the paper *Efficient Algorithms for Detecting Signaling Pathways in Protein Interaction Networks.*

Cytoscape is able to take graphs as large as 2500 nodes and 6500 edges, but it is not able to handle very large graphs, so testing was not possible with very large graphs.

4.2 Future works

- A good idea for a future work would be to write the algorithm by extending the number of colors to $1.3k$ and check the results and verify them with the results given in the paper *Algorithm Engineering for Color-Coding with Applications to Signaling Pathway Detection.*
- Another great idea would be to extend the algorithm to more general data structures like rooted trees and Two-terminal series-parallel graphs.
- Also we would like to propose the app to Cytoscape developers to integrate into the environment.