Paper Title

A. B. AUTHOR* and C. D. AUTHOR

University Department, University Name,
City, State ZIP/Zone, Country
**E-mail: ab_author@university.com*
www.university_name.edu

A. N. AUTHOR

Group, Laboratory, Street,
City, State ZIP/Zone, Country
E-mail: an_author@laboratory.com

Here should come the abstract.

Keywords: keyword 1; keyword 2; keyword 3;

1. Introduction

Studying interactions between proteins has been of utmost importance in understanding how proteins function collectively to govern cellular machinery.[1,2] Such collection of interactions is modeled as a protein interaction network. Mathematically, a protein interaction network is often represented as an edge-weighted graph where each node denotes a protein and each edge represents an interaction between a pair of proteins. The weight of an edge denotes the level of confidence that this interaction truly exists.

One of the key outcomes of computational analysis of protein interaction networks is identification of signaling pathways. A signaling pathway is a series of proteins in which each protein participates in transmitting biological information by modifying its successor through an interaction. Thus, signaling pathways can be viewed as simple paths in protein interaction networks.[3]

The confidence value of an interaction between two proteins is often considered as the probability that a signal is transmitted between those two proteins. Thus, the probability that a signal moves through a pathway is the product of the confidence values of its constituting interactions. Under this model, Scott et al. conjectured that a signal tends to move through the most probable pathway.[4] They showed that such pathways yield signaling pathways, and thus help in reconstructing signaling networks. Following defines the problem of identifying most probable pathways in a protein interaction network.

Problem Consider a protein interaction network (V, E, p) where V denotes the set of proteins, $E = \{(u, v)\} | u, v \in V$ denotes the set of interactions, and the function $p() : E \Rightarrow [0, 1]$ denotes interaction confidence for each interaction in E. Assume that we are given a set of starting proteins $S \subseteq V$ and a set of target proteins $T \subseteq V$. Given a path length denoted by a positive integer m, the problem is to find a simple path $P = v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_m$, where $\prod_{i=1}^{m-1} p(v_i, v_{i+1})$ is maximum among all paths where $v_1 \in S$ and $v_m \in T$.

The problem above is equivalent to finding a simple path $P = v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_m$, where $\sum_{i=1}^{m-1} -\log\ p(v_i, v_{i+1})$ is minimum among all paths where $v_1 \in S$ and $v_m \in T$. Scott et al.[4] mentioned that the traveling-salesman problem is polynomial-time reducible to this problem; therefore it is NP-hard. They developed a method using a technique devised by Alon et al.,[5] called *color-coding.* The basic idea of this method is to randomly assign each node in the graph one of m different colors, and search for an optimal pathway in the restricted domain of colorful pathways. A pathway is colorful if and only if all of its nodes are in different color. Finding a colorful path is computationally much cheaper than finding a path without assigning colors. The drawback is that the optimal path may not be colorful in a random color assignment. In this case, the color coding method fails to find the true result. To deal with this, color coding method repeats this process for several iterations until it reaches a given level of confidence that the unknown optimal pathway was among the colorful ones in at least one of these iterations. The confidence in the optimality of the result monotonically increases with each iteration. This confidence value depends solely on the pathway length m and does not capitalize on readily available information such as the network topology and color assignment. As a result, the method provides a theoretically correct but very conservative confidence value. Hence it requires many iterations in order to achieve a given confidence level.

Gülsoy et al.[6] presented an enhanced color-coding technique called *k-hop coloring.* A colored network is k-hop colorable if the shortest path between any pair of same-color nodes is more than k hops in length. This method exploits the network topology and the node colors to assign the network a maximal value k such that the network is k-hop colorable. This additional piece of information allows for higher success probability at each iteration, yielding fewer iterations than that by Scott et al. However, subnetworks with high connectivity quickly diminishes the ability to k-hop color the whole network for large values of k. For example, a network containing a clique of size m cannot be colored with $(m-1)$-hop coloring using m colors.[6]

Contribution Our motivation comes from the need for a deeper understanding of the relation between network topology, random color assignment and success probability. We study the possibility of assigning k values to nodes on an individual basis instead of a single k value for the whole network. We also study how this reflects on the resulting success probability for each iteration. We examine the idea that a pathway whose nodes are assigned different k values should result in a higher success probability than if we only consider the minimum of these k values for all nodes. Given different k values for each node on a pathway, we show how to obtain a better bound on success probability.

Based on these findings, We present a new method for detecting signaling pathways in protein interaction networks using an enhanced k-hop coloring technique. For a required optimal pathway of length m, we start by assigning one of m colors to each node in the graph, then we extract the optimal colorful pathway. We then calculate our new bound on success probability. We repeat this process until the cumulative success probability is at least equal to a given confidence level. Although our theoretical findings are based on assuming the knowledge of the k values assigned to the unknown global optimal pathway, we empirically show that the local optimal pathway extracted from the domain of colorful pathways actually fits the purpose.

We provide validation experiments to test the biological significance of our results. We use *weight p-value* and *functional enrichment* as validation measures. We also compare the performance of our method against the one presented by Scott et al.[4] with respect to how fast our method reaches a given confidence level as opposed to theirs.

The rest of the paper is organized as follows. Section 2 discusses the background and related work. Section 3 explains how to obtain a tighter bound on success probability and describes our enhanced k-hop coloring method. Section 4 shows the experiments performed and their results. Section 5 is the conclusion of the paper.

2. Background

Some different but closely related problems have been studied in the literature. Zhao et al.[7] used integer linear programming to find signaling networks in protein interaction networks. They formulated the problem as a linear optimization problem of finding maximum weight subnetwork with a given size. This approach is concerned with finding signaling subnetworks in their general form rather than linear pathways. Kelley et al.[3] detected conserved signaling pathways between related organisms by performing global alignment between their protein interaction networks. They scored each pathway in terms of probability of true homology between aligned pair of proteins, as well as probability of true interactions between pairs of proteins along the pathway. This approach detects conserved pathways only and requires coupling between two datasets. Shlomi et al.[8] introduced QPath, a method for querying protein interaction networks for pathways using known homologous pathways as queries. They scored results based on their similarity to the query, number of insertions and deletions used, as well as the reliability of their interactions. This method only detects pathways that are similar to a given one.

On the other hand, the problem we address has also been studied in the literature. Lu et al.[9] presented a divide-and-conquer algorithm for finding generic pathway structures in protein interaction networks. They recursively partitioned the network into two sets of vertices, enumerated substructures present in each set, and then built larger structures from them. They assumed that all edges have the same weight and scored the resulting pathway structures based on the biological function relatedness of their nodes to the given source and destination nodes. We are more interested in detecting pathways based on confidence in interactions rather than similarity of proteins.

Steffen et al.[10] studied detecting signaling pathways in protein interaction networks as guided by expression data. They listed all pathway candidates in a protein interaction network using exhaustive search. They scored each candidate based on how similar the expression profiles of its genes are. Bebek et al.[11] presented a method called PathFinder for finding new signaling pathways using association rules of known ones. They started with mining association rules for known pathways, guided by the knowledge of functional annotations of their proteins. They then performed an exhaustive search for candidate pathways. From these candidates, they selected the ones having at least a certain number of the known association rules and an average interaction weight above a given threshold. The drawback of both of these methods is that the time complexity of exhaustive graph search is exponential in terms of the network size, and hence is very inefficient.

Gitter et al.[12] presented a method for discovering signaling pathways by adding edge orientation to protein interaction networks. They selected an optimal orientation of all edges in the network that maximizes the weights of all satisfied length-bound paths. A path is satisfied if it follows the same direction along its edges from a source node to a destination node. They proved that this problem is NP-hard. They provided two approximation algorithms for it based on available solution methods for weighted Boolean satisfiability, and a third algorithm based on probabilistic selection. As shown in their results, these methods do not scale well with increasing the number of source and destination nodes and the required path length.

The present work builds on the method presented by Scott et al.[4] for detecting signaling pathways in protein interaction networks using color coding. It also enhances the model presented by Gülsoy et al.[6] for topology-aware color coding. We indroduced both methods in section 1.

3. Methods

In this section, we start by properly formulating the problem and defining common terms that we use in our methods. We then present new thoughts about pathway detection using color coding. We study the opportunity of more involving of network topology in our calculation to obtain a better success probability, and hence needing less number of iterations and improving performance. Last, we present an enhanced color-coding method for detecting pathways in protein interaction networks.

3.1. *Problem Formulation and Term Definition*

Given a graph $G = (V, E, w)$, where V is its set of nodes, E is its set of edges and w is the edge-weight function, and given a set $S \subset V$ and a path length m. Assume $P(i)$ is the set of all simple paths of length m starting at any node $s \in S$ and ending at node i. Our goal is to find, for each node i, the path $p \in P(i)$ whose sum of edge weights is minimum.

The following are the definition of some commonly used terms:

(1) K NEIGHBORHOOD: for a given node $v \in V$ and an integer k, the k neighborhood of v is a set of nodes $U \subset V$ where a node $u \in U$ if and only if u can be reached from v in k hops or less.
(2) MAX_K: for a given node $v \in V$, $max\_k(v)$ is the maximal value of k such that $\forall u \in k$ neighborhood of v, the color assigned to u is not equal to the color assigned to v.
(3) MAX_K CONFIGURATION: for a given path P, the max_k configuration of P is the sequence of max_k values corresponding to the sequence of nodes in P.

3.2. *Success Probability: a Tighter Bound*

Based on the work presented by Scott et al.,[4] a generic color-coding approach to solving the problem consists of three main steps. The first step is coloring the network; $\forall v \in V$ we independently select a color drawn uniformly at random from a set of m different colors. The second step is finding an optimal colorful path; $\forall v \in V$ we want to find the minimum-weight colorful path of length m starting in S and ending at v, and then we extract the minimum of these paths. The third step is calculating the success probability P_s; we calculate a lower bound on the probability that the unknown overall optimal path is indeed colorful, hence the probability that it is indeed the optimal colorful one we found. These three steps are repeated r times until $1 - \prod_{i=1}^{r}[1 - P_s(i)] \geq \epsilon$, where ϵ is a required confidence level. It is obvious that a higher success probability would result in less number of iterations required, hence less execution time.

Calculating success probability in general is a counting problem. The generic rule is:

$$P_s = \frac{m!}{N_c} \tag{1}$$

where $m!$ is the number of coloring possibilities in which the path is colorful, and N_c denotes the total number of coloring possibilities for the path. Scott et al.[4] calculated N_c as equal to m^m. This calculation considered no restrictions on the color selection of each node, and discarded available knowledge of the network topology and colors already assigned to its nodes.

As guided by Gülsoy et al.,[6] knowing the network topology can be useful in calculating success probability, specifically the number of coloring possibilities of the optimal path. $\forall v \in V$, $\forall u \in k$ neighborhood of v, if the color of u is not equal to that of v, then the number of coloring possibilities can be calculated as follows:

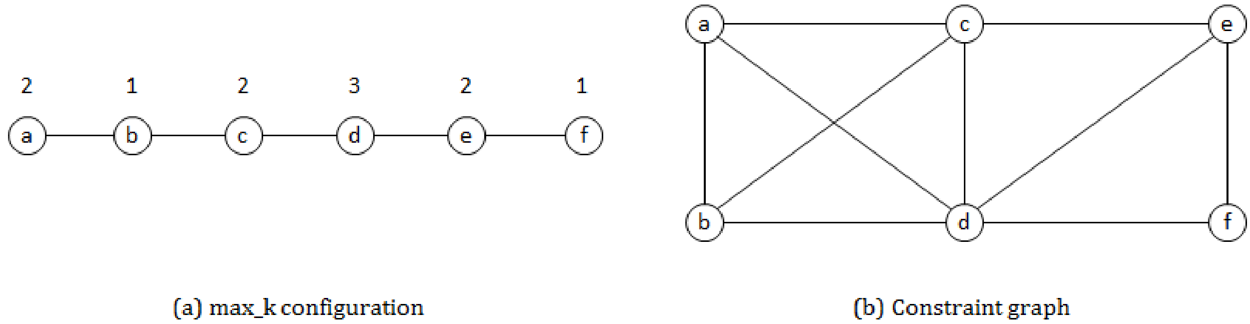
$$N_c \leq (m-k)^{m-k} \prod_{i=0}^{k-1}(m-i) \leq m^m \tag{2}$$

which, according to equation (1), results in a lower bound on success probability that is higher in value if $k > 0$. However, according to this scheme, the node with a minimum value of *max_k* dominates the whole network, which produces a correct but very conservative lower bound on success probability.

Our approach relies on individual *max_k* values of all nodes in an optimal path. Assuming knowledge of the *max_k* configuration of the optimal path, we use it to calculate N_c under the restrictions induced by these values. We also assume that each node in the path is not connected to any other nodes except the ones before and after it in the path. This assumption is valid because any more connections will only induce more coloring restrictions, causing N_c to decrease; therefore we get a solid upper bound on N_c, hence a solid lower bound on P_s according to equation (1). For a given node v in a given path, all *max_k*(v) nodes in either direction from v are not allowed to have the same color as v. We represent this rule as an unweighted constraint graph $W = (H, L)$ where H is its set of nodes and L is its set of edges. H contains a node corresponding to each node in the path, and L contains an edge for each pair of nodes that are not allowed to have the same color, according to the aforementioned rule. Figure 1 shows an example of a path, its *max_k* configuration and the corresponding constraint graph W. The problem now translates to calculating the value of the chromatic polynomial $P(W, m)$: the number of ways of coloring W using m colors without any pair of adjacent nodes having the same color. We calculate this value using the following edge-contraction recursive rule based on the fundamental reduction theorem:[13]

$$P(W, m) = P(W - uv, m) - P(W/uv, m) \tag{3}$$

where u and v are any pair of adjacent nodes, $W - uv$ is the graph W after removing the edge uv, and W/uv is the graph W after merging the nodes u and v.



(a) max_k configuration

(b) Constraint graph

Fig. 1. (a) An example 6-node path with its *max_k* configuration shown above it. Each *max_k* value translates to the number of nodes that have to be of different color on either direction. (b) The corresponding constraint graph W: each pair of adjacent nodes have to be of different color. Finding the value of the chromatic polynomial $P(W, m)$ yields the number of coloring possibilities for the path under the given constraints.

According to this method, the value of N_c for the example path shown in Figure 1(a) is 5,760, while Scott et al.[4] and Gülsoy et al.[6] would respectively yield $N_c = 46{,}656$ and 18,750 for the same example. Such a decrease in the value of N_c leads to an increase in the value of P_s according to equation (1).

3.3. *Method: Enhanced k-hop Coloring*

The approach introduced in the previous section for calculating success probability assumes the knowledge of the *max_k* configuration of the optimal path. Needless to say, this is not the case. We present a conjecture that we can instead use the *max_k* configuration of the local colorful optimal path. We empirically show that this substitution serves the purpose. Our method reports the optimal colorful path in each iteration and computes P_s based on its *max_k* configuration. We also keep a heap of the top 100 reported paths to cover the possibility of a pathway having a suboptimal score. The method is detailed as follows:

(1) Initializations:
   (i) $M \Leftarrow \{1, 2, ..., m\}$: the set of all m colors.
   (ii) $P \Leftarrow 0$: overall success probability.
   (iii) $H \Leftarrow \{\}$: heap of top 100 paths.

(2) $\forall v \in V$, $c(v) \Leftarrow$ a color uniformly drawn from M.

(3) $\forall v \in V$, the minimum weight of a colorful path colored only using $M' \subseteq M$, starting within S and ending at v, can be dynamically tabulated using the following recurrence:[4]

$$W(v, M') = \min_{u:c(u)\in(M'\backslash\{c(v)\})} W(u, M'\backslash\{c(v)\}) + w(u, v), |M'| > 1 \tag{4}$$

where $W(v, \{c(v)\}) = 0$ if $v \in S$ and ∞ otherwise.

(4) Report path X whose weight $= \min_{v \in V} W(v, M)$.
(5) Add X to H.
(6) Compute N_c using the *max_k* configuration of X according to the chromatic polynomial recurrence detailed in equation (3).
(7) Compute $P_s \Leftarrow m!/N_c$.
(8) Update $P \Leftarrow 1 - (1 - P)(1 - P_s)$.
(9) Repeat from step (2) until $P \geq \epsilon$.

4. Experiments

4.1. *Datasets*

Here we list the datasets used in experiments. I think we can use the MINT datasets for multiple organisms.

4.2. *Validation Experiments*

Here we list the validation experiments we did and their results. I think we should do some validation experiments similar to those done in Sharan's paper, using weight p-value and functional enrichment as validitiy measures.

4.2.1. *Validation using Weight p-value*

For each dataset we use, we should obtain the 99 percent confidence optimal pathway and compare it with optimal pathways obtained we obtain from random networks. We generate random networks by shuffling edges of the original network. The weight p-value is the percentage of cases where the algorithm produces a more optimal pathway when run on one of these random networks.

4.2.2. *Validation using Functional Enrichment*

For each dataset, we we obtain the 99 percent confidence optimal pathway and test its functional enrichment. For each GO term appearing on the dataset proteins, we count the total number of proteins annotated by it and the number of proteins in the resulting pathway annotated by it. We use these numbers, along with the total number of proteins and the number of proteins in the pathway, as parameters for a hypergeometric test (I still have to develop further understanding about the details of this test). The maximum enrichment value for any of the tested GO terms gives us the final functional enrichment p-value.

4.3. *Comparison with Sharan*

This is just a temporary title for this subsection, I'm not very sure what to name it.

We measure the time and number of iterations needed by our method to obtain 70%, 90% and 99% confidence pathways of lengths 6, 7, 8 and 9 nodes. We compare these numbers against the ones by Sharan's method for the same cases.

We run our method for 500 iterations and measure the incremental success probabilty against iteration number. We do this experiment many times take the average curve. We do the same experiment using Sharan's method and obtain a second curve. We also measure the average practical success probability, which is the observed probability that the DP algorithm finds the optimal solution in a certain iteration or before it. We compare the three curves targetting two conclusions: (1) our method is experimentally solid because our calculated success probabiilities are lower than the observed ones; and (2) our method outperforms Sharan's method.

5. Conclusion

Here goes the conclusion.

References

1. B. Schwikowski, P. Uetz and S. Fields, *Nature Biotechnology* **18**, 1257 (December 2000).
2. P. Uetz, L. Giot, G. Cagney, T. A. Mansfield, R. S. Judson, J. R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, A. Qureshi-Emili, Y. Li, B. Godwin, D. Conover, T. Kalbfleisch, G. Vijayadamodar, M. Yang, M. Johnston, S. Fields and J. M. Rothberg, *Nature* **403**, 623 (February 2000).
3. B. P. Kelley, R. Sharan, R. M. Karp, T. Sittler, D. E. Root, B. R. Stockwell and T. Ideker, *Proceedings of the National Academy of Sciences* **100**, 11394 (September 2003).
4. J. Scott, T. Ideker, R. M. Karp and R. Sharan, Efficient algorithms for detecting signaling pathways in protein interaction networks, in *Proceedings of the 9th Annual international conference on Research in Computational Molecular Biology*, RECOMB'05 (Springer-Verlag, Berlin, Heidelberg, 2005).
5. N. Alon, R. Yuster and U. Zwick, *J. ACM* , 844 (1995).
6. G. Gülsoy, B. Gandhi and T. Kahveci, Topology aware coloring of gene regulatory networks, in *Proceedings of the 2nd ACM Conference on Bioinformatics, Computational Biology and Biomedicine*, BCB '11 (ACM, New York, NY, USA, 2011).
7. X.-M. Zhao, R.-S. Wang, L. Chen and K. Aihara, *Nucleic Acids Research* **36**, p. e48 (2008).
8. T. Shlomi, D. Segal, E. Ruppin and R. Sharan, *BMC Bioinformatics* **7**, p. 199 (2006).
9. S. Lu, F. Zhang, J. Chen and S.-H. Sze, *Algorithmica* **48**, 363 (August 2007).
10. M. Steffen, A. Petti, J. Aach, P. D'haeseleer and G. Church, *BMC Bioinformatics* **3**, p. 34 (2002).
11. G. Bebek and J. Yang, *BMC Bioinformatics* **8**, p. 335 (2007).
12. A. Gitter, J. Klein-Seetharaman, A. Gupta and Z. Bar-Joseph, *Nucleic Acids Research* **39**, p. e22 (2011).
13. F. Dong, K. Koh and K. Teo, *Chromatic Polynomials And Chromaticity of Graphs* (World Scientific Pub., 2005).