JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 13, Number 2, 2006
© Mary Ann Liebert, Inc.
Pp. 133–144

Efficient Algorithms for Detecting Signaling Pathways in Protein Interaction Networks[1]

JACOB SCOTT,[2] TREY IDEKER,[3] RICHARD M. KARP,[4] and RODED SHARAN[5]

ABSTRACT

The interpretation of large-scale protein network data depends on our ability to identify significant substructures in the data, a computationally intensive task. Here we adapt and extend efficient techniques for finding paths and trees in graphs to the problem of identifying pathways in protein interaction networks. We present linear-time algorithms for finding paths and trees in networks under several biologically motivated constraints. We apply our methodology to search for protein pathways in the yeast protein–protein interaction network. We demonstrate that our algorithm is capable of reconstructing known signaling pathways and identifying functionally enriched paths and trees in an unsupervised manner. The algorithm is very efficient, computing optimal paths of length 8 within minutes and paths of length 10 in about three hours.

Key words: protein interaction network, signaling pathway, color coding, randomized algorithms.

1. INTRODUCTION

A MAJOR CHALLENGE OF POSTGENOMIC BIOLOGY is to understand the complex networks of interacting genes, proteins, and small molecules that give rise to biological form and function. Protein–protein interactions are crucial to the assembly of protein machinery and the formation of protein signaling cascades. Hence, the dissection of protein interaction networks has great potential to improve the understanding of cellular machinery and to assist in deciphering protein function.

The available knowledge about protein interactions in a single species can be represented as a *protein interaction graph* whose vertices represent proteins and whose edges represent protein interactions; each edge can be assigned a weight indicating the strength of evidence for the existence of the corresponding interaction. An important class of protein signaling cascades can be described as chains of interacting proteins in which protein interactions enable each protein in the path to modify its successor so as to transmit biological information. Such structures correspond to simple paths in the protein interaction graph (Kelley *et al.*, 2003).

[1] A preliminary version of this paper appeared in *Proceedings of RECOMB 2005*.

[2] Computer Science Division, University of California, Berkeley, Berkeley, CA 94720.

[3] Department of Bioengineering, University of California, San Diego, La Jolla, CA 92093.

[4] International Computer Science Institute, Berkeley, CA 94704.

[5] School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel.

Steffen *et al.* (2002) studied the problem of identifying pathways in a protein network. They applied an exhaustive search procedure to an unweighted interaction graph, considering all interactions equally reliable. To score the biological relevance of an identified path, they scored the tendency of its genes to have similar expression patterns. The approach was successful in detecting known signaling pathways in yeast. We have also conducted a related study (Kelley *et al.*, 2003) aimed at identifying pathways that are conserved across two species. The study employed a more efficient way of detecting simple paths in a graph that is based on finding acyclic orientations of the graph's edges.

The present work advances the methodology of searching for signaling cascades in two ways: first, by assigning well-founded reliability scores to protein–protein interactions, rather than putting all such interactions on the same footing; and second, by exploiting a powerful algorithmic technique of Alon *et al.* (1995), called *color coding,* to find high-scoring paths efficiently. The color-coding approach reduces the running time of the search algorithm by orders of magnitude compared both to exhaustive search and to the faster acyclic orientation approach, thus enabling the search for longer paths. We also extend the color-coding method to incorporate biologically motivated constraints on the types of proteins that may occur in a path and the order of their occurrence and to search for structures more general than paths, such as trees and two-terminal series-parallel graphs.

As evidence of the success of our approach, we show that our method accurately recovers well-known MAP kinase and ubiquitin-ligation pathways, that many of the pathways we discover are enriched for known cellular functions, and that the pathways and trees we find score higher than those found in random networks obtained by shuffling the edges and weights of the original network while preserving vertex degrees.

The paper is organized as follows: Section 2 presents the path finding problem and describes the color-coding approach. In Section 3, we develop biologically motivated extensions of the color-coding approach. Section 4 describes the estimation of protein interaction reliabilities and the path scoring methods used. Finally, Section 5 presents the applications of our method to yeast protein interaction data.

2. FINDING SIMPLE PATHS: THE COLOR CODING TECHNIQUE

Alon *et al.* (1995) devised a novel randomized algorithm, called *color coding,* for finding simple paths and simple cycles of a specified length k, within a given graph. In this section, we describe this approach. Our presentation generalizes that found in Alon *et al.* (1995) in order to allow succinct description of biologically motivated extensions of the basic technique.

Consider a weighted interaction graph in which each vertex is a protein and each edge (u, v) represents an experimentally observed interaction between proteins u and v and is assigned a numerical value $p(u, v)$ representing the probability that u and v interact (computed as per Section 4 below). Each simple path in this graph can be assigned a *score* equal to the product of the values assigned to its edges. Among paths of a given length, those with the highest scores are plausible candidates for being identified as linear signal transduction pathways. Given a set I of possible start vertices, we would like to find the highest-scoring paths from I to each vertex of the graph. In the case of signaling pathways, I might be the set of all receptor proteins or a single protein of particular interest.

We begin by framing the problem mathematically. In order to work with an additive weight rather than a multiplicative one, we assign each edge (u, v) a weight $w(u, v) \equiv -\log p(u, v)$. We define the *weight* of a path as the sum of the weights of its edges and the *length* of a path as the number of vertices it contains. Given an undirected weighted graph $G = (V, E, w)$ with n vertices, m edges, and a set I of start vertices, we wish to find, for each vertex v, a minimum-weight simple path of length k that starts within I and ends at v. If no such simple path exists, our algorithm should report this fact. This problem is distinct from that of finding a path in G that somehow matches a given query path P, as in Zheng *et al.* (2002) (or, more generally, subgraph matching [Conte *et al.*, 2004]).

For general k, this problem is NP-hard, as the traveling-salesman problem is polynomial-time reducible to it. The difficulty of the problem stems from the restriction to simple paths; without this restriction, the best path of length k is easily found. A standard dynamic programming algorithm for the problem is as follows. For each nonempty set $S \subseteq V$ of cardinality at most k and each vertex $v \in S$, let $W(v, S)$ be the minimum weight of a simple path of length $|S|$ which starts at some vertex in I, visits each vertex in S, and ends at v. If no such path exists, then $W(v, S) = \infty$. The following recurrence can be used to tabulate this function by generating the values $W(v, S)$ in increasing order of the cardinality of S:

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

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

The weight of the optimal path to v is the minimum of $W(v, S)$ over all pairs v, S such that $|S| = k$, and the vertices of the optimal path can be recovered successively in reverse order by a standard dynamic programming backtracking method. The running time of this algorithm is $O(kn^k)$ and its space requirement is $O(n^k)$.

The idea behind color coding is to assign each vertex a random color between 1 and k and, instead of searching for paths with distinct vertices, search for paths with distinct colors. The complexity of the dynamic programming algorithm is thereby greatly reduced, and the paths that are produced are necessarily simple. However, a path fails to be discovered if any two of its vertices receive the same color, so many random colorings need to be tried to ensure that the desired paths are not missed. The running time of the color coding algorithm is exponential in k and linear in m, and the storage requirement is exponential in k and linear in n. This method is superior when n is much larger than k, as is the case in our application, where typical values are $n = 4{,}500$ and $k = 8$.

The color coding algorithm requires repeated randomized trials. In each trial, every vertex $v \in V$ is independently assigned a color $c(v)$ drawn uniformly at random from the set $\{1, 2, \ldots, k\}$. Call a path *colorful* if it contains exactly one vertex of each color. We seek a minimum-weight colorful path from I to each vertex v. This problem can be solved using the following dynamic programming algorithm, which parallels the previous one: for each nonempty set $S \subseteq \{1, 2, \ldots, k\}$ and each vertex v such that $c(v) \in S$, let $W(v, S)$ be the minimum weight of a simple path of length $|S|$ that starts within I, visits a vertex of each color in S, and ends at v. This function can be tabulated (again, in increasing order of the cardinality of S) using the following recurrence:

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

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

The weight of a minimum-weight colorful path ending at v is $W(v, \{1, 2, \ldots, k\})$. For every vertex v, each trial yields a simple path of length k starting within I and ending at v, which is optimal among all the paths that are colorful under the random coloring in that trial. The running time of each trial is $O(2^k km)$ and the storage requirement is $O(2^k n)$. For any simple path P of length k, the probability that the vertices of P receive distinct colors in a given trial is $k!/k^k$, which is at least e^{-k} and is well approximated by $\sqrt{2\pi k}e^{-k}$. Thus, the chance that a trial yields an optimal path to v for our original problem is at least e^{-k}; for any $\epsilon \in (0, 1)$, the chance that the algorithm fails to find such a path in $e^k \ln \frac{1}{\epsilon}$ trials is at most ϵ. After $e^k \ln \frac{n}{\epsilon}$ trials, the probability that there exists a vertex v for which an optimal path has not been found is at most ϵ.

3. EXTENSIONS OF THE COLOR CODING METHOD

In this section, we present color-coding solutions to several biologically motivated extensions of the basic path-finding problem. These include (1) constraining the set of proteins occurring in a path, (2) constraining the order of occurrence of the proteins in a path, and (3) finding pathway structures that are more general than simple paths.

3.1. Constraining the set of proteins

To ensure that a colorful path produced by our algorithm contains a particular protein, we can simply assign a color uniquely to that protein. By adding counters to the state set of the dynamic programming recurrence, we can control the number of occurrences in the path of proteins from a specific family (e.g., proteins with a specific function). To enforce the constraint that our path must contain at least a and at most b proteins from a set T, we can define $W(v, S, c)$ as the minimum weight of a path of length $|S|$ ending at v that contains a vertex of each color in S and exactly c proteins from T. Here c ranges between 0 and b. This extension multiplies the storage requirement and running time of each trial by $b + 1$. Several counters can be added to enforce different constraints; each multiplies the time and storage requirement by a constant factor and does not affect the probability that the optimal path is colorful in any given trial.

3.2. Constraining the order of occurrence: Segmented pathways

In many signaling pathways, the proteins occur in an inward order, from membrane proteins to nuclear proteins and transcription factors (as will later be shown in Fig. 3a). The color-coding method can be adapted to restrict attention to paths that respect such an ordering.

3.2.1. Unique labeling. We restrict attention to simple paths that are the concatenation of t (possibly empty) ordered segments, where each segment contains proteins from a particular class (such as membrane proteins), and each protein is assigned to exactly one class. Subject to this restriction, we seek, for each vertex v, a minimum-weight simple path of length k from some vertex in I to v. The segments are numbered successively in the order of their occurrence along the desired path. Depending on biological information (e.g., cellular component annotation), each protein u is assigned an integer label $L(u)$ which uniquely specifies the segment in which the protein may occur. We require that the labels of the proteins along the path form a monotonically nondecreasing sequence. Such a path is called *monotonic*.

As usual, in each trial we assign each vertex a color drawn uniformly at random from $\{1, 2, \ldots, k\}$. Since each vertex is restricted to a unique segment, the path will be simple provided that vertices in the same segment are assigned different colors. For a vertex v and a subset of the colors $S \supseteq \{c(v)\}$, $W(v, S, k)$ is defined as the minimum weight of a simple monotonic path of length k from I to v, in which no two vertices with the same label have the same color, and the set of colors assigned to vertices with label $L(v)$ is S. We can tabulate this function using the following recurrence:

$$W(v, \{c(v)\}, l) = \min_{u:L(u)<L(v)} \min_{S} W(u, S, l-1) + w(u, v),\, l > 1$$

$$W(v, S, l) = \min_{u:L(u)=L(v), c(u)\in S\setminus\{c(v)\}} W(u, S \setminus \{c(v)\}, l-1) + w(u, v),\, 1 < |S| \leq l$$

where $W(v, \{c(v)\}, 1) = 0$ if $v \in I$ and ∞ otherwise.

Suppose there are at most h vertices in each segment. Then each trial has a running time of $O(2^h hkm)$ and a storage requirement of $O(2^h kn)$. For any simple path, the probability that all vertices in each segment receive distinct colors is at least e^{-h}. Thus, the expected number of trials to discover an optimal segmented pathway with at most h proteins per segment is of order e^h, which is much smaller than e^k—the upper bound on expectation in the nonsegmented case.

3.2.2. Interval restrictions. It may be unrealistic to assume that every protein can be assigned a priori to a unique segment. Instead, we can assume that, for each protein, there is a lower bound $L_1(u)$ and an upper bound $L_2(u)$ on the number of its segments. For example, if the successive segments correspond to membrane, cytoplasm, nucleus, and transcription factor proteins, then a protein that is neither a membrane protein nor a transcription factor will have a lower bound of 2 and an upper bound of 3.

A path $(u_1, u_2, \ldots, u_k)$ is *consistent with segmentation* if it is possible to assign to each protein u_i a segment number s_i such that the sequence of segment numbers along the path is monotonically nondecreasing and, for each i, $L_1(u_i) \leq s_i \leq L_2(u_i)$. We can reformulate this condition as follows: for any path P, let $s(P)$ be the maximum, over all proteins u in P, of $L_1(u)$. Then the path $(u_1, u_2, \ldots, u_k)$ is consistent with segmentation if and only if, for all i, $L_2(u_i) \geq s(u_1, u_2, \ldots, u_{i-1})$.

Let each vertex u be assigned a color $c(u)$ drawn uniformly at random from $\{1, 2, \ldots, k\}$. For each vertex v, the color-coding method seeks a minimum-weight path of length k from I to v which is both colorful and consistent with segmentation. Define $W(v, s, S)$, where $L_1(v) \leq s \leq L_2(v)$, as the minimum weight of a simple path P of length $|S|$ from I to v that is consistent with the segmentation, such that $s(P) = s$ and S is the set of colors assigned to the vertices in P. We obtain the following dynamic programming recurrence:

$$W(v, L_1(v), S) = \min_{u:c(u)\in(S\setminus\{c(v)\})} \min_{s'\leq L_1(v)} W(u, s', S \setminus \{c(v)\}) + w(u, v), |S| > 1$$

$$W(v, s, S) = \min_{u:c(u)\in(S\setminus\{c(v)\})} W(u, s, S \setminus \{c(v)\}) + w(u, v), L_1(v) < s \leq L_2(v), |S| > 1$$

where $W(v, L_1(v), \{c(v)\}) = 0$ if $v \in I$ and ∞ otherwise. The weight of a minimum-weight colorful path ending at v and consistent with the segmentation is $\min_s W(v, s, \{1, 2, \ldots, k\})$. The running time per trial is $O(2^k ktm)$ and the storage requirement is $O(2^k tn)$, where t is the number of ordered segments.

3.3. *Finding more general structures*

In general, signaling pathways need not consist of a single path. For instance, the high osmolarity pathway in yeast starts with two separate chains that merge into a single path (Steffen *et al.*, 2002). We shall demonstrate that the color-coding method can be used to find high-scoring signaling pathways with a more general structure. Our principal examples are rooted trees, which are common when several pathway segments merge, and two-terminal series-parallel graphs, which capture parallel signaling pathways.

3.3.1. Rooted trees. Let $G = (V, E)$ be a weighted graph with $I \subset V$, and let k be a positive integer. For each vertex v, we wish to find a tree of minimum weight among all k-vertex subtrees in G that are rooted at v and in which every leaf is an element of I. In each color coding trial, we assign each vertex u a color drawn uniformly at random from $\{1, 2, \ldots, k\}$. For $v \in V$ and $\{c(v)\} \subseteq S \subseteq \{1, 2, \ldots, k\}$, let $W(v, S)$ be the minimum weight of a subtree with $|S|$ vertices that is rooted at v, contains a vertex of each color in S, and whose leaves lie in I. The following recurrence can be used to compute $W(v, S)$:

$$W(v, S) = \min \left\{ \begin{array}{l} \displaystyle\min_{u:c(u)\in S\setminus\{c(v)\}} W(u, S \setminus \{c(v)\}) + w(u, v), \\[2ex] \displaystyle\min_{(S_1,S_2):S_1\cap S_2=\{c(v)\},S_1\cup S_2=S} W(v, S_1) + W(v, S_2) \end{array} \right\}$$

where $W(v, \{c(v)\}) = 0$ if $v \in I$ and ∞ otherwise. The running time for a trial is $O(3^k km)$ and the storage required is $O(2^k n)$. The factor of 3^k in the running time comes from the second line of the recurrence, which examines for each subset of S all possible bipartitions. This is equivalent to 3^k, the number of ways to divide k colors into three distinct groups.

3.3.2. Two-terminal series-parallel graphs. The definition of a two-terminal series-parallel graph (2SPG) is recursive:

- *Base case.* The graph with two vertices u and v connected by an edge is a 2SPG between terminals u and v.
- *Series connection.* If G_1 is a 2SPG between u and v, G_2 is a 2SPG between v and w, and G_1 and G_2 have no vertices in common except v, then $G_1 \cup G_2$ is a 2SPG between u and w.
- *Parallel connection.* If G_1 and G_2 are 2SPGs between u and v and they have no vertices in common except u and v, then $G_1 \cup G_2$ is a 2SPG between u and v.

Our goal is to find, for each vertex v, a minimum-weight k-vertex 2SPG between some vertex in I and v. Let $W(u, v, S)$ be the minimum weight of a 2SPG between u and v with $|S|$ vertices in which the set of colors occurring is S. Then, following the recursive definition of a 2SPG, we obtain

$$W(u, v, S) = \min \left\{ \min_{w, S_1, S_2 : S_1 \cup S_2 = S, S_1 \cap S_2 = \{c(w)\}} W(u, w, S_1) + W(w, v, S_2), \right.$$
$$\left. \min_{T_1, T_2 : T_1 \cap T_2 = \{c(u), c(v)\}, T_1 \cup T_2 = S} W(u, v, T_1) + W(u, v, T_2) \right\}$$

where $W(u, v, \{c(u), c(v)\}) = w(u, v)$ for every edge (u, v). The execution time of a trial is $O(3^k kn^2)$ and the storage requirement is $O(2^k n^2)$.

4. ESTIMATION OF INTERACTION RELIABILITIES AND EVALUATION OF PATHS

Since experimental interaction data are notoriously noisy (see, e.g., Deng *et al.* [2003] and Bader *et al.* [2004]), estimating and incorporating the reliability of the observed interactions in the path detection process are key to its success. Several authors have suggested methods for evaluating the reliabilities of protein interactions (Bader *et al.*, 2004; Deng *et al.*, 2003; von Mering *et al.*, 2002). Karaoz *et al.* (2004) used one such method in the process of predicting protein function. Here, we use a method we have previously developed (Sharan *et al.*, 2005), which is based on a logistic regression model. For completeness, we describe it briefly. We define the probability of a true interaction as a logistic function of three observed random variables on a pair of proteins: (1) the number of times an interaction between the proteins was experimentally observed; (2) the Pearson correlation coefficient of expression measurements for the corresponding genes (using 794 expression profiles obtained from the Stanford Microarray Database (Gollub *et al.*, 2003)); and (3) the proteins' small world clustering coefficient (Goldberg and Roth, 2003), which is defined as minus the logarithm of the hypergeometric p-value for the overlap in the neighborhoods of two proteins.

According to the logistic distribution, the probability of a true interaction T_{uv} given the three input variables, $X = (X_1, X_2, X_3)$, is

$$\Pr(T_{uv}|X) = \frac{1}{1 + \exp\left(-\beta_0 - \sum_{i=1}^{3} \beta_i X_i\right)}$$

where $\beta_0, \ldots, \beta_3$ are the parameters of the distribution. Given training data, one can optimize the distribution parameters so as to maximize the likelihood of the data. As positive examples, we used the MIPS (Mewes *et al.*, 2004) interaction data, which is an accepted gold standard for yeast interactions. Motivated by the large fraction of false positives in interaction data, we considered observed interactions chosen at random as negative examples. We treated the chosen negative data as noisy indications that the corresponding interactions were false and assigned those interactions a probability of 0.1397 for being true, where this value was optimized using cross-validation.

Denote the reliability of an edge (u, v) by $p(u, v)$. We use the estimated probabilities to assign weights to the interaction graph edges, where edge (u, v) is assigned the weight $-\log p(u, v)$. Under these assignments, we seek minimum weight subnetworks of specified sizes.

We use two quality measures to evaluate the subnetworks we compute: weight p-value and functional enrichment. Given a subnetwork with weight w, its *weight p-value* is defined as the percent of top-scoring subnetworks in random networks (computed, in all cases, using the same algorithm and parameters that are applied to the real network—see below) that have weight w or lower, where random networks are constructed by shuffling the edges and weights of the original network, preserving vertex degrees. Edges whose weights differ by more than a factor of two are not shuffled in order to approximately preserve, for each vertex, the sum of the weights of the edges touching it.

To evaluate the *functional enrichment* of a subnetwork N, we associate its proteins with known biological processes using the gene ontology (GO) annotations (The Gene Ontology Consortium, 2000). We then compute the tendency of the proteins to have a common annotation using a method developed by Sharan *et al.* (2005). The scoring is done as follows: define a protein to be *below* a GO term t if it is associated with t or any other term that is a descendant of t in the GO hierarchy. For each GO term t with at least one protein assigned to it, we compute a hypergeometric p-value based on the following quantities: (1) the number of proteins in N that are below t; (2) the total number of proteins below t; (3) the number of proteins in N that are below all parents of t; and (4) the total number of proteins below all parents of t. The p-value is further Bonferroni corrected for multiple testing.

5. APPLICATION TO THE YEAST PROTEIN NETWORK

We implemented color-coding methods for finding both simple paths and rooted trees in a graph. The algorithm maintains a heap of the best subnetworks found throughout the iterations and, thus, is able to report suboptimal subnetworks in addition to optimal ones. Table 1 presents a benchmark of the running time of the path search across a network with ~4,500 nodes and ~14,500 edges (the yeast network described below), when varying the desired path length, the probability of success, and the number of required paths. In all these runs, the paths were constrained to start at a membrane protein and end at a transcription factor (see details below). The algorithm runs in minutes when searching for a path of length 8 with success probability of 99.9% and in just over three hours when searching for a path of length 10. In comparison, the running time of our implementation of an exhaustive search approach was approximately 1.5-fold lower for length-7 paths (69 seconds), 2-fold higher for length-8 paths (866 seconds), and 7-fold higher for length-9 paths (15,120 seconds).

Note that because the runtime for exhaustive search is proportional to the number of subnetworks to be examined, the difference in runtime between the color-coding and the exhaustive search algorithms grows much wider when the endpoints of the paths are not constrained, or when looking for more general structures such as trees. For instance, while the yeast network contains on the order of 91 million length-7 paths in the constrained case, the total number of unconstrained paths of the same length is over 28 billion. Since this difference has comparatively little impact on the color-coding method, the speedup in runtime for length-7 paths is already 117-fold (138 seconds for color coding, 16,387 seconds for exhaustive search).

We applied our algorithm to search for paths and rooted trees in the yeast protein interaction network. Protein–protein interaction data were obtained from the Database of Interacting Proteins (Xenarios *et al.*, 2002) (February 2004 download) and contained 14,319 interactions among 4,389 proteins in yeast.

As a first test, we applied the algorithm to compute optimal paths and trees of size 8. Paths were constrained to start at a membrane protein (GO:0005886 or GO:0004872) and end at a transcription factor (GO:0030528). Rooted trees were constrained to be rooted at a membrane protein and have all leaf nodes be transcription factors. In order to prevent a large number of minor variations on a small set of high-scoring subnetworks from dominating our results, the heap was used to filter subnetworks with more than 70% of their proteins in common (the one of lowest weight was retained, and all others removed).

TABLE 1. RUNNING TIMES OF THE PATH-FINDING ALGORITHM FOR DIFFERENT PARAMETER SETTINGS

Path length	*Success probability*	*# Paths*	*Time (sec)*
10	99.9%	100	11,650
9	99.9%	100	2,149
8	99.9%	500	498
8	99.9%	300	460
8	99.9%	100	435
8	90%	100	303
8	70%	100	269
8	50%	100	257
7	99.9%	100	97
6	99.9%	100	32

With success probability set to 99.9%, the 100 best subnetworks of each type were recorded, and for each we evaluated its weight p-value and functional enrichment. The results are depicted in Figs. 1 and 2. Clearly, both measures significantly exceed the random expectation. In particular, 68% of the identified paths and 63% of the identified trees had a significant biological process annotation ($p < 0.05$).

Next, we wished to test the utility of the algorithm in reconstructing known pathways in yeast. To this end, we concentrated on three MAPK signal transduction pathways that were also analyzed by Steffen *et al.* (2002): pheromone response, filamentous growth, and cell wall integrity. For each of the pathways, we searched the network for paths of lengths 6–10 using the pathway's endpoints to define the start and end vertices. In all cases, our results matched the known pathways well. We describe these findings in detail below.

The pheromone response (mating type) pathway prepares the yeast cells for mating by inducing polarized cell growth toward a mating partner, cell cycle arrest in G1, and increased expression of proteins needed for cell adhesion, cell fusion, and nuclear fusion. The main chain of this pathway (consisting of nine proteins) is shown in Fig. 3a. In addition, proteins Bem1p, Rga1p, Cdc24p, Far1p, Ste50p, and Ste5p contribute to the operation of the pathway by interacting with proteins in the main chain.

Looking for the optimal path of length 9 in the yeast network yielded the path depicted in Fig. 3b. This path mainly consists of proteins in the pheromone response pathway, with the exception of Kss1p, which is a MAP kinase redundant to Fus3p, and Akr1p, which is a negative regulator of this pathway. The occurrence of the latter protein is an artifact that arises because the direct link between Ste3p and Ste4p is missing from the interaction data.

The aggregate of all the paths that the algorithm computed between Ste3p and Ste12p, across a range of lengths (6–10), is depicted in Fig. 3c. All the proteins that we have identified are part of the pathway, except for Kss1p and Akr1p (see discussion above). A previous study by Steffen *et al.* (2002) reported similar results for this pathway. In comparison to Fig. 3c, Steffen *et al.* identified three additional proteins (Sst2p, Mpt5p, and Sph1p), which are related to the pathway, but are not part of the main chain.

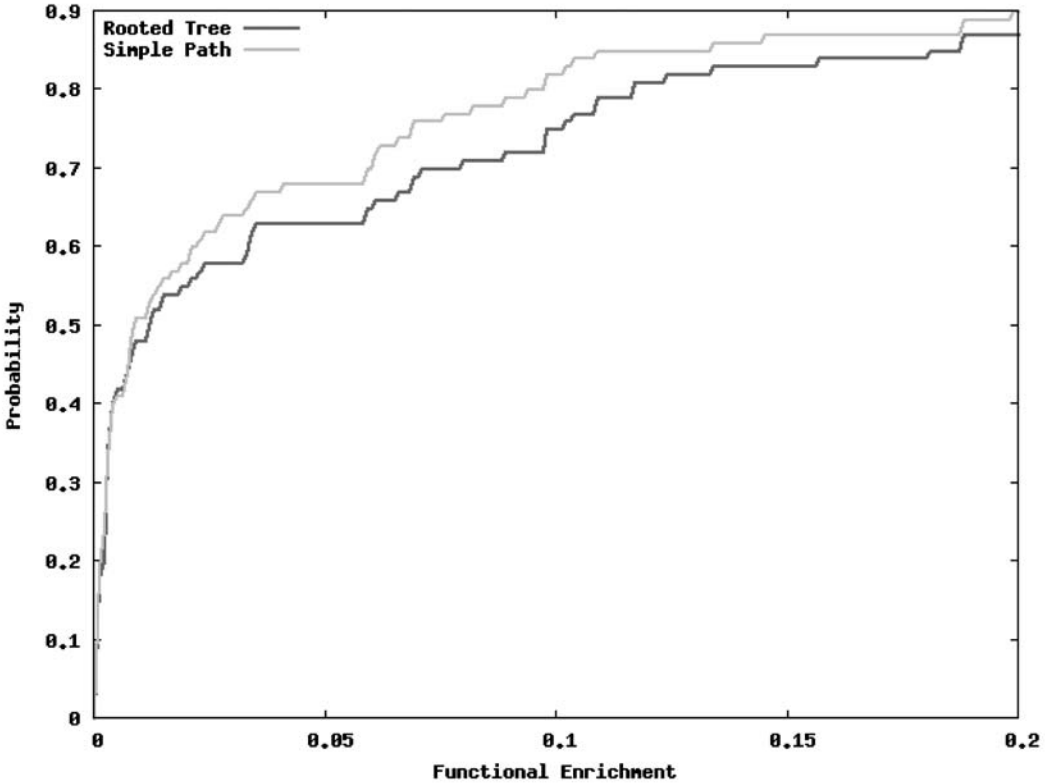

FIG. 1. Cumulative distributions of functional enrichment p-values; x-axis: p-value; y-axis: percent of paths or trees with p-value x or better.

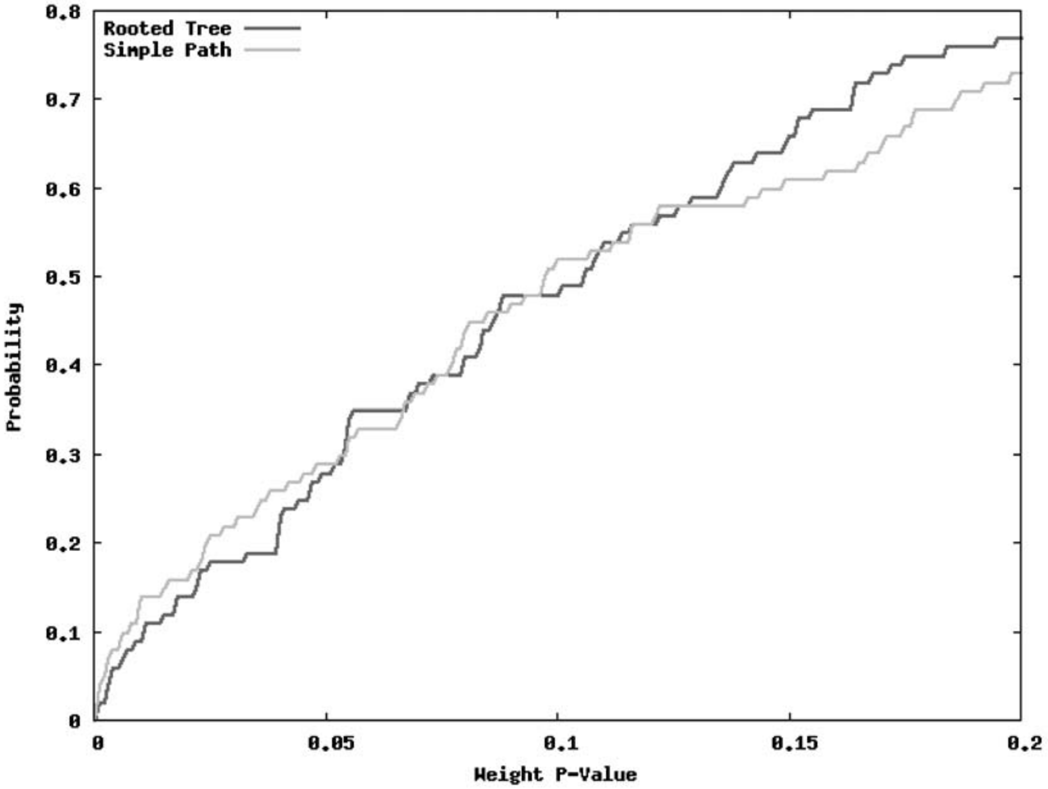

FIG. 2. Cumulative distributions of weight p-values; x-axis: p-value; y-axis: percent of paths or trees with p-value x or better.

Steffen *et al.* failed to recover the true positive Cdc42p. Interestingly, this latter protein participates mainly in paths of length 9 and 10 in our computations (only two additional paths of length 8 contained this protein). Such long paths are very costly to compute using an exhaustive approach.

The filamentous growth pathway is induced under stress conditions and causes yeast diploid cells to grow as filaments of connected cells. The pathway is depicted in Fig. 4a. Searching the network for the minimum-weight path of the same length as the known pathway (length 8), yielded the path shown in Fig. 4b, which largely matches the known pathway. The introduction of the proteins Cdc25p and Hsp82p is again an artifact that arises due to a missing link between Ras2p and Cdc42p in the network data.

The cell wall integrity pathway mediates cell cycle regulated cell wall synthesis. It is depicted in Fig. 4c. A search for the minimum-weight path of the same length starting at Ras2p and ending at Tec1p yielded the path shown in Fig. 4d. Again, the identified path matches the known pathway well. The only falsely detected protein, Rom2p, could be explained by the fact that the network does not contain a direct interaction between Mid2p and Rho1p.

In addition, we used our algorithm to search for the high osmolarity MAPK pathway, starting at Sln1p and ending at Hog1p (leading to several transcription factors, including Mcm1p and Msn2/4p [Roberts *et al.*, 2000]). For this run, although we could recover the exact known pathway, it was only the 11th-scoring among the 64 identified paths.

As a final test, we applied our algorithm to look for ubiquitin-ligation pathways by searching for paths of length 4–6 that start at a cullin (Cdc53p or Apc2p) and end at an F-box protein (Met30p, Cdc4p, or Grr1p). For each pair of endpoints, we output the best path for each specified length. To evaluate our success, we computed the enrichment of the identified proteins within the GO category “cellular catabolism” (GO:0044248). In total, 18 paths were computed, all of which were found to be highly enriched for this GO category ($p < 0.01$). A more careful examination of these paths revealed that they highly overlapped: In addition to their endpoints, these paths spanned four other proteins (Skp1p, Cdc34p, Hrt1p, and Sgt1p), all of which are known ubiquitin-ligation proteins.

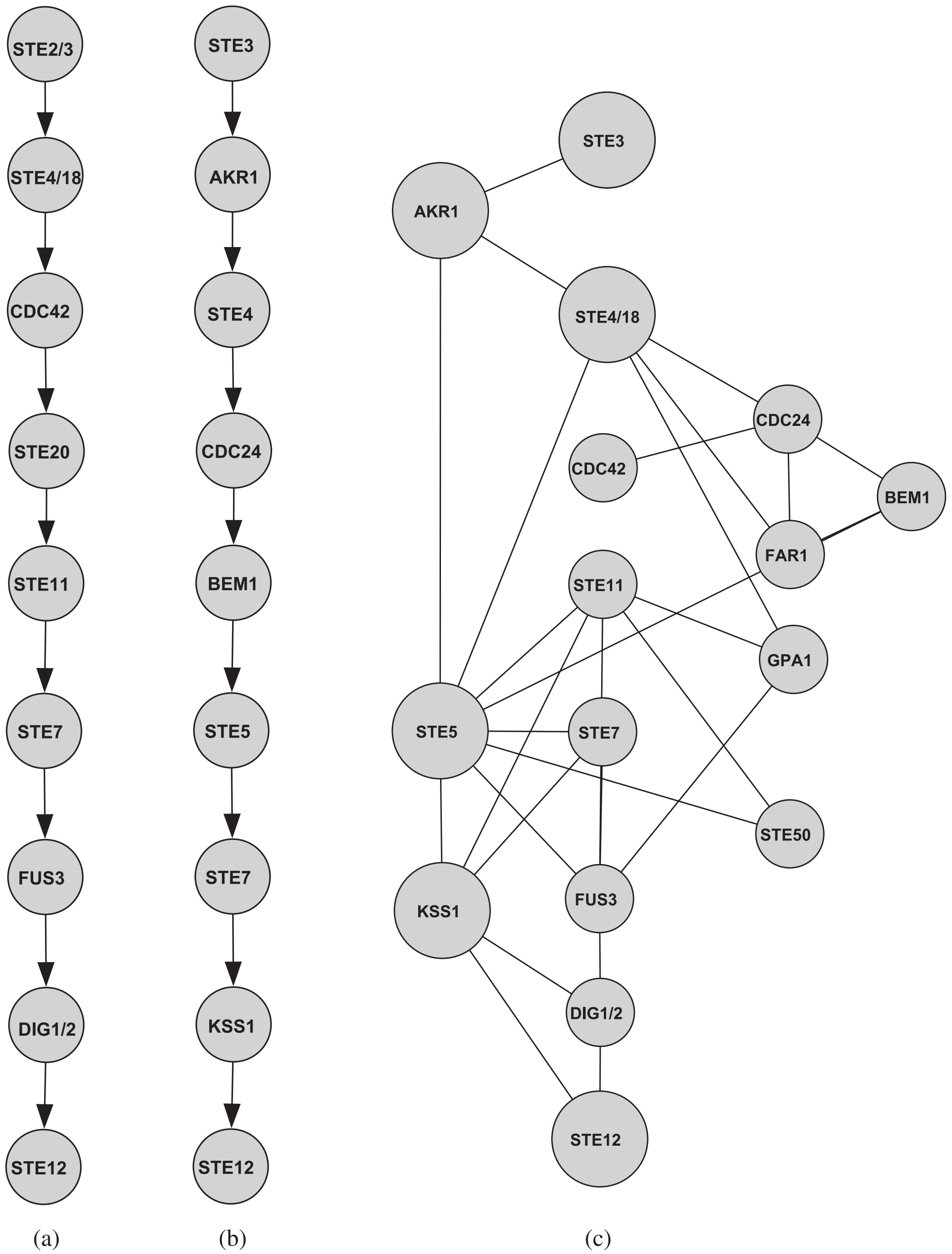

FIG. 3. The pheromone response signaling pathway in yeast. (**a**) The main chain of the known pathway, adapted from Roberts *et al.* (2000). (**b**) The best path of the same length (9) in the network. (**c**) The assembly of all light-weight paths starting at STE3 and ending at STE12 that were identified in the network. Nodes that occur in at least half of the paths are drawn larger than the rest. Nodes that occur in less than 10% of the paths are omitted.

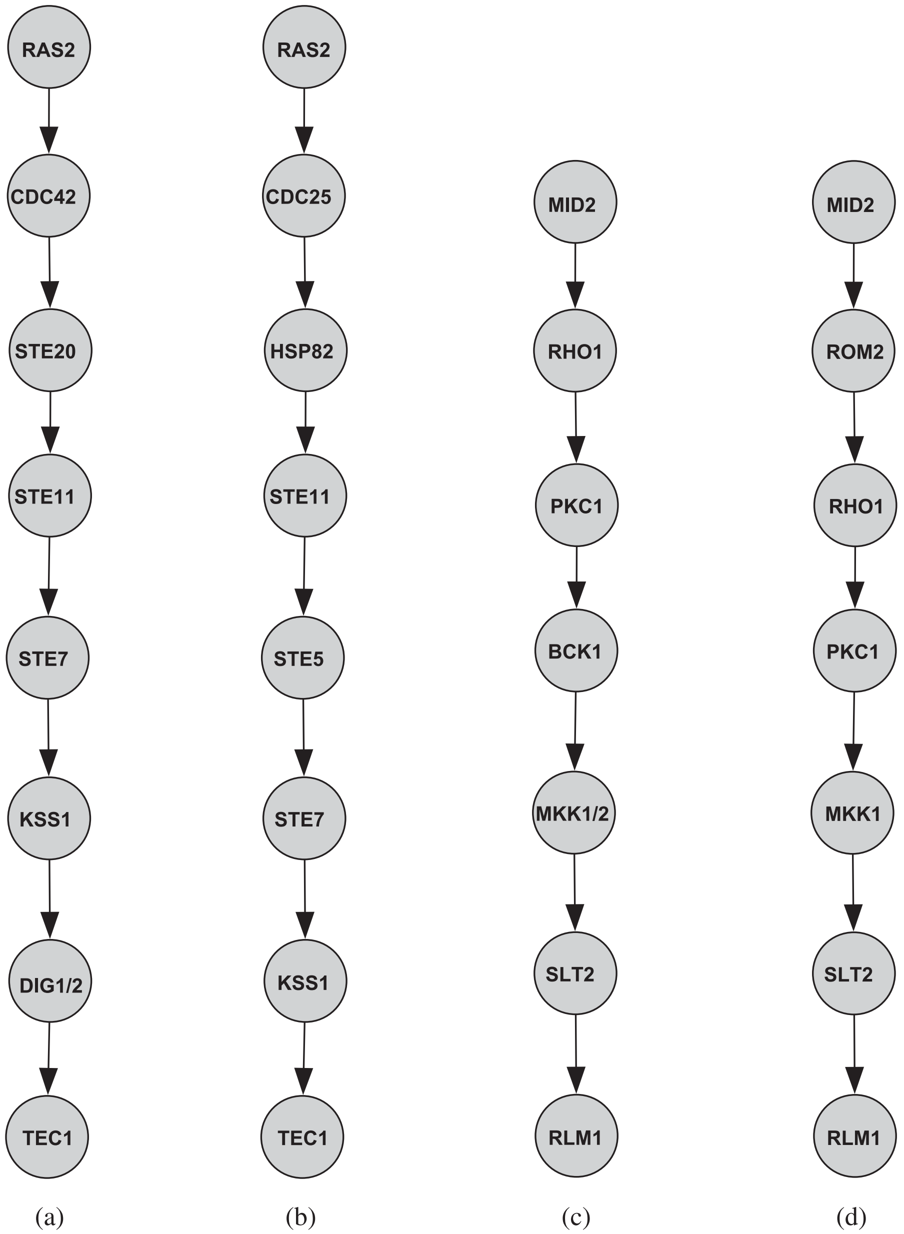

FIG. 4. Search results for the filamentous growth and cell wall integrity pathways. (**a**) The known filamentous growth pathway, adapted from Roberts *et al.* (2000). (**b**) The best path of length 8 between RAS2 and TEC1. (**c**) The known cell wall integrity pathway (Roberts *et al.*, 2000). (**d**) The best path of length 7 between MID2 and RLM1.

6. CONCLUSIONS

We have presented efficient algorithms for finding simple paths and rooted trees in graphs based on the color-coding technique and several biologically motivated extensions of this technique. We applied these algorithms to search for protein interaction pathways in the yeast protein network. Sixty-eight percent of the identified paths and 63% of the identified trees were significantly functionally enriched. We have also shown the utility of the algorithm in recovering known MAP-kinase and ubiquitin-ligation pathways in yeast. While these results are promising, there are a number of possible improvements that could be incorporated into this framework: (1) adapt the color-coding methodology to identify more general subnetworks, building on our ideas for detecting trees and two-terminal series-parallel subgraphs, and

(2) extend the framework to identify conserved pathways across multiple species, similarly to Kelley *et al.* (2003). In addition, our algorithms could be applied to other biological networks, most evidently to metabolic networks.

ACKNOWLEDGMENTS

The authors thank Noga Alon for proposing the application of the color-coding technique to finding paths in protein interaction graphs. R.S. was supported by an Alon fellowship. J.S. was supported by a Melvin M. Goldberg fellowship.

REFERENCES

Alon, N., Yuster, R., and Zwick, U. 1995. Color-coding. *J. ACM* 42, 844–856.

Bader, J., Chaudhuri, A., Rothberg, J., and Chant, J. 2004. Gaining confidence in high-throughput protein interaction networks. *Nature Biotechnol.*, 78–85.

Conte, D., Foggia, F., Sansone, C., and Vento, M. 2004. Thirty years of graph matching in pattern recognition. *IJPRAI* 12, 265–298.

Deng, M., Sun, F., and Chen, T. 2003. Assessment of the reliability of protein–protein interactions and protein function prediction. *Proc. 8th Pac. Symp. on Biocomputing*, 140–151.

The Gene Ontology Consortium. 2000. Gene ontology: Tool for the unification of biology. *Nature Genet.* 25, 25–29.

Goldberg, D., and Roth, F. 2003. Assessing experimentally derived interactions in a small world. *Proc. Natl. Acad. Sci. USA* 100, 4372–4376.

Gollub, J., Ball, C., Binkley, G., *et al.* 2003. The Stanford Microarray Database: Data access and quality assessment tools. *Nucl. Acids Res.* 31, 94–96.

Karaoz, U., Murali, T.M., Letovsky, S., Zheng, Y., Ding, C., Cantor, C., and Kasif, S. 2004. Whole-genome annotation by using evidence integration in functional-linkage networks. *Proc. Natl. Acad. Sci. USA* 101, 2888–2893.

Kelley, B., Sharan, R., Karp, R., Sittler, T., Root, D., Stockwell, B., and Ideker, T. 2003. Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *Proc. Natl. Acad. Sci. USA* 100, 11394–11399.

Mewes, H., Amid, C., Arnold, R., *et al.* 2004. MIPS: Analysis and annotation of proteins from whole genomes. *Nucl. Acids Res.* 32, D41–44.

Roberts, C., Nelson, B., Marton, M., *et al.* 2000. Signaling and circuitry of multiple MAPK pathways revealed by a matrix of global gene expression profiles. *Science* 287, 873–880.

Scott, J., Ideker, T., Karp, R., and Sharan, R. 2005. Efficient algorithms for detecting signaling pathways in protein interaction networks. *Proc. 9th Ann. Int. Conf. on Research in Computational Molecular Biology*, 1–13.

Sharan, R., Suthram, S., Kelley, R., Kuhn, T., McCuine, S., Uetz, P., Sittler, T., Karp, R., and Ideker, T. 2005. Conserved patterns of protein interaction in multiple species. *Proc. Natl. Acad. Sci. USA* 102, 1974–1979.

Steffen, M., Petti, A., Aach, J., D'haeseleer, P., and Church, G. 2002. Automated modelling of signal transduction networks. *BMC Bioinformatics* 3, 34–44.

von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S.G., Fields, S., and Bork, P. 2002. Comparative assessment of large-scale data sets of protein–protein interactions. *Nature* 417, 399–403.

Xenarios, I., Salwínski, L., Duan, X., Higney, P., Kim, S.M., and Eisenberg, D. 2002. DIP, the database of interacting proteins: A research tool for studying cellular networks of protein interactions. *Nucl. Acids Res.* 30, 303–305.

Zheng, Y, Szustakowski, J., Fortnow, L., Roberts, R., and Kasif, S. 2002. Computational identification of operons in microbial genomes. *Genome Res.* 12, 1221–1230.

Address correspondence to:
Roded Sharan
School of Computer Science
Tel-Aviv University
Tel-Aviv 69978, Israel

E-mail: roded@tau.ac.il