JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 13, Number 9, 2006
© Mary Ann Liebert, Inc.
Pp. 1546–1557

Hypothesis Generation in Signaling Networks

DEREK A. RUTHS,[1] LUAY NAKHLEH,[1] M. SRIRAM IYENGAR,[2]
SHRIKANTH A.G. REDDY,[3] and PRAHLAD T. RAM[3]

ABSTRACT

Biological signaling networks comprise the chemical processes by which cells detect and respond to changes in their environment. Such networks have been implicated in the regulation of important cellular activities, including cellular reproduction, mobility, and death. Though technological and scientific advances have facilitated the rapid accumulation of information about signaling networks, utilizing these massive information resources has become infeasible except through computational methods and computer-based tools. To date, visualization and simulation tools have received significant emphasis. In this paper, we present a graph-theoretic formalization of biological signaling network models that are in wide but informal use, and formulate two problems on the graph: the Constrained Downstream and Minimum Knockout Problems. Solutions to these problems yield qualitative tools for generating hypotheses about the networks, which can then be experimentally tested in a laboratory setting. Using established graph algorithms, we provide a solution to the Constrained Downstream Problem. We also show that the Minimum Knockout Problem is NP-Hard, propose a heuristic, and assess its performance. In tests on the Epidermal Growth Factor Receptor (EGFR) network, we find that our heuristic reports the correct solution to the problem in seconds. Source code for the implementations of both solutions is available from the authors upon request.

Key words: algorithms, biology, computational molecular biology, evolution, genomics, phylogenetic trees

1. INTRODUCTION

In this paper, we use the term “biological networks” to refer to cell signaling networks, chains of reactions involved in triggering, propagating, and processing signals within the cell. These networks regulate many cellular activities that are critical to the health of the cell and the larger systems to which it may belong. Altered biological networks have been implicated as the cause of many devastating diseases including cancer (Hunter, 2000), heart disease (Feldman et al., 2005), congenital abnormalities (Belloni et al., 1996), metabolic disorders (Hunter, 2000), and immunological abnormalities (Hunter, 2000).

Significant research efforts to identify and map biological networks, aided by new technologies and scientific methods, have amassed vast databases of molecules and putative interactions among them. Given

[1] Department of Computer Science, Rice University, Houston, Texas.
[2] UT School of Health Information Sciences, Houston, Texas.
[3] UT M.D. Anderson Cancer Center, Houston, Texas.

the immense scale of networks now in common use, computational techniques to filter, search, and reason about them have become indispensable.

Existing research on computational tools in this area has focused on three forms of analysis: visualization of the networks (Kitano et al., 2005; Funahashi et al., 2003; Oda et al., 2005; Aladjem et al., 2004), high-level parametric simulation (Tsavachidou and Liebman, 2002; Peterson, 1981), and detailed simulations of small subnetworks based on initial conditions, reaction rates, and other molecule and reaction-specific parameters (Meng et al., 2004; Fu et al., 2004; Phillips and Cardelli, 2005; Nagasaki et al., 2004). Because of the difficulty of determining these parameters, higher-level models are used whenever possible. Recent efforts have also developed hypothesis generation techniques that use model checking and formal verification in order to qualitatively reason about networks (Chabrier-Rivier and Fages, 2003; Chabrier-Rivier et al., 2004, 2005; Sriram, 2003; Eker et al., 2002; Tran et al., 2005). Hypothesis testing tools establish the set of most-likely outcomes of an experiment, providing insights into experimental design, thereby reducing the investment of time and labor-intensive laboratory work. Existing hypothesis generation tools require statements about the properties of individual reactions in networks, details that are often unavailable for many networks. In this paper, we present a framework for computational hypothesis testing that only depends on the simplest property of a reaction—its reactants and products. Our framework combines currently used graph-based network representations with graph algorithms. We also formalize two biologically significant problems useful for hypothesis testing.

The Constrained Downstream Problem seeks the set of reactions in a biological network that leads from one set of molecules to another, such that the set is constrained to include reactions from a given set and exclude reactions from another given set. This is a useful tool in the design of drugs to modify or inhibit certain biological functions while preserving others. At the signaling network level this would help to identify molecules or sets of molecules that have to be targeted to inhibit function of a sub-network while preserving signal flow to a different sub-network. A biological endpoint for this type of problem would be if one wanted to identify a molecule (or a set of molecules) to inhibit proliferation while at the same time preserving metabolic or secretary functions. We provide a polynomial-time algorithm for solving this problem.

The Minimum Knockout Problem seeks a minimum-size set of molecules whose removal (or knocking out) from the biological network makes the production of a set of molecules impossible given an initial set of molecules. The minimum knockout problem is very important in the identification of molecular targets for therapies, especially in cancer. Traditional chemotherapeutics function by killing rapidly dividing cells, the end result being both cancer cells as well as normal cells are killed, hence the hair loss and gastrointestinal side effects of these drugs. In the past few years there has been a great effort in developing drugs that specifically target signaling molecules that are aberrantly functioning in cancer cells. The clinical trials and data from these drugs show that they are limited in their ability, and function best in combination with other targeted drugs. Therefore, the biological problem here is to identify the optimal and minimal sets of molecules that have to be targeted to block network function. This will allow the development of therapeutics that can efficiently kill cancer cells while still preserving normal cells.

The rest of the paper is organized as follows. In section 2, we introduce a graph formalization of biological networks. In section 3, we formulate the Constrained Downstream problem, and present a polynomial-time solution of it. In section 4, we formulate the Minimum Knockout problem, prove its NP-hardness, and devise a randomized heuristic for solving it. In section 5, we analyze the accuracy and performance of the proposed heuristic for the Minimum Knockout Problem on a large biological network. In section 6, we conclude and outline future research directions.

2. BIOLOGICAL NETWORKS

Standard models of biological networks encompass various molecules and interactions among them that occur on and within the cell membrane. An example of such a model is given in Figure 1. These models consist of instances of two fundamental components: (1) a molecule, either inorganic (such as oxygen, O_2), or organic such as proteins, segments of DNA, RNA, or even complexes consisting of one or more molecules attached to one another; and (2) an interaction, which is a change that occurs to one or more molecules. A change to a molecule will either change a property of the molecule (activity and/or localization), bind one or more molecules together, or break one or more molecules apart.

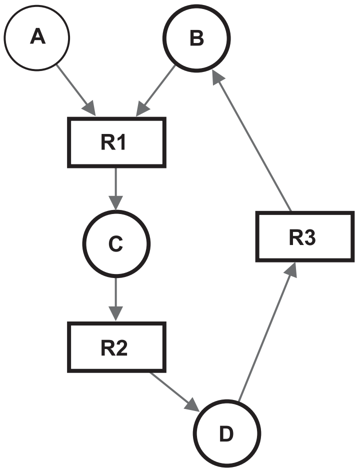

FIG. 1. A biological network that contains four molecules A, B, C, and D, and three interactions $R1$, $R2$, and $R3$.

A salient feature of biological networks is the common occurrence of feedback loops (for example, the loop $B \rightarrow R1 \rightarrow C \rightarrow R2 \rightarrow D \rightarrow R3 \rightarrow B$ in Fig. 1) in which a molecule a, through a series of interactions, gives rise to another molecule b that directly changes properties of molecule a through interactions. A negative feedback loop is a chain of interactions which decreases the activity of molecule a in the network; a positive feedback loop increases the activity of molecule a in the network.

2.1. Existing representations and models of biological networks

Standard models of biological networks have served as the basis for a number of computational models allowing simulating and reasoning about cell processes. We briefly review some of these models in this section.

The Systems Biology Markup Language (SBML) is an XML schema for representing biological networks in addition to regulatory and metabolic networks (SBML, 2005). The model uses chemical nomenclature: molecules are called “species” and interactions are called “reactions.” In the model, a reaction has reactant, product, and modifier species. Reactants and modifiers are both inputs to the reaction, differing only in that a modifier affects the reaction without undergoing any changes. The SBML file format and underlying network structure is well-accepted and supported by a large number of tools (SBML, 2005).

Petri Nets are considered hybrid models of networks because they model both the global topology of the network as well as some of the reaction-specific parameters that determine the quantitative behavior of the system. They have been used with varying success to simulate biological networks (Tsavachidou and Liebman, 2002). In translating the biological network into a Petri Net, each molecule and each interaction is given its own node. Each molecule is initialized with some number of “tokens,” which are then iteratively reallocated by the interactions to which they are connected. For a more detailed discussion of PetriNets, see Peterson (1981); for an example application, see Tsavachidou and Liebman (2002).

Differential and algebraic models attempt to simulate the quantitative characteristics of a network using mathematical formulae as well as constants and parameters that have been determined for each reaction in the network (Bhalla and Iyengar, 1999; Bhalla et al., 2002; Hoffmann et al., 2002; Smith et al., 2002; Meng et al., 2004; Fu et al., 2004; Phillips and Cardelli, 2005; Nagasaki et al., 2004). While undisputedly the most accurate of available techniques, these methods are not yet able to simulate large networks efficiently and accurately.

Model checking and formal verification techniques use logical models of networks to make qualitative assertions about their temporal properties (i.e., whether a certain reaction will ever take place under certain conditions). These tools have received significant attention due to their ability to support rapid qualitative hypothesis generation without requiring significant information specific to the networks of interest (Chabrier-Rivier and Fages, 2003; Chabrier-Rivier et al., 2004, 2005; Sriram, 2003; Eker et al., 2002; Tran et al., 2005). In contrast to differential, algebraic, and Petri Net models which require numerical parameters, logical models require qualitative properties of individual reactions. Further, the temporal logic mechanisms that these approaches use are limited in their expressive power vis-à-vis general querying of biological networks.

The rapid pace of lab-based research on biological networks forces biologists to deal with large numbers of reactions and molecules. The lack of much quantitative or qualitative data for these reactions and molecules, coupled with the complexity of questions that biologists would ask about the networks, significantly hinder the applicability and appropriateness of the techniques described above. There is a significant need for tools that provide hypothesis generation capabilities in the absence of detailed network information. For many known networks, the only experimentally confirmed detail is the existence of reactions and the identities of their reactants and products. To the best of our knowledge, hypothesis generation tools that operate on this information alone are lacking. In this paper, we propose a model and approach for such tools and provide a working implementation of two problems useful for hypothesis generation.

2.2. A graph formulation of biological networks

Numerous tools exist that visualize networks as graphs in which molecules and interactions appear as interconnected nodes (Kitano et al., 2005; Funahashi et al., 2003; Oda et al., 2005; Aladjem et al., 2004). Beyond visualization, PetriNets use parametric graphs to model biological systems. In this paper, we construct hypotheses using only graph topology. The model employed is essentially a non-parametric PetriNet in which only the graph topology is retained. One can view this simplification of the model as a level of abstraction, in which information not needed for solving the problems (e.g., the parameters of the PetriNet) is hidden. Here, we formalize this graph representation in order to provide a model on which hypothesis testing problems can be posed, analyzed and solved. In this section, we introduce this graph-theoretic model, which we call the "Pathway Graph."

Definition 1. *A* **Pathway Graph** *is a directed graph, $G = (V^\circ, V^\square, E)$, with two types of nodes:* **molecule-nodes**, *V°, and* **interaction-nodes**, *$V^\square$, with the following properties:*

1. $V^\circ \cap V^\square = \emptyset$ and
2. For every $(u, v) \in E$, u and v are not of the same type.

Property (1) in Definition 1 implies that each node in the graph is either a molecule-node or an interaction-node. Property (2) reflects that the fact that, biologically, a molecule cannot directly produce another molecule except through an interaction, nor can a reaction lead to another reaction except through a molecule.

The effect of the "removal" of molecule-nodes from the pathway graph is of significant interest to researchers because it models the effect of drugs that inhibit sections of the network. In particular, they are interested in the connectivity of the graph resulting from the removal of those nodes. Biologically, the effect of removing a node v in a pathway graph usually propagates further to other nodes reachable from v: interactions involving the removed molecule can no longer occur, the products of those interactions are no longer produced, and so on. Procedure *Remove* in Figure 2 is a formal description of the "propagation effect" of the removal of a node v in a pathway graph. An additional set of nodes, Y, is also specified to indicate the nodes at which to terminate the propagation.

Note that the *Remove* operation is a simplification of the behavior of the underlying biochemical system. In defining such an operation, we assume that a reaction cannot proceed at all when deprived of any of its reactants. In biochemical systems this does not always hold as certain enzymes have the effect of amplifying or *activating* the reaction. In the absence of an activating enzyme, the reaction may still proceed, only at a much slower rate. Because the goal of this work is hypothesis generation, and not simulation, we believe this assumption reasonable, though in future work we will continue to refine and modify the assumptions to yield more accurate predictions.

The Remove procedure can be extended to apply to a set of nodes in a straightforward manner: *Remove*$(G(V^\circ, V^\square, E),\ X,\ Y)$. In this case, *Remove* is applied successively to the nodes in X (this application yields the same result regardless of the order of nodes).

```
Remove($G(V^\circ, V^\square, E), v, Y \subseteq V^\circ$)

  1. If $v \in Y$, Return.
  2. Let $X \subseteq V^\circ \cup V^\square$ be the set of children of $v$.
  3. For every $x \in X$
      (a) Delete edge $(v, x)$ from $E$
      (b) If $x \in V^\circ$ and $indegree(x) = 0$
          i. Remove($G$, $x$, $Y$).
      (c) If $x \in V^\square$
          i. Remove($G$, $x$, $Y$).
  4. Delete $v$ from $V^\circ \cup V^\square$.
```

FIG. 2. The *Remove* procedure for removing a node from a pathway graph and propagating its effect.

3. THE CONSTRAINED DOWNSTREAM PROBLEM

A critical piece of information necessary to predict the outcome of biological network experiments is the set of molecules and interactions dependent on a given set of molecules and/or interactions. In the pathway graph model, all elements in the graph that are dependent on (*downstream from*) a set of molecules and interactions are reachable from that set.

Because of feedback loops in the network, often the set of downstream nodes will contain a significant portion of the network, sometimes the entire network. In order to reduce the number of downstream nodes returned, a biologist may choose to apply certain constraints to the downstream node search. Constraints restrict the solution to the set of nodes belonging to a path from nodes in set S to nodes in set T that includes one or more nodes contained in set I and not containing any nodes in set X. The exclusion of set X of nodes in this context means the removal of the nodes in X through the Remove procedure. These nodes belong to a subset of all possible downstream nodes. This is a useful tool in the design of drugs to modify or inhibit certain biological functions while preserving others. At the signaling network level this would help to identify molecules or sets of molecules that have to be targeted to inhibit function of a sub-network while preserving signal flow to a different sub-network.

Problem 1. THE CONSTRAINED DOWNSTREAM PROBLEM

Input: *Pathway graph* $G = (V^\circ, V^\square, E)$ *and four sets* $S, T \subset V^\circ$ *and* $I, X \subset (V^\circ \cup V^\square)$.
Output: *Subgraph* $G' = (U^\circ, U^\square, E')$ *where*

1. $U^\circ \subseteq V^\circ$, $U^\square \subseteq V^\square$, *and* $E' \subseteq E$;
2. $\forall u \in (U^\circ \cup U^\square)$, $\exists [s \in S\ ,\ t \in T]$ *such that* $s \overset{G'}{\rightsquigarrow} u$ *and* $u \overset{G'}{\rightsquigarrow} t$[1];
3. $(U^\circ \cup U^\square) \cap X = \emptyset$;
4. *every path from a node in* S *to a node in* T *passes through a node in* I; *and*
5. G' *is the maximum subgraph that satisfies conditions 1–4.*

The algorithm in Figure 3 solves the Constrained Downstream Problem.

Theorem 1. *Algorithm FindDownstream runs in time* $O(|V^\circ \cup V^\square|)$, *where* V° *and* $V^\square$ *are the nodes of the input graph* G.

[1] We write $x \overset{G}{\rightsquigarrow} y$ to denote that node y is reachable from node x in graph G.

```
$G'$ = FindDownstream($G = (V^{\circ}, V^{\square}, E)$, $S$,$T$,$I$,$X$)

  1. $G = Remove(G, X, T)$;
  2. $\forall t \in T$, $Visited[t] = 1$;
  3. $\forall t \notin T$, $Visited[t] = 0$;
  4. $\forall t \in T$, $OnPath[t] = 1$;
  5. $\forall t \notin T$, $OnPath[t] = 0$;
  6. $\forall v \in V^{\circ} \cup V^{\square}$, $AboveInclude[v] = 0$;
  7. $G' = (\emptyset, \emptyset, \emptyset)$;
  8. For every $s \in S$
     $CalcDownstream(G, s, I, \emptyset, G')$.
  9. Return $G'$;

CalcDownstream($G, v, I, P, G'$)

  1. If $Visited[v] == 0$
     (a) $Visited[v] = 1$;
     (b) Let $C$ be the children of $v$;
     (c) For every $c \in C$
         $CalcDownstream(G, c, (p_1, \dots, p_k, v))$;
  2. Else
     (a) If $OnPath[v] == 0$
         Return;
     (b) $\forall p \in P$, $OnPath[p] = 1$;
     (c) If $AboveInclude[v] == 1$
           i. $V_{G'} = V_{G'} \cup P$;
          ii. $E_{G'} = E_{G'} \cup \{(p_1, p_2), \dots, (p_{k-1}, p_k)\}$;
         iii. $\forall p \in P$, $AboveInclude[p] = 1$;
     (d) Else If $P \cap I \neq \emptyset$
           i. $V_{G'} = V_{G'} \cup P$;
          ii. $E_{G'} = E_{G'} \cup \{(p_1, p_2), \dots, (p_{k-1}, p_k)\}$;
         iii. $\forall p \in P$, $AboveInclude[p] = 1$;
     (e) Else, Return;
```

FIG. 3. The algorithm for the Constrained Downstream Problem.

Proof. *CalcDownstream* executes a depth first search, which has time complexity $O(|V^{\circ} \cup V^{\square}|)$. The intersection performed on line 2d can be computed in constant time because the path is built incrementally. All remaining assignments of *AboveInclude* and *OnPath* are performed at most once for any given node. Thus, the time complexity of all calls to label a node is $O(|V^{\circ} \cup V^{\square}|)$. As a result, the overall time complexity for the Constrained Downstream algorithm is $O(|V^{\circ} \cup V^{\square}|)$. ■

Next we prove that the algorithm in Figure 3 always provides the correct solution.

Theorem 2. *Let $\langle G, S, T, I, X \rangle$ be an input to the algorithm FindDownstream, and G' be the output of the algorithm on the input. Then, G' satisfies conditions (1)–(5) in Problem 1.*

Proof. The only nodes in G' are added by steps 2c and 2d in CalcDownstream. Since these two steps add only nodes and edges from G, condition (1) in Problem 1 follows.

Step 8 in FindDownstream ensures that every path in G' is starts from a node $s \in S$, and step 1c in CalcDownstream ensures that these paths are continued from these nodes s. Further, since only nodes $t \in T$ are initialized with $Visited = 1$ (step 2 in FindDownstream), CalcDownstream adds to G' only nodes and edges that are on paths started from an $s \in S$ and terminated in some $t \in T$. Hence, G' satisfies condition (2) in Problem 1.

Since the Remove procedure is applied to all nodes in X in step 1 in FindDownstream, G does not include any of these nodes, and so does G'. Hence, G' satisfies condition (3) in Problem 1.

Step 2d in CalcDownstream ensures that every path in G' passes through at least one node in I. Hence, G' satisfies condition (4) in Problem 1.

Finally, assume G'' is a graph that satisfies conditions (1)–(5) in Problem 1 such that the number of nodes in G'' is larger than that in G', and let x is a node in G'' that is not in G'. Then x is reachable from a node $s \in S$, reaches a node $t \in T$, and $x \notin I$. Therefore, x would be added to $V_{G'}$ in either step 2c or 2d in CalcDownstream, which implies x is a node in G'. Therefore, G' is the maximum subgraph that satisfies all conditions in Problem 1. ■

4. THE MINIMUM KNOCKOUT PROBLEM

A problem of significant interest to experimental biologists researching networks implicated in disease is the minimum knockout problem. In this problem, for a given pathway graph, a minimal set of nodes is sought such that the removal of these nodes disconnects a given set of (source) molecules, $S \subset V^\circ$, from another given set of (target) molecules, $T \subset V^\circ$. The minimum knockout problem is very important in the identification of molecular targets for therapies, especially in cancer. Traditional chemotherapeutics function by killing rapidly dividing cells, the end result being both cancer cells as well as normal cells are killed, hence the hair loss and gastro-intestinal side effects of these drugs. Therefore, the biological problem here is to identify the optimal and minimal sets of molecules that have to be targeted to block network function. Formally, we define the problem (decision version) as follows.

Problem 2. The Minimum Knockout Problem (MKO)

Input: *Pathway graph $G = (V^\circ, V^\Box, E)$, two sets of nodes $S, T \subset V^\circ$, and a positive integer Q.*
Question: *Does there exist a set $U \subseteq ((V^\circ \cup V^\Box) - (S \cup T))$ with $|U| \leq Q$ such that Remove(G,U, $S \cup T$) yields graph $G' = (V'^\circ, V'^\Box, E')$ in which for every $s \in S$ and $t \in T$, $s \not\rightsquigarrow t$?*

We first prove that MKO is NP-Hard, and then present an efficient and accurate randomized heuristic for solving it. We prove the NP-hardess of the problem by a reduction from the Minimum Set Cover Problem (Karp, 1972).

Problem 3. The Minimum Set Cover Problem (MSC)

Instance: *Collection C of subsets of a finite set B and a positive integer $K \leq |C|$.*
Question: *Does C contain a cover for B of size K or less, i.e., a subset $C' \subseteq C$ with $|C'| \leq K$ such that every element of B belongs to at least one member of C'?*

Theorem 3. *MKO is NP-Hard.*

Proof. Given an instance $\langle B = \{b_1, \ldots, b_m\}, C = \{C_1, \ldots, C_n\}, K\rangle$ of MSC, we construct an instance $\langle G, S, T, Q\rangle$ of MKO as follows.

- Pathway graph $G = (V^\circ, V^\Box, E)$ where
  $V^\circ = \{s_i, u_i : C_i \in C\} \cup \{t_i : b_i \in B\}$.
  $V^\Box = \{f_i : C_i \in C\} \cup \{g_i : b_i \in B\}$.
  $E = \{(s_i, f_i) : 1 \leq i \leq n\} \cup \{(f_i, u_i) : 1 \leq i \leq n\} \cup \{(u_i, g_j) : b_j \in C_i\} \cup \{(g_i, t_i) : 1 \leq i \leq m\}$.

- $S = \{s_i \in V^\circ\}$.
- $T = \{t_i \in V^\circ\}$.
- $Q = K$.

Figure 4 gives an example of the construction. The graph G constructed by the reduction satisfies the conditions of Definition 1, and hence it is a pathway graph. We now establish the validity of the reduction by showing that $\langle B, C, K\rangle$ is a yes-instance of MSC if and only if $\langle G, S, T, Q\rangle$ is a yes-instance of MKO.

$\Rightarrow$ Let $C' \subseteq C$ with $|C'| \leq K$ be a cover for B. Then, by construction, every node in the set $Y = \{g_i \in V^\square\}$ has an incoming edge from a node in the set $X = \{u_i : C_i \in C'\}$. Since Y contains only interaction nodes, applying Remove (Fig. 2) to all nodes in X will disconnect all paths from nodes in S to nodes in T. Since $|X| = |C'| \leq K$ and $Q = K$, it follows that $\langle G, S, T, Q\rangle$ is a yes-instance of MKO.

$\Leftarrow$ Assume there does not exist a cover of size K or less for B. Then, for every set $C' \subseteq C$ with $|C'| \leq K$, there is at least one $b' \in B$ such that $b' \notin \cup_{c\in C'} c$. By construction of G, it follows that for any subset of $X = \{u_i : C_i \in C\}$ of size Q or less, there exists at least one node in $Y = \{g_i \in V^\square\}$ that is not a child of any node in X. Hence, removing all nodes in X will not disconnect all paths from S to T. Since every node in T has a unique parent in Y, it follows that there does not exist a set of nodes of size Q or less that disconnects all paths from nodes in S to nodes in T. Hence, $\langle G, S, T, Q\rangle$ is not a yes-instance of MKO. This finishes the proof, thus establishing that MKO is NP-hard. ■

4.1. An efficient and accurate randomized heuristic for MKO

We now give an efficient and accurate heuristic for solving MKO; the heuristic is an iterative randomized search, with running time $O(nmk)$, where n is the number of nodes in the input pathway graph, m is the number of nodes in the constrained downstream subgraph, and k is the number of iterations. In the worst-case scenario, $m = n$; however, in our experiments on a large pathway graph, we found that $k, m << n$. The heuristic is outlined in Figure 5, and makes use of the following lemma.

Lemma 1. *Let U be a minimum knockout set for S and T in graph $G_d = Downstream(G, S, T)$, where $G_d = (V_d^\circ, V_d^\square, E_d)$. Then, there exists a minimum knockout set U' such that*

1. $|U'| = |U|$ *and*
2. $U \subseteq (V_d^\circ \cup (Children(S) \cap V_d^\square))$.

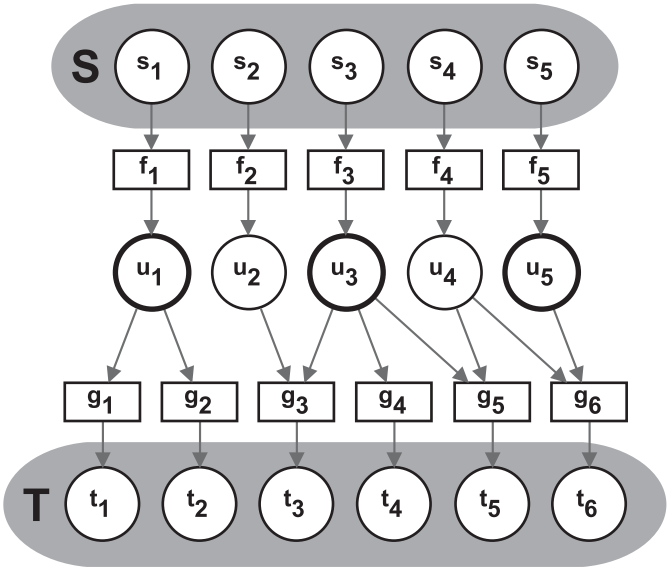

FIG. 4. The G, S, and T components of the MKO instance constructed by the reduction in the proof of Theorem 3 for the MSC instance with $B = \{b_1, b_2, b_3, b_4, b_5, b_6\}$, $C = \{\{b_1, b_2\}, \{b_3\}, \{b_3, b_4, b_5\}, \{b_5, b_6\}, \{b_6\}\}$, and $K = 3$; $Q = 3$.

```
**MinKnockout**($G = (V^{\circ}, V^{\square}, E)$, $S$,$T$,$m$)
  1. $U = FindDownstream(G, S, T, \emptyset, \emptyset)$
  2. $U^{\circ} = U \cap V^{\circ}$
  3. $U^{\square} = U \cap V^{\square}$
  4. $C = \{u \in U : u \in U^{\circ} \vee u \in (Children(S) \cap U^{\square})\}$
  5. For $i = 1$ to $m$
      (a) $G' = G$
      (b) $S_i = \emptyset$
      (c) While $S \overset{G'}{\rightsquigarrow} T$
          i. $c \in (C - S_i)$
          ii. $G' = Remove(G', c, S, T)$
          iii. $S_i = S_i \cup \{c\}$
  6. $j = argmax_i |S_i|$
  7. Return $S_j$
```

FIG. 5. An iterative and randomized heuristic for MKO.

The formal proof is omitted due to space constraints. Intuitively, this lemma states that if node $v \in V^{\square}$ is an element of a solution to MKO, then v can be replaced by some $v' \in V^{\circ}$ where (v', v) is an edge in the graph. The validity of this lemma follows from the definition of the *Remove* procedure (Fig. 2). The only $V^{\square}$ nodes that cannot be replaced in this manner are the children of S (since elements of S cannot appear in the solution).

The intuition for the heuristic is to exhaustively search a small (relative to the size of the actual pathway graph) set of nodes for the smallest knockout set for S and T. By constructing the set of nodes so that it does contain a knockout set (though not necessarily a globally minimal knockout set), the algorithm is guaranteed to find a solution, though it may not be minimal.

Before we describe our heuristic, we review background material that will be used in the heuristic. Given a directed graph $G = (V, E)$, and two sets $S, T \subseteq V$, a path in G is an S–T path if it runs from a node in S to a node in T. A set $C \subseteq V$ is called S–T disconnecting if C intersects each S–T path (C may intersect $S \cup T$).

Theorem 4 (Menger's Theorem [Menger, 1927]). *Let $G = (V, E)$ be a directed graph and let $S, T \subseteq V$. Then, the maximum number of node-disjoint S–T paths is equal to the minimum size of an S–T disconnecting node set.*

Now, we are in position to describe our heuristic. We construct the search set, C (line 4), to have the properties of containing a knockout set and being small relative to the number of nodes in the entire pathway graph as follows.

1. $C =$ *FindDownstream*$(G, S, T, \emptyset, \emptyset)$. By Menger's theorem, a knockout set is contained in the set of nodes that comprise all paths connecting S and T because one such knockout set is a node from each disjoint path connecting S to T. Since the Constrained Downstream Problem constructs this set, C contains a knockout set. In addition, the constrained set of downstream nodes for most choices of S and T will contain far fewer nodes than the entire pathway graph.

2. $C = (C \cap V^{\circ}) \cup (C \cap Children(S))$. Lemma 1 states that, except for the nodes with elements of S as inputs, any $V^{\square}$ node that occurs in a minimum knockout set can be replaced by a single V° node. Thus, by searching only the V° nodes between S and T, the search set is further reduced in size and still contains a knockout set.

After constructing the search set C (lines 1–4), the algorithm performs m randomized searches over all nodes in set C for a knockout set. Within each search (loop on line 5c), a knockout set is iteratively constructed by removing a randomly selected node (line 5ci) from the graph until S and T are disconnected. Of the m knockout sets constructed, the knockout set with fewest members is returned.

While the heuristic does not guarantee an upper-bound on error, our experiments show that, for the Epidermal Growth Factor Receptor (EGFR) biological network (Oda et al., 2005), this heuristic finds a minimum-knockout set every time. We discuss the experiments and performance in more detail next.

5. RESULTS AND DISCUSSION

We studied the performance of the heuristic for the Minimum Knockout Problem on the biological data set published in Oda et al. (2005). In this work, the authors constructed a comprehensive EGFR signaling network. This network is known to have a significant role in cancer development and proliferation. Given the amount of research currently focused on this network, benchmarks for our heuristic on this network will likely give an accurate sense of how well the heuristic will perform.

The EGFR network contains 292 interactions involving 330 different molecules. The graph of this network is highly connected, and nearly half of the molecules reach between 350 and 400 nodes in the graph.

To test the heuristic, we manually selected 30 pairs of S and T node sets from the network. The only selection criteria applied was a rough attempt to choose S and T so that the nodes in opposite sets were far from one another, increasing the likelihood of non-trivial solutions. Beyond this, the nodes were selected at random. Sets varied in size from 1 to 10.

Heuristic accuracy. The heuristic was set to run for 100 iterations on each of the (S, T) pairs. In every case, the heuristic reported a minimum knockout set of size 1. Since the smallest possible minimum knockout set has cardinality equal to 1, we concluded that every time the heuristic correctly identified a minimum knockout set. This result is remarkable for two reasons.

(1) A minimum knockout set of 1 occurs with unexpected frequency. This result is best explained by the degree of connectivity in the network. Figure 6 shows that over half of the nodes in the network have very extensive connectivity within the graph. This is consistent with other studies of connectivity within biological networks (Ma'ayan et al., 2005; Barabasi and Oltvai, 2004; Wuchty et al., 2003). Because of the properties of the *Remove* operation, removing such a highly connected node from the graph will have global impact on the connectivity of other nodes.

(2) The heuristic correctly found a minimum knockout set every time. This is certainly a property of the network: if one chooses a molecule at random in the network, there is a 50% chance that it will connect to 400 other nodes in the network. Ultimately, while it is easy to envision cases that will be difficult for the heuristic to handle, our results indicate that the EGFR network, well-studied and important in research, has few difficult cases, if any.

We observed that, though correct, often the heuristic chose nodes which disconnected large sections of the graph from S. Biologically, it is favorable to target nodes that have the smallest global impact while still disconnecting S and T. While, preliminary analysis was not able to establish whether the disconnection of these nodes was necessary, we consider the problem of identifying the minimum knockout set that also minimizes the number of non-T nodes disconnected from S to be an important extension to this problem.

Heuristic performance. We implemented the heuristic shown in Figure 5 in Java. A set of 100 iterations of the algorithm took approximately one second to complete on a Apple 1.33 GHz G4 laptop running Mac OS X.

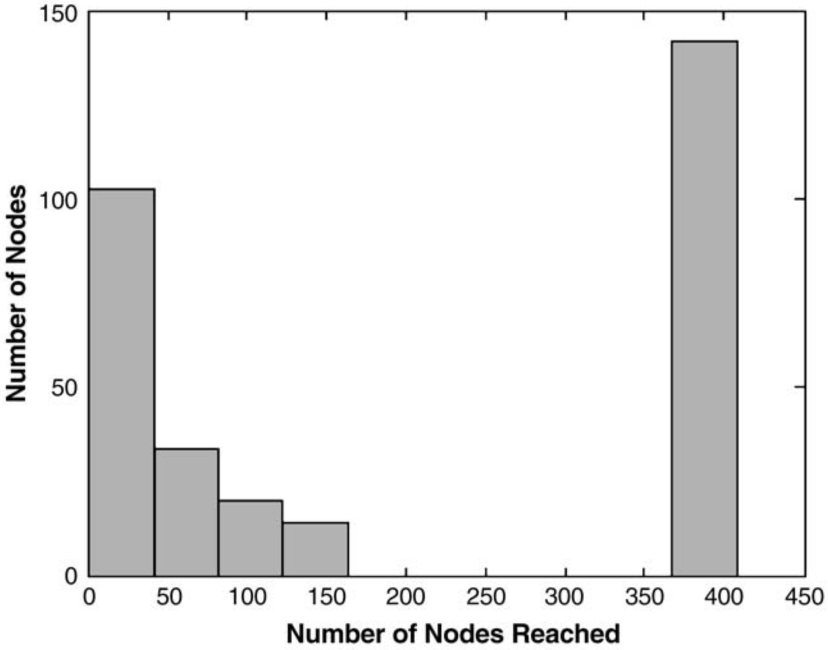

FIG. 6. The distribution of nodes in the EGFR pathway graph by the total number of nodes in the pathway graph they can reach.

A Java implementation of the model and algorithms described in this paper is available for download. The tool can load networks stored in the SBML format, allowing biologists to import networks designed in CellDesigner and other biological network editors (Funahashi et al., 2003).

6. CONCLUSION

In this paper we have presented a formal graph model, inspired by PetriNet models, that permits the use of graph theory to reason about the properties of biological networks. In addition, we have characterized two important research questions pertaining to biological networks, formulated them on our model, and provided efficient and accurate algorithms for solving them. To our knowledge, this is the first paper to formally define and propose a computational solution to the Minimum Knockout Problem. Despite being NP-Hard, our heuristic shows excellent performance on a large and important network in the research community.

Moving forward, we recognize that a useful addition to the current heuristic for the Minimum Knockout Problem is the ability to return a set of minimum knockout sets rather than just a single one. Furthermore, we intend to consider additional biological constraints, such as selecting the minimum knockout set with the least impact to global connectivity of the graph. There is also work to be done in studying other existing and new problems under the pathway graph model.

REFERENCES

Aladjem, M.I., Pasa, S., Parodi, S., et al. 2004. Molecular interaction maps—a diagrammatic graphical language for bioregulatory networks. *Sci. STKE* pe8.

Barabasi, A.L., and Oltvai, Z.N. 2004. Network biology: understanding the cell’s functional organization. *Nat. Rev. Genet.* 5, 101–113.

Bhalla, U.S., and Iyengar, R. 1999. Emergent properties of networks of biological signaling pathways. *Science* 283, 381–387.

Bhalla, U.S., Ram, P.T., and Iyengar, R. 2002. MAP kinase phosphatase as a locus of flexibility in a mitogen-activated protein kinase signaling network. *Science* 297, 1018–1023.

Chabrier-Rivier, N., and Fages, F. 2003. Symbolic model checking of biochemical networks. *Lect. Notes Comput. Sci.* 2602, 149–162.

Chabrier-Rivier, N., Chiaverini, M., Danos, V., et al. 2004. Modeling and querying biomolecular interaction networks. *Theoret. Comput. Sci.* 325, 25–44.

Chabrier-Rivier, N., Fages, F., and Soliman, S. 2005. The biochemical abstract machine (BIOCHAM). *Lect. Notes Comput. Sci.* 3082, 172–191.

Eker, S., Knapp, M., Laderoute, K., et al. 2002. Pathway logic: executable models of biological networks. Electronic Notes in *Theoret. Comput. Sci.* 71, *eker-etal-02wrla*.

Belloni, E., et al. 1996. Identification of Sonic hedgehog as a candidate gene responsible for holopro-sencephaly. *Nat. Genet.* 14, 353–356.

Feldman, D.S., Carnes, C.A., Abraham, W.T., et al. 2005. Mechanisms of disease: β-adrenergic receptors alterations in signal transduction and pharmacogenomics in heart failure. *Nat. Clin. Pract. Cardiovasc. Med.* 2, 475–483.

Fu, C., Qi, Z., and You, J. 2004. A BioAmbients based framework for chain-structured biomolecules modelling. *Lect. Notes Comput. Sci.* 3314, 455–459.

Funahashi, A., Morohashi, M., Kitano, H., et al. 2003. CellDesigner: a process diagram editor for gene-regulatory and biochemical networks. *BIOSILICO* 1, 159–162.

Hoffmann, A., Levchenko, A., Scott, M.L., et al. 2002. The IKappaB-NF-kappaB signaling module: temporal control and selective gene activation. *Science* 298, 1241–1245.

Hunter, T. 2000. Signaling—2000 and beyond. *Cell* 100, 113–127.

Karp, R.M. 1972. Reducibility among combinatorial problems. *In*: Miller, R.E., and Thatcher, J.W., eds., *Complexity of Computer Computations*, 85–103. Plenum Press, New York.

Kitano, H., Funahashi, A., Matsuoka, Y., et al. 2005. Using process diagrams for the graphical representation of biological networks. *Nat. Biotechnol.* 23, 961–966.

Ma'ayan, A., Jenkins, S.L, Neves, S., et al. 2005. Formation of regulatory patterns during signal propagation in a mammalian cellular network. *Science* 309, 1078–1083.

Meng, T.C., Somani, S., and Dhar, P. 2004. Modeling and simulation of biological systems with stochasticity. *Silico Biol.* 4, 293–309.

Menger, K. 1927. Zur allgemeinen Kurventheorie. *Fundament. Math.* 10, 96–115.

Nagasaki, M., Doi, A., Matsuno, H., et al. 2004. A versatile Petri Net based architecture for modeling and simulation of complex biological processes. *Genome Inform.* 15, 180–197.

Oda, K., Matsuoka, Y., Funahashi, A., et al. 2005. A comprehensive pathway map of epidermal growth factor signaling. *Mol. Syst. Biol.* msb41000014, E1–E17.

Peterson, J.L. 1981. *Petri Net Theory and the Modelling of Systems*. Prentice-Hall, Englewood Cliffs, NJ.

Phillips, A., and Cardelli, L. 2005. A correct abstract machine for the stochastic pi-calculus. Electronic Notes in *Trans. Comput. Syst. Biol.*

SBML. 2005. Systems biology markup language webpage. Available at: *http://www.sbml.org*. Accessed September 13, 2006.

Smith, A.E., Slepchenko, B.M., Schaff, J.C., et al. 2002. Systems analysis of Ran transport. *Science* 295, 488–491.

Sriram, M. 2003. Modelling protein functional domains in signal transduction using Maude. *Briefings Bioinform.* 4, 236–245.

Tran, N., Baral, C., Nagaraj, V., et al. 2005. Knowledge-based interacative framework for hypothesis formation in biochemical networks. *Lecture Notes in Computer Science* 3615, 121–136.

Tsavachidou, D., and Liebman, M. 2002. Modeling and simulation of pathways in menopause. *J. Am. Med. Inform. Assoc.* 9, 461–471.

Wuchty, S., Oltvai, Z.N., and Barabasi, A.L. 2003. Evolutionary conservations of motif constituents in the yeast protein interaction network. *Nat. Genet.* 35, 176–179.

Address reprint requests to:
Dr. Luay Nakhleh
Department of Computer Science
Rice University
6100 Main St., MS 132
Houston, TX 77005

E-mail: nakhleh@cs.rice.edu