Functional significance checking in noisy gene regulatory networks

S. Akshay[1], Sukanya Basu[1], Supratik Chakraborty[1], Rangapriya Sundararajan[1,2], and Prasanna Venkatraman[2,3]

[1] Department of CSE, IIT Bombay
[2] ACTREC, Kharghar
[3] Tata Memorial Hospital, HBNI

Abstract. Finding gene regulatory pathways that explain outcomes of wet-lab experiments is one of the holy grails of systems biology. SAT-solving techniques have been used in the past to find few small explanatory pathways assuming either zero or a few known perturbations in the experimental observations. Unfortunately, these approaches do not work when (i) there is noise in the experimental data or domain knowledge, as opposed to known perturbations, and (ii) the number of possible pathways generated by repeatedly invoking a SAT-solver is too large to be analyzed by enumeration. In such settings, determining if an actor plays a functionally significant role towards explaining experimental observations is very difficult using existing SAT-based techniques.

In this paper, we formalize the problem of functional significance checking in gene-regulatory pathways in the presence of a bounded amount of noise. We show that this problem is Δ_2^P-hard and hence cannot be efficiently encoded into SAT (unless the polynomial hierarchy collapses). We then propose an algorithm that uses a polynomial number of SAT-oracle invocations to solve a practically useful version of this problem. Finally, we present results on checking functional significance of suspect genes in real microarray data obtained from cancer cell-line experiments, some of which are corroborated by subsequent wet-lab knock-off experiments.

1 Introduction

A central problem in systems biology concerns finding gene regulatory pathways that explain observed outcomes of wet-lab experiments. In a typical wet-lab experiment, a pre-determined stimulus is given to specially prepared cells under controlled conditions, and the expressions of various genes (i.e. concentrations of corresponding gene products) measured at carefully timed instants. Practical constraints (including cost, unknown time constants of biological processes etc.) often limit the number of gene expression profiles that can be measured during the course of an experiment. In addition, measured gene expression profiles almost inevitably have noise. As a consequence, it becomes difficult to infer if a suspected gene plays a functionally significant role in the outcome of the experiment. This motivates us to ask if we can computationally predict the functional significance of a gene even when a single noisy expression profile (in addition to a reference profile) is available, by taking into account domain knowledge about gene interactions from public-domain databases, and by bounding a quantitative metric of the admissible noise.

The gene expression profile (often measured using microarray [5] or RNA-sequencing [52]) is usually given as log fold changes relative to a reference profile corresponding to a normal (or wild-type) cell, and serves as a proxy for the activation level of a gene. An activated (resp. inhibited) gene in the experimental cells usually yields higher (resp. lower) concentrations of the corresponding gene product compared to a normal cell. The use of contextual gene interaction information from a public-domain database like KEGG [23] provides a reasonable encoding of domain knowledge. "Noise" in our setting can be along two dimensions: (a) some gene expression measurements can be erroneous, (b) interactions between gene pairs in the context of the experiment under study may differ from what is recorded in KEGG, giving rise to "noise" in gene interaction information. Given these noisy inputs, we wish to identify if a suspect gene plays a functionally significant role in the outcome of the wet-lab experiment. Informally, this happens if the presence of the gene makes it possible to "easily" explain the measured gene expression profile consistently with domain knowledge, while its absence makes it difficult to provide any such explanation. We quantify the "easiness" via a quantitative metric, which we formalize as the number of relaxations or changes that must be admitted in the input to obtain an explanation.

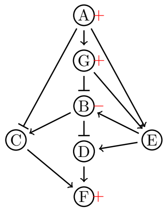


Fig. 1: Example gene interactions

An illustrative example: To better understand the computational aspects of the problem, consider a hypothetical wet-lab experiment in which cancer cells are treated with a drug known to activate gene A. Suppose we wish to determine how this affects the activation of another gene F in the cancerous cells. For simplicity, assume that only 7 genes named A, B, C, D, E, F and G potentially play any role in the outcome of the experiment. Let the gene expression profile obtained at an appropriate time instant be as follows: A, F, G over-expressed, B under-expressed, C, D, E did not show any significant difference in expressions relative to that of a normal (non-cancerous) cell. From this, we infer that A, F and G are activated and B is inhibited in the context of the experiment. Genes C, D and E in the cancerous cells could either be in their respective ground states (as in a normal cell), or could even be in mildly activated or mildly inhibited states (mild enough so that they do not express their effect overwhelmingly in the gene expression profile). Suppose we are also told that the domain knowledge about mutual interaction of genes A through G are as in the graph shown in Fig. 1 (sans the ± labelings). In this figure, a $\rightarrow$ denotes an activating interaction ($A \rightarrow E$ implies that if A is active, so must E be) and a $\dashv$ denotes an inhibiting edge ($G \dashv B$ implies that if G is active, B must be inactive). Since the ground state (in a normal cell) of a gene may itself be activated/inhibited, we must be careful in interpreting the $\rightarrow$ and $\dashv$ edges. For example, the edge $A \rightarrow E$ not only admits both A and E being activated, but also admits both being inhibited. To see why this makes sense, note that if E has an inhibited ground state, then an inhibited A cannot activate E through an $A \rightarrow E$ edge. Similarly, $G \dashv B$ not only admits G activated and B inhibited, but also vice versa, i.e. B has an activated ground state, and G being inhibited, cannot inhibit B.

Given domain knowledge encoded in a graph like Fig. 1, we represent activation levels of genes in the experiment under study by ± labelings of nodes, where activated genes are labeled "+" and inhibited genes are labeled "-". Our first goal is to determine if there exists a set of paths from A to F in Fig. 1, and a ± labeling of nodes along these paths, such that the labeling is consistent with both the observed gene expressions and the domain knowledge. Informally, such a set of paths "explains" the experimental observations consistently with domain knowledge. In this example, it is indeed possible to find such an explanation with three paths from A to F, namely: $A(+) \rightarrow G(+) \dashv B(\text{-}) \dashv D(+) \rightarrow F(+)$, $A(+) \rightarrow E(+) \rightarrow D(+) \rightarrow F(+)$ and $A(+) \rightarrow G(+) \rightarrow E(+) \rightarrow D(+) \rightarrow F(+)$ There are several points to note here: (i) although E was not differentially expressed in the observed profile, it is fine to assign label "+" to E in the explanation, since E could indeed have been in a mildly activated state that didn't result in a strong gene expression profile, (ii) although E and B are included in the explanation, the induced edge $E \rightarrow B$ is not included since the labelings of B, E are not consistent with $E \rightarrow B$, and (iii) the presence of a topological path from A to F through B doesn't necessarily imply that this path explains the experimental observations consistently with domain knowledge. For example, although there is a topological path $A(+) \rightarrow G(+) \dashv B(\text{-}) \rightarrow C(?) \rightarrow F(+)$ in Fig. 1, there is no way of assigning a label ("+" or "-") to C that is consistent with the interpretation of activating and inhibiting edges. Thus, finding explanations is significantly harder than finding topological paths or induced sub-graphs.

We now ask: *Does gene D play a functionally significant role in explaining the observed expressions consistently with domain knowledge?* While the precise notion of functional significance will be discussed later, informally, we ask if we can find a domain knowledge-consistent explanation of the observed gene expressions *even if node D is removed from Fig. 1*. It is easy to see from Fig. 1 that the answer is in the negative. In contrast, if node E or C (or both) is (are) removed from Fig. 1, the path $A(+) \rightarrow G(+) \dashv B(\text{-}) \dashv D(+) \rightarrow F(+)$ continues to explain the observed gene expressions. Therefore, if we assume that all gene expression measurements are noise-free, D is functionally significant, while E and C are not. However, if we admit that one gene expression measurement can be noisy, then functional significance of D warrants re-examination. Indeed, with D removed from Fig. 1, the paths $A(+) \rightarrow E(+) \rightarrow B(+) \rightarrow C(+) \rightarrow F(+)$ and $A(+) \rightarrow G(+) \rightarrow E(+) \rightarrow B(+) \rightarrow C(+) \rightarrow F(+)$ explain the observed gene expressions with the (noisy) label of B *changed* from "-" to "+". This shows that functional significance of a gene can vary depending on the admissible noise.

Generalizing from the above discussion, our objective is to study computational techniques that (i) work with a single gene expression profile (in addition to a reference profile), (ii) are tolerant to a bounded amount of noise in both gene expression measurements and in the encoding of domain knowledge as gene interactions, and (iii) allow us to check whether a suspect gene (provided as input) plays a functionally significant role in explaining observed gene expression levels. By bounded noise, we mean that the number of errors either in the gene expression measurements or difference wrt KEGG must be at most a fixed constant, which is typically small. Note that this does not mean we know the errors, just that their number is limited. This is a reasonable assumption since allowing an arbitrarily large amount of noise would invalidate the experiment and any inferences made from it entirely.

In this paper, we formalize the problem described above, show that it is Δ_2^P-hard and in Π_2^P as well as present an algorithm to solve a useful variant of the problem. We are not aware of any earlier proof of hardness of even the simplest problem of finding explanations in the absence of noise. We fill this gap and go much beyond to prove the Δ_2^P-hardness of functional significance checking with bounded weighted noise. This shows that functional significance checking cannot be reduced to propositional SAT-solving (unless the polynomial hierarchy collapses). Our treatment of noise is also more robust than that used in earlier work. Specifically, we allow different genes and gene interactions to contribute in a weighted manner to the overall noise metric. Additionally, we don't need the user to specify the exact set of gene expressions or gene interactions that may be noisy. Instead, we allow all combinations of noisy gene expressions and gene interactions subject to the weighted noise metric staying within specified bounds. This permits exploring a much larger space of possible explanations than that in earlier work (viz. [10]). Finally, our algorithm detects functional significance of a gene without actually enumerating the potentially explosively many explanations of the observed gene expressions while admitting bounded noise. This makes it possible to analyze much larger systems of gene interactions.

The entire work reported in this paper was done by a team of three computer scientists and two molecular biologists. As such, the biological relevance of modeling artifacts and predictions were discussed and validated at each step. However, this paper is focused more on the computational aspects.

Related Work Biological phenomena have been modeled in various ways, viz. using Petri-nets, ODEs, sets of rules, Boolean networks, etc (see [20, 51]). A popular way of representing biological networks, especially gene regulatory networks, is *influence graphs* [42], which are (partially) edge-labeled graphs, used to model incomplete data. The Sign Consistency Model (SCM) of [41] enhances this with a (partial) labeling of nodes such that the whole labeled graph is consistent with a set of constraints [19]. In [18], a SAT-solver and MAX-SAT solver are used to check for consistency, somewhat similar to our work. Answer set programming (ASP) is yet another technique for obtaining models to a set of logical constraints used in AI [43], for searching models of NP-hard problems [30] and to detect inconsistencies, repair and prediction in biological networks [11, 12]. The works in [33, 46] model different notions of sign consistency and use an ILP solver to obtain a minimal set of nodes whose sign needs to be changed to be consistent. Such variants can also be encoded in our approach.

Several tools have been built over the years to analyze biological pathway networks [4, 8, 15, 16, 44, 50]. Of these [15, 16] apply statistical methods to correlate the network topology and gene expression data, which allows them to also identify some functional associations, assuming the availability of sufficient gene interaction data. The approaches in [6, 9, 31, 38, 53] try to find enriched pathways based only on gene information, whereas [2, 7, 29, 32, 48, 54] use both gene-expression and topology information for selecting candidate enriched pathways. In spite of the apparent difficulty, some tools such as [13,21,35,45,49] have tried to exploit the full annotation on the interactions, while recently [24,27,28] have stringently analyzed relations between genes. In [14], SMT-solvers have been used to analyze robustness under mutations of gene regulatory networks.

While encoding the problem of finding an explanation from known pathways and expression data as a SAT problem has similarities with other work in literature [10,40], such an encoding often gives no explanation subgraphs or too many of them as solutions to the SAT problem. Crucially it doesn't solve functional significance checking when expression data and knowledge about pathways are noisy, unless we examine every explanation subgraph for all noisy inputs – an impractical task. The primary differentiator of our work vis-a-vis these earlier work is in the way we model noise and *implicitly* consider all possible noisy inputs subject to the weighted noise being bounded, while still requiring a polynomial number of SAT invocations.

Finally, identifying important actors in a network has been studied in multiple contexts, including the web, social media networks, gene regulatory and protein-protein interaction networks. Various graph theoretic metrics have been used to detect crosstalk and identify *hubs* and *bottlenecks* in large biological networks [37, 55]. Our work can be used in tandem with these techniques by first obtaining potential candidates using graph theoretic techniques, and then checking their functional significance using our approach.

2 Problem formulation

While we have used KEGG [23] to encode domain knowledge of gene interactions in our experiments, our abstract problem formulation is not KEGG specific. To keep the exposition simple, we assume that there are only two types of edges – activating ($\mathcal{A}$) and inhibiting ($\mathcal{I}$). The domain knowledge of gene interactions is given as an edge labeled graph $G_{dom} = (V, E, \mu)$, where V is the set of genes, $E \subseteq V \times V$ is the set of interactions (directed edges) between genes, and $\mu : E \longrightarrow L_e$ is a labeling of edges with $L_e = \{\mathcal{A}, \mathcal{I}\}$. The interpretation of activating and inhibiting edges is as follows: For an edge $e = (u, v)$, if $\mu(e) = \mathcal{A}$, then gene v must be activated whenever u is active. In addition, as discussed in Section 1, an activating edge (u, v) is consistent with both u and v being in inhibited states. Similarly, if $\mu(e) = \mathcal{I}$, v must be inhibited whenever u is active. In addition, an inhibiting edge (u, v) is consistent with u being inhibited and v being active.

In order to represent the gene expression profile, we decorate each node v in the graph G_{dom} with a label $\lambda(v)$ from the set $L_v = \{+, -, ?\}$. Here, $+$ denotes an over-expressed (and by implication, active) gene, $-$ denotes an under-expressed (and by implication, inhibited) gene, and ? denotes a gene that is not significantly differentially expressed with respect to the expression level of a normal cell. For clarity of exposition, we use $*$ to denote either $+$ or $-$, but not both.

The domain knowledge and gene expression profile can be represented together as a node- and edge-labeled graph $G = (V, E, \lambda, \mu)$, where $\lambda : V \to L_v$ and $\mu : E \to \{\mathcal{A}, \mathcal{I}\}$. We also assume that we are given three nodes $s, t, i \in V$ as follows: s represents a stimulus gene, the effect of whose activation we wish to study, t represents a target gene that is eventually activated (possibly after a long chain of interactions) due to activation of s, and i represents a suspect gene whose functional significance in the activation of t by s is the subject of our investigation. For simplicity, we fix $\lambda(s) = \lambda(t) = +$; other combinations of $\lambda(s)$ and $\lambda(t)$ are easily handled. To formally define the notion of functional significance, we first define an *explanation subgraph*. Informally, this is a subgraph of G that contains s-t paths along with a labeling of nodes that "explains" the observed gene expression profile while being consistent with the domain knowledge.

Definition 1 (Explanation subgraph). *Let $G = (V, E, \lambda, \mu)$ be as defined above, and let s and t be nodes in V s.t. $\lambda(s) = \lambda(t) = +$. An* explanation subgraph *of (G, s, t) is a node- and edge-labeled graph $G' = (V', E', \lambda', \mu')$ s.t.,*

1. ***Subgraph containing** s, t: We require $V' \subseteq V$, $E' \subseteq E \cap (V' \times V')$, μ' is the restriction of μ to E', and $s, t \in V'$.*
2. ***Labels consistent with observed expressions**: $\lambda'(v) \in \{+, -\}$ for all $v \in V'$, and $\lambda'(v) = \lambda(v)$ if $\lambda(v) \neq ?$.*
3. ***No floating nodes:** Every $v \in V'$ is reachable from s in G'.*
4. ***Activity condition**: Every s-t path of length > 1 in G' passes through some node $v \notin \{s, t\}$ with $\lambda(v) = +$, and every such node v in G' appears on some s-t path in G'. Effectively, for a pathway to credibly explain how s eventually activates t, it must be supported by at least one other active node along the pathway. Also, every node in G' that was originally active must contribute towards explaining how s activates t along some path in G'.*
5. ***Compatible labeling**: For every edge $e = (u, v)$ in E', if $\mu'(e) = \mathcal{A}$, then $\lambda'(u) = \lambda'(v)$, and if $\mu'(e) = \mathcal{I}$, then $\lambda'(u) \neq \lambda'(v)$. Moreover, every node other than s in G' must have at least one incoming compatible edge.*

For the example in Fig. 1, the path $A(+) \to E(+) \to D(+) \to F(+)$ doesn't constitute an explanation subgraph because the activity condition is violated. However, $A(+) \to G(+) \dashv B$(-) $\dashv D(+) \to F(+)$ is an explanation subgraph.

2.1 Graph relaxation: modeling errors and noise

A startling finding of our initial experiments with real micro-array data and KEGG pathways was that often no explanation subgraphs could be found at all. Delving deeper, we realized that there were two primary reasons for this: (a) the pathway information in KEGG didn't relate to the context in which the experiments were performed (i.e., some edge attributes were incorrectly labeled), and (b) there was noise in the micro-array data (i.e., node attributes were incorrectly labeled). Thus we need to search for explanation subgraphs not on the original graphs, but graphs obtained by changing some (unknown) edges and nodes. To formalize this, we introduce the notion of *relaxations*. Specifically, we associate an integer relaxation weight to each node and edge, and allow node and edge labels to be changed when finding an explanation subgraph. The total *node noise* (resp. *edge noise*) introduced to obtain an explanation subgraph is simply the sum of relaxation weights of all nodes (resp. edges) whose labels had to be ignored or changed to obtain the explanation subgraph. We bound the admissible noise by specifying an upper bound (n, e) of node and edge noise respectively. For notational convenience, we refer to (n, e) as *relaxation bounds* in the subsequent discussion. Thus, these bounds provide a quantitative metric to deal with noise (caused by errors or inconsistencies in KEGG and microarray data), as mentioned in the introduction. Our definition of noise is driven by specific biological experiments and hypotheses as explained in Section 5. However, our techniques and encoding can also model other related notions considered in the literature, such as creation of new edges.

Formally, for a subgraph G' of G, a node in G is said to be *relaxed* in G' if one of the following hold: (i) it is labeled $+$ in G, but is absent in G', i.e. a node active in G is excluded from G', (ii) it is labeled $+$ (resp. $-$) in G, and is present but labeled $-$ (resp. $+$) in G'. If a node is inhibited in G but excluded from G', we do not treat it as relaxed. Similarly, edge $e = (u, v)$ in G is *relaxed* in G' if $u, v \in V', e \in E'$ and either $\mu(e) = \mathcal{I}$ and $\mu'(e) = \mathcal{A}$ or $\mu(e) = \mathcal{A}$ and $\mu'(e) = \mathcal{I}$.

Definition 2 (Relaxed explanation). *Given $G = (V, E, \lambda, \mu)$, source s, target t, a relaxation weight $R : V \cup E \to \mathbb{N}$ and $(n, e) \in \mathbb{N}^2$, we call H an (n, e)-*relaxed explanation *of (G, s, t) under R if (a) there exists a subgraph G' of G obtained by relaxing nodes and/or edges, (b) $\sum_{\{v \in V | v \text{ relaxed in } G'\}} R(v) \leq n$, (c) $\sum_{\{e \in E | e \text{ relaxed in } G'\}} R(e) \leq e$ (d) H is an explanation subgraph of (G', s, t).*

2.2 Pareto optimality and functional significance

As mentioned earlier, often there are no explanation subgraphs with 0 node and edge relaxations. Interestingly, our experiments indicate that there is a large multiplicity (literally 1000s) of explanation subgraphs if we allow small node and/or edge relaxations. In this context, solutions obtained with very large values of node and/or edge relaxations may not be meaningful. Relaxing too many nodes allows activation status of many nodes to differ from the observed gene expression profile. Similarly, relaxing too many edges amounts to making significant modifications to a curated database of regulatory pathways. None of these are desirable. Indeed, if we allow all nodes or all edges to be relaxed, we can always find an explanation subgraph that may hardly relate to the wet-lab experiment under investigation. Hence it makes sense to ask for minimal node and edge relaxations that yield at least one explanation subgraph. Not surprisingly, increasing node relaxations reduces the requirement of edge relaxations, and vice versa. Therefore, we have a multi-objective optimization problem and obtain a set of minimal or Pareto-optimal (n, e) values. Further, since large node and edge relaxations are undesirable, we want explanations where node and edge relaxations are within given bounds. This motivates us to define a *window of relaxation* W as a pair of intervals, $\langle [n_l, n_u], [e_l, e_u] \rangle$, for node and edge relaxations respectively. We say that $(n, e) \in W$ iff $n \in [n_l, n_u]$ and $e \in [e_l, e_u]$.

Consider the partial order $\sqsubseteq$ on $\mathbb{N} \times \mathbb{N}$ defined by $(n', e') \sqsubseteq (n, e)$ iff $n' \leq n$ and $e' \leq e$. We say (n, e) *dominates* (n', e') if $(n', e') \sqsubseteq (n, e)$, and that (n, e) *strictly dominates* (n', e') if $(n', e') \sqsubseteq (n, e)$ but $(n', e') \neq (n, e)$. Given an input instance (G, s, t, W, R), where G, s, t, are as before, R is a relaxation weight function and W a relaxation window, let $Sol(G, s, t, W, R)$ denote the set of $(n, e) \in W$ such that there exists an (n, e)-relaxed explanation of (G, s, t) under R. If $(n, e) \in Sol(G, s, t, W, R)$ but both $(n-1, e)$ and $(n, e-1)$ are not in $Sol(G, s, t, W, R)$, we say (n, e) is on the *solution curve* of (G, s, t, W, R). The set of points on the solution curve forms a Pareto-optimal curve; any point in W that dominates a point on the curve is in $Sol(G, s, t, W, R)$ and any point in W that is strictly dominated by a point on the curve is not in $Sol(G, s, t, W, R)$.

We now make two reasonable, yet important, assumptions.

A1: The "golden truth" pathway for the wet-lab experiment under study, henceforth called *true explanation subgraph*, is present, modulo relaxations and inter-pathway crosstalk, in the input graph $G = (V, E, \lambda, \mu)$.

A2: The true explanation subgraph corresponds to a Pareto-optimal point $(n^\star, e^\star)$ in the relaxation window W of interest for the given relaxation weight function R. It is reasonable to expect $(n^\star, e^\star)$ to be a Pareto-optimal point, as otherwise, we'd have an alternative explanation of the microarray data with fewer relaxations than that required for the true explanation subgraph to provide a plausible explanation.

Definition 3. *Under assumptions A1 and A2, a node v is said to be* functionally significant *in (G, s, t, W, R) if its removal from G leaves no $(n^\star, e^\star)$-relaxed explanation subgraph. In other words, $Sol(G \setminus \{v\}, s, t, \langle [n^\star, n^\star], [e^\star, e^\star] \rangle, R) = \emptyset$.*

Unfortunately, Defn 3 does not yield a practical algorithm for checking functional significance of a node, due to two reasons. First, we do not know the values of $n^\star$ and $e^\star$ for a given experiment. Second, our studies show that there are literally thousands of explanation subgraphs at each Pareto-optimal point in the window of relaxation of interest. So, even if we knew $(n^\star, e^\star)$, it would be practically impossible to examine all $(n^\star, e^\star)$-relaxed explanation subgraphs and identify a common node. Thus, we must find a way to decide the functional significance of a node without knowing $(n^\star, e^\star)$ exactly, and without generating all explanation subgraphs corresponding to Pareto-optimal pairs. The following lemma provides a sufficient condition to surmount the above hurdles.

Lemma 1. *Suppose $Sol(G, s, t, W, R) \neq \emptyset$ and either $Sol(G \setminus \{i\}, s, t, W, R) = \emptyset$ or for every $(n, e) \in Sol(G \setminus \{i\}, s, t, W, R)$, there exists $(n', e') \in Sol(G, s, t, W, R)$ such that (n, e) strictly dominates (n', e'). Then i is functionally significant in (G, s, t, W, R) under assumptions A1 and A2.*

3 Complexity results

Theorem 1. *Checking the existence of an explanation subgraph, even without relaxations, is* NP*-complete.*

Proof. It is easy to see that the problem is in NP, since we can guess the explanation subgraph, and check it in polynomial time. To prove NP-hardness, we reduce 3-SAT to our problem. Let φ be an instance of 3-SAT in CNF, with ℓ variables $x_1, \ldots, x_\ell$ and m clauses.

For each variable x_i, we first construct a gadget A_i of 6 nodes depicted in Figure 2, three for variable x_i (which we call a_i, b_i, d_i) and three for $\neg x_i$ (which we denote na_i, nb_i, nd_i). We add activating edges from source s to a_i and na_i for all i. Also add 4 activating edges from a_i to b_i, b_i to d_i, na_i to nb_i and nb_i to d_i and 4 inhibiting edges from a_i to nb_i, na_i to b_i, b_i to nd_i and nb_i to nd_i. Finally, we add node label + for d_i and nd_i and activating edges from both to target t. This gadget A_i ensures that x_i and its negation are not active at the same time.

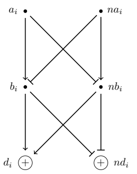


Fig. 2: gadget A_i

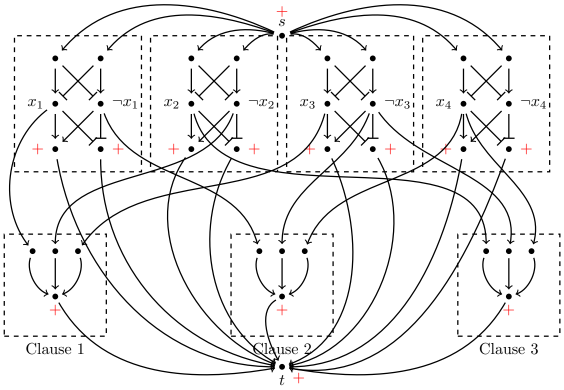


Fig. 3: Construction for reduction from 3-SAT

For each clause c, we construct gadget B_c with 4 nodes: one for each of the 3 literals that occurs in the clause, denoted l_i^c (i.e., $l_i^c = x_i$ or $\neg x_i$) and an additional node l_c^c. Now if $l_i^c = x_i$, (resp $= \neg x_i$) then we add an edge from b_i in gadget A_i to it, else we add an edge from nb_i. Further we add edges from each literal in a clause c to the additional node l_c^c. Each of these additional nodes l_c^c are labeled +, which is used to ensure that each clause does evaluate to true in a valid explanation. Each clause of the original instance is replicated by edges from a variable or its negation as appropriate, which finally converge at l_c^c emulating the disjunctions within each clause. Finally from each l_c^c node we add an edge to the target node t. Recall that the target is also labeled +.

We claim that an explanation subgraph from s to t exists iff the formula is satisfiable. In one direction, if there is an explanation from s to t, the additional node l_c^c at clause c for every clause must be active and each variable is assigned a unique value. Further, to make this active, by the compatibility condition, one of the literals in that clause gadget must be active. In turn to make that literal active, node corresponding to the literal should be active and this gives the satisfying assignment. Conversely, if the formula is satisfiable, then the satisfying assignment defines an explanation subgraph. This completes the proof of NP-hardness. An example is the formula $\varphi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4 \vee \neg x_1) \wedge (x_2 \vee \neg x_3 \vee x_4)$, whose graph is shown in Fig 3 with a source s and target t. □

Depending on whether the relaxation window W is fixed or part of the input to a decision procedure, we obtain the following results.

Theorem 2. *For every $(n, e) \in \mathbb{N}^2$, for every relaxation weight function R,*

1. *Checking for an (n, e)-relaxed explanation subgraph under R is NP-complete.*
2. *Checking functional significance of a node in (G, s, t, W, R), where $W = \langle [0, n], [0, e] \rangle$ is co-NP complete.*

Proof. Part 1. follows from proof of Theorem 1 with a simple modification: we replicate the gadget for each variable and clause $n + e + 1$ times, so that even if n nodes and e edges are relaxed, finding an explanation subgraph would require setting each variable in way that all clauses are satisfied. For Part 2., we modify the construction in Theorem 1 by adding a special node n_i, where i is the node whose significance we wish to check. We add an edge from the source s to n_i and from n_i to each node l_c^c for each clause c and to nodes d_i and nd_i for each x_i. In the resulting graph G', there is a path from s to n_i to each of l_c^c (for each clause c), d_i, nd_i and then to t. Thus, with no relaxations, we can find an explanation. However, if n_i is removed, then there is an explanation with no relaxations iff φ is satisfiable. In other words the solution curve shifts, i.e., i is functionally significant in G iff φ is unsatisfiable.

Theorem 3. *If the relaxation window is part of the input, functional significance checking is Δ_2^P-hard and is contained in Π_2^P.*

Proof. For the hardness, we show a reduction from the following Δ_2^P-complete problem [25]: Given a satisfiable CNF formula φ and a linear ordering of $x_1 \prec \ldots x_n$ in φ, does the lexicographically largest satisfying assignment of φ have its least significant bit $x_1 = 1$? To reduce this to functional significance checking, consider the construction in the proof of NP-hardness above, but with the following modification. The gadget in Fig. 2 is modified so that nodes a_i, na_i are removed and so are all edges coming in and out of them. We add an inhibiting edge from the source s to each nb_i, and an inhibiting edge from each nb_i to the corresponding node b_i. Let G be the resulting graph. Let R be the relaxation weight function that assigns weight 2^{i-1} to the inhibiting edge from s to nb_i and 2^n to all other edges. R also assigns weight 2^n to all nodes. We ask if b_1 is functionally significant in (G, s, t, W, R), where $W = \langle [0, 0], [0, 2^n - 1] \rangle$. The size of (G, s, t, W, R) is polynomial in $|\varphi|$. Also, the choice of W disallows relaxation of any node and any edge other than those from s to some nb_i. The lexicographically largest satisfying assignment of φ corresponds to an explanation graph with the smallest edge relaxation noise. If this explanation includes b_1, then removing b_1 from G disallows this explanation. This proves Δ_2^P hardness of functional significance checking.

Containment in Π_2^P is easy to see. We encode the problem as: for all (n', e')-relaxed solutions without the actor, there is an (n, e)-solution with the actor, where (n', e') strictly dominates (n, e) and both are within relaxation bounds. Since n, e, n', e' are integers within given relaxation bounds, the quantifier free part has a polynomial sized propositional encoding.

The problem of *counting explanation subgraphs* corresponds to #SAT, which is widely believed to be beyond the polynomial hierarchy (by Toda's theorem [47]). Thus, unless long-standing complexity-theory conjectures are falsified, checking functional significance (in Π_2^P) is easier than counting explanations.

4 SAT encoding and Pareto-curve generation

Given a problem instance (G, s, t, W, R), and a path length bound Δ, we first extract a sub-graph $\hat{G} = (\hat{V}, \hat{E}, \hat{\lambda}, \hat{\mu})$ of G that contains every simple path of length $\leq \Delta$ from s to t in G. This can be done easily using a forward and backward bounded search. Once $\hat{V}$ is defined, $\hat{E}$, $\hat{\lambda}$ and $\hat{\mu}$ are obtained by restricting E, λ and μ respectively to $\hat{V}$ and $\hat{E}$. In practice, Δ is chosen based on domain expert inputs, such that all potentially important s-t paths are included. Henceforth, whenever we refer a labeled graph G, we mean the pruned graph $\hat{G}$ for a value of Δ that is assumed to be constant.

The problem of deciding whether an (n, e)-relaxed explanation subgraph exists was shown to be NP-complete in Section 3. A SAT encoding of the problem is rather straightforward. Given a labeled graph $G = (V, E, \lambda, \mu)$, nodes $s, t \in V$, a relaxation weight function R, and a relaxation window $W = \langle [0, n], [0, e] \rangle$, we construct a propositional formula $\varphi_{G,s,t,W,R}$ that is satisfiable iff there is an (n, e)-relaxed explanation subgraph of (G, s, t) under R. The formula $\varphi_{G,s,t,W,R}$ has seven sub-formulas: (i) φ_{conn} encoding topological connectivity between nodes in the explanation subgraph (this uses the fact that all paths are of length $\leq \Delta$), (ii) φ_{data} encoding the labeling of nodes obtained from microarray data, (iii) φ_{act} encoding the activity condition in Defn 1, (iv) φ_{comp} encoding the compatibility condition in Defn 1, (v) φ_{rel} encoding that total node relaxation is $\leq n$ and total edge relaxation is $\leq e$, and (vii) φ_{imp} encoding that every node is reachable from s by a path of length at most Δ. These sub-formulas use a set of variables as described below. For each $v \in V$, we use 3 boolean variables, p_v, a_v and r_v, that encode whether v is present, active and relaxed respectively, in the explanation subgraph. Similarly, for each edge $e \in E$, we use 3 boolean variables, p_e, r_e and f_e that encode whether e is present, relaxed and contributes to the activity condition in Defn 1 respectively, in the explanation subgraph. Finally, for each $v \in V$, we use $\log \Delta + 1$ propositional variables $d_{v,0}, \ldots d_{v,\log \Delta}$ to encode a measure of "distance" from source s to v in the explanation subgraph.

Once $\varphi_{G,s,t,W,R}$ is obtained, a SAT solver (Z3 [34] in our case) can be used to obtain an (n, e)-relaxed explanation subgraph. We exploit the observation that satisfiability of $\varphi_{G,s,t,W,R}$ implies satisfiability of $\varphi_{G,s,t,W',R}$ where $W' = \langle [0, n'], [0, e'] \rangle$ and $(n, e) \sqsubseteq (n', e')$. Therefore, given any set of (n, e) pairs linearly ordered w.r.t. $\sqsubseteq$, we can use binary search to determine the smallest (under $\sqsubseteq$) pair (n, e) for which $\varphi_{G,s,t,W,R}$ is satisfiable. This suggests the following simple algorithm for constructing the Pareto-optimal curve. We first use binary search along the (n_l, e_l) to (n_u, e_u) diagonal of the window $W = \langle [n_l, n_u], [e_l, e_u] \rangle$ to find the smallest (under $\sqsubseteq$) pair (n_d, e_d) for which $\varphi_{G,s,t,W_d,R}$ is satisfiable, where $W_d = \langle [n_l, n_d], [e_l, e_d] \rangle$. Note that (n_d, e_d) may not be a Pareto-optimal point. We then use binary search on (n, e) pairs in $\langle [n_d, e_l], [n_d, e_d] \rangle$ and $\langle [n_l, e_d], [n_d, e_d] \rangle$ to find the projections of (n_d, e_d) on the Pareto-optimal curve. Once a Pareto-optimal point (n_p, e_p) is obtained, the problem can be recursively decomposed into those of generating Pareto-optimal curves in the relaxation windows $\langle [n_p, n_u], [e_l, e_p] \rangle$ and $\langle [n_l, n_p], [e_p, e_u] \rangle$. This requires a total of $\mathcal{O}(k \log_2 k)$ invocations of a SAT solver, where $k = \max(n, e)$, and gives us the Pareto curves, from which we can determine functional significance.

Note that our methodology is not contingent on a specific choice of relaxation, but implicitly considers all relaxations within given bounds. However, our tool also has the functionality of printing a set of relaxations used to obtain explanation subgraphs, if the user so desires.

5 Experimental results and a case-study

We began by constructing a database of existing pathways, by merging the 163 pathways from the KEGG database [22, 23], giving a master network of 2498 nodes and 10497 edges. In discussion with molecular biologists, we then fixed the gene expression data from a specific microarray experiment, with the following features: (i) the source, target and the differentially expressed nodes were not merged with any other id, (ii) if a gene occurred more than once in the expression data, we took the average of the fold-change for more than one occurrence of a gene, (iii) after considering realistic lengths of regulatory chains in the biological context, the path bound ($\varDelta$ in Section 4) was chosen to be 7. This resulted in a pruned subgraph with 297 nodes and 1858 edges. Of these nodes, 55 are up-regulated and 26 are down-regulated, as per the microarray data (see [1] for details). Finally, we also fixed an upper bound on number of relaxations that we allow among the nodes and edges in the worst case, i.e., the window size, denoted below as W to be at most 30×30. Note that this does not mean that we cannot have fewer perturbations, just that more than 30 errors (of either nodes or edges) were considered impractical. While we fix all the above parameters to be able to present results, we emphasize that these are easily tunable by the user. In our experiments, the relaxation weight function R assigned weight 1 to all nodes and edges. But the formulation allows generalizing to other weight functions, e.g., to not relax a node or edge, it suffices to assign a large weight to that node/edge.

With this setup, we encoded finding a relaxed explanation graph, as discussed in Section 4, and considered different source and target pairs, as well as different candidate actors which were checked for functional significance.We computed the Pareto optimal curves with and without the actor to check functional significance of the actor. All experiments were performed on an Intel(R)-Core(TM)-i7-3770 CPU. It had 8 cores with clock speed 3.40 GHz and total of 32 GB RAM. The code used C++ API of Z3 version 4.7.1 on Ubuntu 18.04.

Source-Target pair (Expt condition)	Func. Sign. Cand.	Pareto shift (Y/N)	# SAT Calls	Time (in hrs)
Synthetic1-5var-$W(5,5)$	x	Y	5	.035
Synthetic2-15var-$W(5,5)$	x	Y	6	.35
Synthetic3-45var-$W(0,0)$	x	Y	2	.004
TNFa-IkBa (Expr/Act merged)	None	-	62	5
TNFa-IkBa (Expr/Act merged)	p38	Y	72	5
TNFa-IkBa (Expr/Act merged)	ERK	N	62	2.6
TNFa-IkBa (Expr/Act merged)	PIK3CA	Y	71	1.5
TNFa-IkBa (Expr/Act merged)	AKT	Y	42	11
TNFa-IkBa (Expr only)	None	-	63	9
TNFa-IkBa (Expr only)	p38	Y	63	15
TNFa-IkBa (Expr only)	ERK	Y	63	15
TNFa-IkBa (Expr only)	PIK3CA	N	68	14
TNFa-IkBa (Expr only)	AKT	N	68	18.4
TNFa-IkBa (Act only)	None	-	64	15.6
TNFa-IkBa (Act only)	p38	Y	64	37
TNFa-IkBa (Act only)	ERK	N	64	25.8
TNFa-IkBa (Act only)	PIK3CA	Y	64	18.5
TNFa-IkBa (Act only)	AKT	Y	54	44
TNFa-A20	None	-	56	0.3
TNFa-A20	ERK	Y	57	0.7
TNFa-A20	AKT	N	52	0.3
TNFa-A20	p38	N	54	0.3

Table 1: Shift of Pareto curves

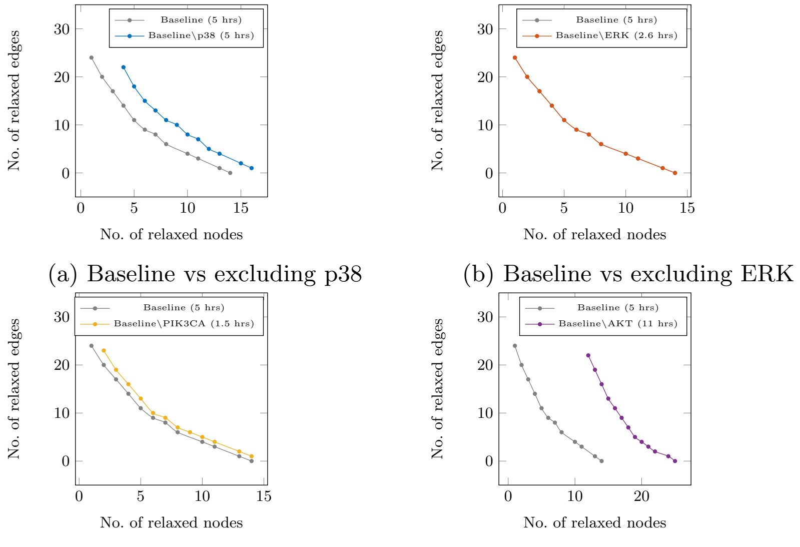
One way to understand the explanations is to enumerate and exhaustively look at each solution. However, with window size 30 × 30, there are 900 points, of which all points on or above the PO curve have multiple solutions. In our case, we found that for all such points there were at least > 1000 solutions per point. And enumerating these, and printing the solutions for just 30 of them (for inspection), for a single PO curve took over 100 hours of computations time. Thus, examining all solutions even at each point on the Pareto-optimal curve (to identify key players in the solution) is already prohibitively expensive. This leads us to use the shift of the Pareto-optimality curves to identify key players in context of an experiment. In Table 1, we present the results for a few different source-target pairs, different candidate actors and whether a shift was observed in the Pareto-curves or not, along with the time taken to plot these curves. The Pareto-optimality curves themselves, along with further experiments with more source-target pairs including ITGB1-ACTB, ITGB1-STAT3 are in [1]. We also performed experiments on synthetically constructed benchmarks motivated by Proof of Theorem 2. The benchmarks were parametrized by number of variables (in the 3SAT problem), and node, edge relaxation upper bounds, and a special node x that was made functionally significant. A select few results are in Table 1, with more in [1]. Interestingly, almost the entire time taken by our tool went into SAT solving using a state-of-the-art solver (Z3). Our tool minimizes the number of SAT calls as described in Section 4. The scalability of our approach hence crucially depends on the performance of the SAT solver, and is expected to improve with further improvements in SAT solvers.

In Table 1, Act/Expr merged means we included both types of edges in our potential explanation. However, we also experimented by (i) asking for the target IkBa to be expressed, and not just activated (by required the solution to have at least one expression edge reaching the target) and (ii) asking target IkBa to be activated (by requiring the solution to have at least one activation edge reaching the target), which led to surprisingly different Pareto-shifts.

Case-study: role of ERK,A20 in PSMD9-induced inhibition of NFkB: We performed a detailed case-study on a mammalian cell line model system, created as part of a joint project with researchers from a Cancer research institute: these were the embryonic human kidney cell lines called HEK 293 cell that stably over express PSMD9 (an important gene associated with radio resistance in breast cancer and glioblastoma [26, 36].) and obtained differential gene expression data specific to PSMD9. Among the many signalling events that could possibly be modulated by PSMD9, we were interested in finding key players that regulated the expression of IkBa, for a very specific reason. IkBa is a potent inhibitor of NFkB a transcription factor induced upon chemo and radiation therapy in cancer treatment [3, 17].

One of the mechanisms by which PSMD9 may achieve this is by inducing NFkB activation [39]. However, besides the reported mechanism, there are a number of other ways in which the activity of this gene can be modulated and this can vary depending on the context. Several kinases and transcription factors are involved and inflammatory cytokines such as TNFa can modulate activity of these players. NFkB is also under a remarkable tight feed-back loop involving both positive and negative regulators that are transcribed by NFkB and other TFs. Therefore any attempt towards developing therapeutic mechanism to overcome therapy resistance associated with PSMD9 demands a comprehensive understand of the many mechanism leading to the expression of the target genes of NFkB including IkBa, the contribution of other TFs, the role of kinases and their crosstalk. Since this also involves feed-back loops and it can become challenging to identify the activation/repression status of the genes involved both for experimental verification and computational approaches. This provided us with a case study: we considered the gene expression data from above and took TNFa, a gene induced by PSMD9 overexpression as the stimulus and IkBa as the target to help uncover the key players involved in the expression of IkBa, the endogenous inhibition of NFkB. From the literature and using domain knowledge, 4 candidate key actors were chosen, namely p38, ERK, PIK3CA and AKT. The Pareto-optimality curves generated for TNFa to IkBa are shown in Figure 4.

Biological Validation. Among nodes explored for functionality, we completed wetlab investigations at submission-time for ERK and AKT kinases, which showed PSMD9-induced phosphorylation. As mentioned earlier, we used a merged KEGG-graph combining activation and expression edges for simplicity. Since negative feedback loops involving both IkBa and A20 control NFkB activation and target gene expression, we also conducted experiments for both these targets after separating the composite graphs into activation and expression graphs (see [1]). Phosphorylation impacts (in)activation status of transcription factors (in-built in Response: KEGG) and hence must be integrated into gene expression studies. Indeed, excluding ERK from composite graphs did not induce Pareto shift, whereas separating into activation and expression graphs did. As can be gleaned from Table 1 (and [1]), ERK exclusion, but not AKT exclusion induced a Pareto shift



(a) Baseline vs excluding p38

(b) Baseline vs excluding ERK

(c) Baseline vs excluding PIK3CA

(d) Baseline vs excluding AKT

Fig. 4: Individual plots of the exclusion experiments for TNFa-IkBa

indicating its requirement in both IkBa and A20 expression. Only AKT induced IkBa (in)activation Pareto shift (see [1]). We tested ERK's significance in IkBa and A20 gene expression using qPCR. A two-fold decrease in A20 mRNA was observed in PSMD9 overexpression cells upon ERK inhibition [p=0.03] whereas AKT inhibition did not impact IkBa or A20 mRNA levels, a trend consistent even upon TNFa stimulation (t=3hrs). The lack of impact of ERK inhibition on IkBa mRNA levels is likely due to as yet unexplored PSMD9-specific effects. The NFkB-dependence for IkBa or A20 expression was evident from lack of solutions upon its exclusion. The routinely-used PD98059 and LY294002 signaling inhibitors achieved ERK ($\sim$ 100%) and AKT ($\sim$ 90%) phosphorylation inhibition, respectively, at recommended IC50 values. They may have off-target effects. *Importantly, these inhibition-dependent mRNA level changes were PSMD9-specific, consistent with computational predictions.*

6 Conclusion

We presented a novel problem formulation to capture functional signficance of a node in an interaction pathway between a stimulus and a target observation, in a highly noisy environment with minimal experimental data and using publicly available pathway databases. Our definition comes closest to a computational simulation of a knockout experiment that is classically done to establish the functional significance of a node in wet-lab experiments. After showing theoretical hardness results, we design practical encodings using SAT, which we implemented and validated by some wet-lab experiments and domain knowledge.

References

1. Akshay, S., Basu, S., Chakraborty, S., Sundararajan, R., Venkatraman, P.: Constraint-based functional significance checking in biological networks (supplementary material). https://github.com/sukanyabasu2009/network_tool_CP19 12, 13, 14, 15
2. Alcaraz, N., Kck, H., Weile, J., Wipat, A., Baumbach, J.: Keypathwayminer: Detecting case-specific biological pathways using expression data. Internet Mathematics 7(4), 299–313 (2011), http://dx.doi.org/10.1080/15427951.2011.604548 5
3. Bai, M., Ma, X., Li, X., Wang, X., Mei, Q., Li, X., Zhiqiang, w., Han, W.: The accomplices of NF-kB lead to radioresistance. Current Protein and Peptide Science 16 (04 2015) 14
4. Beltrame, L., Rizzetto, L., Paola, R., Rocca-Serra, P., Gambineri, L., Battaglia, C., Cavalieri, D.: Using pathway signatures as means of identifying similarities among microarray experiments. PLOS ONE 4(1), 1–11 (01 2009), https://doi.org/10.1371/journal.pone.0004128 5
5. Bolón-Canedo, V., Sánchez-Maroño, N., Alonso-Betanzos, A., Benítez, J.M., Herrera, F.: A review of microarray datasets and applied feature selection methods. Inf. Sci. 282, 111–135 (Oct 2014), https://doi.org/10.1016/j.ins.2014.05.042 2
6. Cavalieri, D., Castagnini, C., Toti, S., Maciag, K., Kelder, T., Gambineri, L., Angioli, S., Dolara, P.: Eu.gene analyzer a tool for integrating gene expression data with pathway databases. Bioinformatics 23(19), 2631–2632 (2007), +http://dx.doi.org/10.1093/bioinformatics/btm333 5
7. Chen, X., Xu, J., Huang, B., Li, J., Wu, X., Ma, L., Jia, X., Bian, X., Tan, F., Liu, L., Chen, S., Li, X.: A sub-pathway-based approach for identifying drug response principal network. Bioinformatics 27(5), 649–654 (2011), +http://dx.doi.org/10.1093/bioinformatics/btq714 5
8. Cline, M.S., Smoot, M., Cerami, E., Kuchinsky, A., Landys, N., Workman, C., Christmas, R., Avila-Campilo, I., Creech, M., Gross, B., Hanspers, K., Isserlin, R., Kelley, R., Killcoyne, S., Lotia, S., Maere, S., Morris, J., Ono, K., Pavlovic, V., Pico, A.R., Vailaya, A., Wang, P.L., Adler, A., Conklin, B.R., Hood, L., Kuiper, M., Sander, C., Schmulevich, I., Schwikowski, B., Warner, G.J., Ideker, T., Bader, G.D.: Integration of biological networks and gene expression data using cytoscape. Nat Protoc 2, 2366–82 (2007) 5
9. Drăghici, S., Khatri, P., Martins, R.P., Ostermeier, G., Krawetz, S.A.: Global functional profiling of gene expression. Genomics 81(2), 98 – 104 (2003), http://www.sciencedirect.com/science/article/pii/S0888754302000216 5
10. Dunn, S.J., Martello, G., Yordanov, B., Emmott, S., Smith, A.G.: Defining an essential transcription factor program for naïve pluripotency. Science 344(6188), 1156–1160 (2014), https://science.sciencemag.org/content/344/6188/1156 4, 5
11. Gebser, M., Schaub, T., Thiele, S., Veber, P.: Detecting inconsistencies in large biological networks with answer set programming. CoRR abs/1007.0134 (2010), http://arxiv.org/abs/1007.0134 4
12. Gebser, M., Schaub, T., Thiele, S., Veber, P.: Detecting inconsistencies in large biological networks with answer set programming. TPLP 11(2-3), 323–360 (2011), https://doi.org/10.1017/S1471068410000554 4
13. Geistlinger, L., Csaba, G., Kffner, R., Mulder, N., Zimmer, R.: From sets to graphs: towards a realistic enrichment analysis of transcriptomic systems. Bioinformatics

27(13), i366–i373 (2011), +http://dx.doi.org/10.1093/bioinformatics/btr228 5

14. Giacobbe, M., Guet, C.C., Gupta, A., Henzinger, T.A., Paixão, T., Petrov, T.: Model checking gene regulatory networks. In: Tools and Algorithms for the Construction and Analysis of Systems - 21st International Conference, TACAS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015. Proceedings. pp. 469–483 (2015) 5
15. Glaab, E., Baudot, A., Krasnogor, N., Schneider, R., Valencia, A.: Enrichnet: network-based gene set enrichment analysis. Bioinformatics 28(18), i451–i457 (2012), +http://dx.doi.org/10.1093/bioinformatics/bts389 5
16. Glaab, E., Baudot, A., Krasnogor, N., Valencia, A.: Topogsa: network topological gene set analysis. Bioinformatics 26(9), 1271–1272 (2010), +http://dx.doi.org/10.1093/bioinformatics/btq131 5
17. Godwin, P., Baird, A.M., Heavey, S., Barr, M., O'Byrne, K., Gately, K.: Targeting nuclear factor-kappa b to overcome resistance to chemotherapy. Frontiers in Oncology 3, 120 (2013), https://www.frontiersin.org/article/10.3389/fonc.2013.00120 14
18. Guerra, J., Lynce, I.: Reasoning over biological networks using maximum satisfiability. In: Milano, M. (ed.) Principles and Practice of Constraint Programming. pp. 941–956. Springer Berlin Heidelberg, Berlin, Heidelberg (2012) 4
19. Guziolowski, C., Borgne, M.L., Radulescu, O.: Checking consistency between expression data and large scale regulatory networks: A case study (2007) 4
20. Jong, H.D.: Modeling and simulation of genetic regulatory systems: A literature review. Journal of Computational Biology 9, 67–103 (2002) 4
21. Judeh, T., Johnson, C., Kumar, A., Zhu, D.: Teak: Topology enrichment analysis framework for detecting activated biological subpathways. Nucleic Acids Research 41(3), 1425–1437 (2013), +http://dx.doi.org/10.1093/nar/gks1299 5
22. Kanehisa, M., Goto, S.: KEGG: Kyoto Encyclopedia of Genes and Genomes. Nucleic Acids Res. 28, 27–30 (2000) 12
23. Kanehisa, M., Sato, Y., Kawashima, M., Furumichi, M., Tanabe, M.: KEGG as a reference resource for gene and protein annotation. Nucleic Acids Res. 44, D457–D462 (2016) 2, 5, 12
24. Koumakis, L., Kanterakis, A., Kartsaki, E., Chatzimina, M., Zervakis, M., Tsiknakis, M., Vassou, D., Kafetzopoulos, D., Marias, K., Moustakis, V., Potamias, G.: Minepath: Mining for phenotype differential sub-paths in molecular pathways. PLOS Computational Biology 12(11), 1–40 (11 2016), https://doi.org/10.1371/journal.pcbi.1005187 5
25. Krentel, M.W.: The complexity of optimization problems. J. Comput. Syst. Sci. 36(3), 490–509 (1988), https://doi.org/10.1016/0022-0000(88)90039-6 10
26. Langlands, F.E., Dodwell, D., Hanby, A.M., Horgan, K., Millican-Slater, R.A., Speirs, V., Verghese, E.T., Smith, L., Hughes, T.A.: Psmd9 expression predicts radiotherapy response in breast cancer. Molecular Cancer 13(1), 73 (Mar 2014), https://doi.org/10.1186/1476-4598-13-73 14
27. Lee, H., Shin, M.: Mining pathway associations for disease-related pathway activity analysis based on gene expression and methylation data. BioData Mining 10(1), 3 (Feb 2017), https://doi.org/10.1186/s13040-017-0127-7 5
28. Lee, S., Park, Y., Kim, S.: Midas: Mining differentially activated subpaths of kegg pathways from multi-class rna-seq data. Methods 124(Supplement C), 13 – 24 (2017), http://www.sciencedirect.com/science/article/pii/S1046202317300488, integrative Analysis of Omics Data 5

29. Li, C., Li, X., Miao, Y., Wang, Q., Jiang, W., Xu, C., Li, J., Han, J., Zhang, F., Gong, B., Xu, L.: Subpathwayminer: a software package for flexible identification of pathways. Nucleic Acids Research 37(19), e131 (2009), +http://dx.doi.org/10.1093/nar/gkp667 5
30. Lifschitz, V.: What is answer set programming? In: Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008. pp. 1594–1597 (2008), http://www.aaai.org/Library/AAAI/2008/aaai08-270.php 4
31. Ma, S., Kosorok, M.R.: Detection of gene pathways with predictive power for breast cancer prognosis. BMC Bioinformatics 11(1), 1 (Jan 2010), https://doi.org/10.1186/1471-2105-11-1 5
32. Martini, P., Sales, G., Massa, M.S., Chiogna, M., Romualdi, C.: Along signal paths: an empirical gene set approach exploiting pathway topology. Nucleic Acids Research 41(1), e19 (2013), +http://dx.doi.org/10.1093/nar/gks866 5
33. Melas, I.N., Samaga, R., Alexopoulos, L.G., Klamt, S.: Detecting and removing inconsistencies between experimental data and signaling network topologies using integer linear programming on interaction graphs. PLOS Computational Biology 9(9), 1–19 (09 2013) 4
34. de Moura, L., Bjørner, N.: Z3: An Efficient SMT Solver, pp. 337–340. Springer Berlin Heidelberg, Berlin, Heidelberg (2008), http://dx.doi.org/10.1007/978-3-540-78800-3_24 11
35. Nam, S., Chang, H., Kim, K., Kook, M., Hong, D., Kwon, C., Jung, H., Park, H., Powis, G., Liang, H., Park, T., Kim, Y.: Pathome: An algorithm for accurately detecting differentially expressed subpathways. Oncogene 33(41), 4941–4951 (3 2014) 5
36. Rajendra, J., Datta, K.K., Thorat, R., Kumar, K., Gardi, N., Kaur, E., Nair, J., Salunkhe, S., Patkar, K., Desai, S., Goda, J.S., Moiyadi, A., Dutt, A., Venkatraman, P., Gowda, H., Dutt, S.: Enhanced proteasomal activity is essential for long term survival and recurrence of innately radiation resistant residual glioblastoma cells. Oncotarget 9(25) (2018) 14
37. Ramadan, E., Alinsaif, S., Hassan, M.R.: Network topology measures for identifying disease-gene association in breast cancer. BMC Bioinformatics 17(7), 274 (Jul 2016), https://doi.org/10.1186/s12859-016-1095-5 5
38. Rhodes, D.R., Kalyana-Sundaram, S., Mahavisno, V., Varambally, R., Yu, J., Briggs, B.B., Barrette, T.R., Anstet, M.J., Kincead-Beal, C., Kulkarni, P., Varambally, S., Ghosh, D., Chinnaiyan, A.M.: Oncomine 3.0: Genes, pathways, and networks in a collection of 18,000 cancer gene expression profiles. Neoplasia 9(2), 166 – 180 (2007), http://www.sciencedirect.com/science/article/pii/S1476558607800479 5
39. Sahu, I., Sangith, N., Ramteke, M., Gadre, R., Venkatraman, P.: A novel role for the proteasomal chaperone PSMD9 and hnRNPA1 in enhancing IkBa degradation and NF-kB activation - functional relevance of predicted PDZ domain-motif interaction. {FEBS} Open Bio 281(11) (2014) 14
40. Sharan, R., Karp, R.M.: Reconstructing boolean models of signaling. Journal of Computational Biology 20(3), 249–257 (2013), https://doi.org/10.1089/cmb.2012.0241 5
41. Siegel, A., Radulescu, O., Borgne, M.L., Veber, P., Ouy, J., Lagarrigue, S.: Qualitative analysis of the relation between {DNA} microarray data and behavioral models of regulation networks. Biosystems 84(2), 153 – 174 (2006), http://www.sciencedirect.com/science/article/pii/S0303264705001723, dynamical Modeling of Biological Regulatory Networks 4

42. Soule, C.: Mathematical approaches to differentiation and gene regulation. Comptes Rendus Biologies 329(1), 13 – 20 (2006), http://www.sciencedirect.com/science/article/pii/S1631069105001800, modelisation de systemes complexes en agronomie et environnement 4
43. Steel, S., Alami, R.: Recent Advances in AI Planning: 4th European Conference on Planning, ECP'97, Toulouse, France, September 24 - 26, 1997, Proceedings. Lecture Notes in Computer Science / Lecture Notes in Artificial Intelligence, Springer (1997), https://books.google.co.in/books?id=QSBoQgAACAAJ 4
44. Subramanian, A., Tamayo, P., Mootha, V.K., Mukherjee, S., Ebert, B.L., Gillette, M.A., Paulovich, A., Pomeroy, S.L., Golub, T.R., Lander, E.S., Mesirov, J.P.: Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. Proceedings of the National Academy of Sciences 102(43), 15545–15550 (2005), http://www.pnas.org/content/102/43/15545.abstract 5
45. Tarca, A.L., Draghici, S., Khatri, P., Hassan, S.S., Mittal, P., Kim, J.s., Kim, C.J., Kusanovic, J.P., Romero, R.: A novel signaling pathway impact analysis. Bioinformatics 25(1), 75–82 (2009), +http://dx.doi.org/10.1093/bioinformatics/btn577 5
46. Thiele, S., Cerone, L., Saez-Rodriguez, J., Siegel, A., Guziołowski, C., Klamt, S.: Extended notions of sign consistency to relate experimental data to signaling and regulatory network topologies. BMC Bioinformatics 16(1), 345 (Oct 2015) 4
47. Toda, S.: PP is as hard as the polynomial-time hierarchy. SIAM J. Comput. 20(5), 865–877 (1991) 11
48. Ulitsky, I., Krishnamurthy, A., Karp, R.M., Shamir, R.: Degas: De novo discovery of dysregulated pathways in human diseases. PLOS ONE 5(10), 1–14 (10 2010), https://doi.org/10.1371/journal.pone.0013367 5
49. Vaske, C.J., Benz, S.C., Sanborn, J.Z., Earl, D., Szeto, C., Zhu, J., Haussler, D., Stuart, J.M.: Inference of patient-specific pathway activities from multi-dimensional cancer genomics data using paradigm. Bioinformatics 26(12), i237–i245 (2010), +http://dx.doi.org/10.1093/bioinformatics/btq182 5
50. Wang, L., Zhang, B., Wolfinger, R.D., Chen, X.: An integrated approach for the analysis of biological pathways using mixed models. PLOS Genetics 4(7), 1–9 (07 2008), https://doi.org/10.1371/journal.pgen.1000115 5
51. Wang, R.S., Saadatpour, A., Albert, R.: Boolean modeling in systems biology: an overview of methodology and applications. Physical Biology 9(5), 055001 (2012), http://stacks.iop.org/1478-3975/9/i=5/a=055001 4
52. Wang, Z., Gerstein, M., Snyder, M.: RNA-Seq: a revolutionary tool for transcriptomics. Nature Reviews Genetics 10(1), 57–63 (Jan 2009), http://dx.doi.org/10.1038/nrg2484 2
53. Warde-Farley, D., Donaldson, S.L., Comes, O., Zuberi, K., Badrawi, R., Chao, P., Franz, M., Grouios, C., Kazi, F., Lopes, C.T., Maitland, A., Mostafavi, S., Montojo, J., Shao, Q., Wright, G., Bader, G.D., Morris, Q.: The genemania prediction server: biological network integration for gene prioritization and predicting gene function. Nucleic Acids Research 38(suppl_2), W214–W220 (2010), +http://dx.doi.org/10.1093/nar/gkq537 5
54. Xia, J., Wishart, D.S.: Metpa: a web-based metabolomics tool for pathway analysis and visualization. Bioinformatics 26(18), 2342–2344 (2010), +http://dx.doi.org/10.1093/bioinformatics/btq418 5
55. Yu, H., Kim, M.P., Sprecher, E., Trifonov, V., Gerstein, M.: The importance of bottlenecks in protein networks: Correlation with gene essentiality and expression dynamics. PLoS Computational Biology 3(4) (2007), https://doi.org/10.1371/journal.pcbi.0030059 5