BIOINFORMATICS

Vol. 30 ISMB 2014, pages i96–i104
doi:10.1093/bioinformatics/btu262

Large scale analysis of signal reachability

Andrei Todor*, Haitham Gabr, Alin Dobra and Tamer Kahveci
CISE Department, University of Florida, Gainesville, FL 32611, USA

ABSTRACT

Motivation: Major disorders, such as leukemia, have been shown to alter the transcription of genes. Understanding how gene regulation is affected by such aberrations is of utmost importance. One promising strategy toward this objective is to compute whether signals can reach to the transcription factors through the transcription regulatory network (TRN). Due to the uncertainty of the regulatory interactions, this is a #P-complete problem and thus solving it for very large TRNs remains to be a challenge.

Results: We develop a novel and scalable method to compute the probability that a signal originating at any given set of source genes can arrive at any given set of target genes (i.e., transcription factors) when the topology of the underlying signaling network is uncertain. Our method tackles this problem for large networks while providing a provably accurate result. Our method follows a divide-and-conquer strategy. We break down the given network into a sequence of non-overlapping subnetworks such that reachability can be computed autonomously and sequentially on each subnetwork. We represent each interaction using a small polynomial. The product of these polynomials express different scenarios when a signal can or cannot reach to target genes from the source genes. We introduce polynomial collapsing operators for each subnetwork. These operators reduce the size of the resulting polynomial and thus the computational complexity dramatically. We show that our method scales to entire human regulatory networks in only seconds, while the existing methods fail beyond a few tens of genes and interactions. We demonstrate that our method can successfully characterize key reachability characteristics of the entire transcriptions regulatory networks of patients affected by eight different subtypes of leukemia, as well as those from healthy control samples.

Availability: All the datasets and code used in this article are available at bioinformatics.cise.ufl.edu/PReach/scalable.htm.

Contact: atodor@cise.ufl.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

Major disorders, such as cancer, have been shown to alter the transcription of a large number of genes and thus affect the mechanism that governs cells functions (Krivtsov, 2009; Valk *et al.*, 2004). Many complex disorders, such as acute lymphoblastic leukemias, however, yield a varying spectrum of expression profiles and, as a result, cannot be robustly characterized by merely studying the gene expressions (Armstrong, 2002).

An important part of cell biology research is the study of the causal relationship between extracellular conditions and the cell response. Such causality is governed by a chain of biochemical reactions through which extracellular signals are transmitted from membrane receptors to transcription factors (i.e., reporters) via protein–protein interactions (Bu and Callaway, 2011). While the pattern of this mechanism is similar for all organisms, important variations in its quantitative aspects such as gene expressions result from external perturbations, differentiation stage of the cell, timing of DNA replication and various epigenetic mutations (Los *et al.*, 2009; Mattick *et al.*, 2009). Therefore, detecting these quantitative variations is an important source of information for assessing the fitness of the organism and ultimately for diagnosis and prognosis.

Extensive evidence suggests that there is a degree of uncertainty in our knowledge of interactions within cells (Bader *et al.*, 2004; Ceol *et al.*, 2010; Deng *et al.*, 2003; Ourfali *et al.*, 2007; Sharan *et al.*, 2002; Suthram *et al.*, 2006; Szklarczyk *et al.*, 2011; von Mering *et al.*, 2002). The source of this uncertainty is 2-fold. First, the biological processes that are modeled as protein interactions in biological networks are stochastic events (Bader *et al.*, 2004). Second, the evidence in support of an interaction is not entirely decisive for the actual presence of the interaction (Bader *et al.*, 2004; Ourfali *et al.*, 2007; Sharan *et al.*, 2002; Shlomi *et al.*, 2006) due to many reasons, such as epigenetic variations across different cells (Gerstein *et al.*, 2012). Several schemes have already been proposed to assess the reliability of protein interactions in the form of confidence values (Bader *et al.*, 2004; Deng *et al.*, 2003; Suthram *et al.*, 2006). Such interaction confidence values are now available in large biological network databases, such as MINT (Ceol *et al.*, 2010) and STRING (Szklarczyk *et al.*, 2011).

Recent studies often model the uncertainty of the interactions in biological networks using probabilistic networks (Gabr *et al.*, 2013; Todor *et al.*, 2012; Todor *et al.*, 2013). We adopt the same model in this article, namely, each node of the network denotes a gene and the directed edge from a node v_i to node v_j denotes that the gene corresponding to v_i can regulate the gene denoted by v_j through an interaction. Each edge in this network is a probabilistic event. That is, it is considered possible, but not certain, reflecting the insecure knowledge of the gene regulation process. A common way to model the uncertainty of each edge is to associate it with a probability value, which is computed for each interaction from several factors: gene expressions, available evidence for it and network topology around it (Sharan *et al.*, 2002).

The ability to compute confidence values for interactions provides opportunities to model and study biological networks accurately. It, however, comes at a high price as the uncertainty of the topology of interactions makes studying biological networks a computationally challenging task. The challenge is that a probabilistic network represents a large number of alternative deterministic network topologies. More precisely, a network with n probabilistic edges yields 2^n possible network configurations, as each one of the n edges may be present or absent. For instance, in

*To whom correspondence should be addressed.

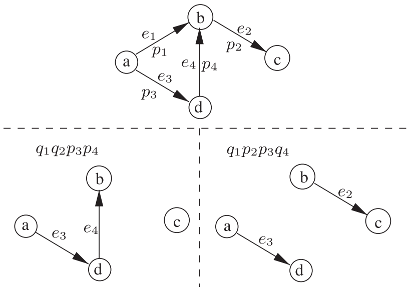

Fig. 1. A probabilistic network (top) and two of the deterministic networks corresponding to it (bottom). Each of the deterministic networks is obtained from the probabilistic network with some probability determined by the probabilities of the edges that are included or not in the deterministic network. p_i denotes the probability of edge e_i being present. $q_i = 1 - p_i$ is the probability of the edge being absent. The expression above each deterministic network is the probability of observing it

Figure 1, the probabilistic network shown on top corresponds to 16 deterministic networks since it contains 4 probabilistic edges.

In this article, we address the problem of characterizing the signaling reachability in transcription regulatory networks (TRNs). Unlike most of the existing literature, we eliminate the limitations of the classical assumption that all interactions are deterministic and adopt the more descriptive probabilistic network. More specifically, given a set of source genes $S = \{s_1, s_2, \ldots, s_a\}$ and a set of target genes $T = \{t_1, t_2, \ldots, t_b\}$, we compute the *reachability profile* of that network as a doubly indexed vector R where, for all i, j such that $1 \leq i \leq a$, $1 \leq j \leq b$, the entry $R[i,j]$ is the probability that a signal originating at s_i can reach t_i (i.e., s_i regulates t_j). We show that the reachability profile can help us understand how different disorders that alter the cellular functions based on the signaling patterns of the gene regulatory networks. We particularly focus on leukemias, which is challenging due to the heterogeneity of the transcription patterns.

Summary of related work. The problem of computing reachability in uncertain network topologies has drawn significant attention in the context of network reliability. Various exact methods, as well as approximate methods, have been proposed. We refer interested readers to several surveys on the topic (Aggarwal *et al.*, 1975; Hwang *et al.*, 1981). Theoretical results on the complexity of the problem reveal that it is #P complete (Brown and Colbourn, 1996; Husfeldt and Taslaman, 2010; Provan and Ball, 1983). The problem is significantly simplified in the case of acyclic graphs. This type of graphs can be represented as Bayesian Networks, for which various inference algorithms exist. However, for this simple case sophisticated inference algorithms are unnecessary. In the context of biological networks, the problem for general graphs was first addressed by Ourfali *et al.* (2007). The goal of these authors was to infer the structure of the signaling network that best explains a set of gene knockout pairs, given a protein–protein interaction network. To achieve this goal, they developed a method to compute the reachability probability for each knockout gene pair. Their method is an exact solution based on the inclusion–exclusion principle (van Lint and Wilson, 1992). However, due to its high time complexity, this method works accurately only for very small networks (i.e., those with a few tens of nodes). PReach (Gabr *et al.*, 2013) computes the exact reachability probability based on polynomial multiplication. It is significantly faster than the inclusion–exclusion method of Ourfali *et al.* (2007) for networks where there are many paths. However, it does not scale to large networks. *Thus, the existing solutions cannot be used to study entire TRNs, and there is a great need for accurate yet efficient methods.*

Contributions. Here, we develop a novel method that computes the probability that a signal originated at a given source gene can reach to a given target gene in a given probabilistic network. Unlike existing methods, our solution is both precise (i.e., it computes this probability without error) and it scales to large networks. Our method follows a divide-and-conquer approach. We partition the given probabilistic network into a sequence of loosely connected clusters of nodes. On the boundary between two such consecutive clusters lies a set of nodes called *node separators*. Any signal which originates from the source node and arrives at any node in the latter cluster must visit the node separators. Similar to PReach (Gabr *et al.*, 2013), we model the given probabilistic network using polynomials. The form of the polynomials of our method however differs from that of PReach in a way that allows us to collapse the polynomial to very small size that is determined by the size (number of interactions) of the clusters and the number of nodes in a given boundary. Each term in our polynomial evaluates the existence probability of a collection of subsets of interactions. In brief, instead of computing the reachability probability from the given source node to the target node, we incrementally compute the reachability probability from the source node to each node separator in sequential order. That allows us to avoid storing a massive fraction of terms of the polynomial (i.e., the terms corresponding to the nodes in earlier clusters). Our experimental results on real and synthetic datasets demonstrate that our method scales to very large network sizes while the inclusion–exclusion method (Ourfali *et al.*, 2007) and PReach (Gabr *et al.*, 2013) fail. We also observe that the reachability profiles provide a valuable resource for characterizing leukemias and differentiating the centrality of the genes across different leukemias as well as healthy control groups.

In summary, the key contributions of this work are:

- We introduce a new quantity for evaluating the state of a biological network, the *reachability profile*.
- We introduce a novel, fast and scalable method to compute the reachability profile of large networks, based on polynomials and *polynomial collapsing operators*.
- We demonstrate the usefulness of reachability profiles in *detailed analysis of different types of leukemias*.

The rest of the article is organized as follows. Section 2 describes our method. Section 3 presents our experimental results. Section 4 concludes with a brief discussion.

2 METHOD

In this section we present our method in detail. We first define the essential theoretical concepts in Section 2.1. We then present an overview of our method in Section 2.2. We discuss how to compute intermediate reachability probabilities in Section 2.3. We elaborate on how to partition the network in Section 2.4.

2.1 Preliminary definitions

We start by formally defining the probabilistic network concept. We provide a list of notations used throughout the article in the Supplementary Material.

Definition 1 (probabilistic network). A probabilistic network *is a graph* $\mathcal{G} = (V, E, P)$*, where V is the set of nodes, E is the set of edges and* $P : E \to (0, 1]$ *is a function that associates to each edge a probability value.*

In our context, each node in V represents a gene, each edge in E represents an interaction between two genes and P associates to each edge the probability of the existence of the interaction it represents. For instance, in Figure 1 (top figure), $V = \{a,b,c,d\}$ and $E = \{e_1,e_2,e_3,e_4\}$. We assume that each edge exists independent of all other edges. This assumption is commonly used in the literature for similar problems (Ceol *et al.*, 2010; Szklarczyk *et al.*, 2011). We limit our description to directed networks, although undirected networks can be dealt with by replacing each undirected edge with two edges in opposite directions.

Given a probabilistic network $\mathcal{G} = (V, E, P)$, we call the deterministic network $G = (V,E)$ the *maximal deterministic network* of $\mathcal{G}$. In other words, the maximal deterministic network is the deterministic network in which all possible interactions of $\mathcal{G}$ are present.

The computational problem we address in this article is: given a probabilistic network, $\mathcal{G} = (V, E, P)$, a source node $s \in V$ and a target node $t \in V$, what is the probability that the t can be reached from s?

Next, we define a graph concept, *node separator*, which will help us in explaining our method.

Definition 2 (Node Separator). *Let* $G = (V,E)$ *be a deterministic network and* $s, t \in V$ *be two of its nodes. An s-t node separator of G is a set of nodes* $K \subseteq V$ *whose removal disconnects t from s in G.*

Figure 2 illustrates this concept. Here, the source node s and the target node t are labeled with 1 and 8, respectively. The set of nodes {4, 5} is an *s-t* node separator. We say that a node separator is *minimal* if none of its proper subsets is a node separator.

A node separator partitions the nodes of that network into three disjoint subsets:

(1) The *left nodes* are the nodes that are reachable from the source, but the target cannot be reached from any of them without going through the node separator (e.g., nodes 1, 2 and 3 in Figure 2, for the node separator {4, 5}).

(2) The node separator itself (e.g., nodes 4 and 5 in Fig. 2).

(3) The *right nodes* are the remaining nodes (e.g., nodes 6, 7 and 8 in Fig. 2). Notice that these are the nodes from which the target can be reached, but they are not reachable from the source without passing through the node separator.

A node separator K also partitions the edges of the given network into three subsets:

(1) The *left edges* are the edges between the nodes in the union of left and separator nodes (e.g., edges e_1,e_2,e_3,e_5 and e_6 in Fig. 2). We denote the set of left edges with $L(K)$.

(2) The *right edges* are the edges between right nodes or from a separator node to a right node (e.g., edges e_7,e_9,e_{10},e_{11} in Fig. 2).

(3) The *backward edges* are the edges from right nodes to the separator nodes or from right nodes to left nodes (edges e_4,e_8 in Fig. 2).

Theorem 1. *Let* $G = (V,E)$ *be a deterministic network. Given two nodes,* $s, t \in V$*, let K be an s-t node separator. For any right node u of K, it is guaranteed that K is also an s-u node separator.*

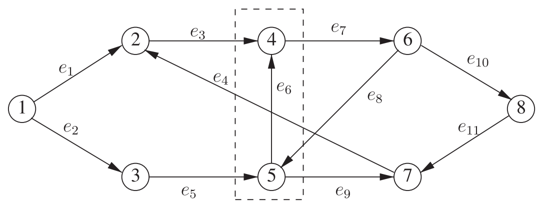

Fig. 2. A network with an *s-t* node separator. The source is node 1 and the target is node 8. The dotted rectangle indicates an *s-t* node separator

We prove Theorem 1 in the Supplementary Material.

If a node separator does not yield backward edges, we call it a *good node separator*. We only use good node separators in the rest of the article. So, in what follows by node separator we refer to a good node separator, unless otherwise specified. Finally we define the concept of subset reachability in probabilistic networks.

Definition 3 (Subset Reachability). *Let* $\mathcal{G} = (V, E, P)$ *be a probabilistic network. Let s and t* ($s, t \in V$) *be source and target nodes in* $\mathcal{G}$*. Consider two s-t node separators* K_i *and* K_j *of* $\mathcal{G}$ *such that for all nodes* $u \in K_j \backslash K_i$*, u is a right node of* K_i*. Let S and T be two subsets of* K_i *and* K_j*, respectively. We say that* K_j *is T-reachable from S if all nodes in T are reachable from at least one node in S and none of the nodes in* $K_j \backslash T$ *is reachable from any node in S. We denote the probability that* K_j *is T-reachable from S by* $p(S,T,K_j)$.

2.2 Overview of the method

Our method works in two steps.

Step 1. Given a probabilistic network $\mathcal{G} = (V, E, P)$ and source and target nodes s and t, in the first step, we partition $\mathcal{G}$ into a sequence of subnetworks that are connected to each other through node separators. In general terms, let us denote the sequence of node separators with K_0, K_1, …, K_c, K_{c+1}, where $K_0 = \{s\}$ and $K_{c+1} = \{t\}$. We choose these node separators such that $\forall i < j$, for all nodes $u \in K_j \backslash K_i$, u is a right node of K_i. Following from Definition 3, the problem we solve in this article is equivalent to computing $p(K_0, K_{c+1}, K_{c+1}) = p(\{s\}, \{t\}, \{t\})$.

Step 2. At this step we compute the reachability probability from s to t. More specifically, using this notation above, for any i ($0 < i \leq c$), we write the probability $p(\{s\},T,K_{i+1})$ as

$$p(\{s\}, T, K_{i+1}) = \sum_{S \subseteq K_i, S \neq \emptyset} p(\{s\}, S, K_i) p(S, T, K_{i+1}) \qquad (1)$$

The case $i = 0$ is a special one. Since K_0 contains s, we have $T = \{s\}$. Thus the probability to reach the source node is 1. Following from Equation (1), our algorithm iteratively computes $p(K_0,K_{c+1},K_{c+1})$ by moving from one node separator to the next, starting from K_0.

The correctness of Equation (1) follows from the definition of node separator and Theorem 1. More specifically, in order to reach to any node in $T \subseteq K_{i+1}$, we have to visit at least one node in K_i. The product $p(\{s\},S,K_i)p(S,T,K_{i+1})$ in Equation (1) is the probability that a signal reaches T by visiting all the nodes in S and no other node in K_i– S. The summation in this equation enumerates all possible subsets $S \subseteq K_i$. Thus, it is accumulates the probability of all possible alternative routes from s to T defined by all possible subsets S.

Figure 3 illustrates our method. In this example, the set of edges in E is split into three non-overlapping sets using four node separators K_0, K_1, K_2 and K_3 where $K_0 = \{s\}$ and $K_3 = \{t\}$. These sets are $L(K_1)$, $L(K_2) \backslash L(K_1)$, and $L(K_3) \backslash L(K_2)$. Each of these sets define a subnetwork of $\mathcal{G}$. Once the network is partitioned this way, instead of computing the reachability

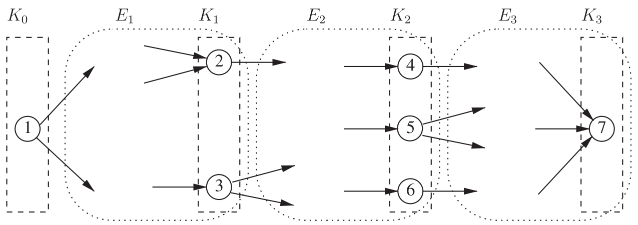

Fig. 3. A hypothetical network with two disjoint *s-t* node separators, $K_1 = \{2,3\}$ and $K_2 = \{4,5,6\}$. Source and target nodes are labeled with $s = 1$ and $t = 7$. For uniformity, we consider $K_0 = \{1\}$ and $K_3 = \{7\}$ also to be node separators

probability directly from s to t, we compute it incrementally by advancing from one node separator to the next. For the example in Figure 3, we first consider the separator K_1, then K_2, finally K_3. At each node separator, we only consider the subnetwork which contains the left edges of that separator.

To understand Equation (1) better, consider the separators K_1 and K_2 in Figure 3. There are three possible scenarios to reach to a subset T of K_2, say $T = \{4\}$ from $s = 1$. Each of these scenarios corresponds to a nonempty subset of $K_1 = \{2, 3\}$.

1. Visit $S = \{2\}$ and do not visit $K_1 \backslash S = \{3\}$. This happens with probability $p(\{1\},\{2\},K_1)p(\{2\},\{4\},K_2)$.
2. Visit $S = \{3\}$ and do not visit $K_1 \backslash S = \{2\}$. This happens with probability $p(\{1\},\{3\},K_1)p(\{3\},\{4\},K_2)$.
3. Visit both nodes in $S = \{2, 3\}$. This happens with probability $p(\{1\},\{2,3\},K_1)p(\{2,3\},\{4\},K_2)$.

The sum of the three probabilities above yields the probability $p(\{1\},\{4\},K_2)$.

In Section 2.3, we explain how we compute Equation (1) efficiently for $i>0$ (Step 2). In Section 2.4, we explain how we choose the node separators (Step 1).

2.3 Computing the reachability probability

In Equation (1), we presented an iterative formula to compute the reachability probability $p(\{s\},\{t\},\{t\})$ by splitting the network using cuts $K_0, K_1, \ldots, K_{c+1}$. Although this equation reduces the scale of the problem to the subnetworks between consecutive cuts, computing it efficiently still remains to be a challenge. Here, we describe how we compute this probability efficiently yet provably correctly. More specifically, given two consecutive node separators K_i and K_{i+1} $(0<i\leq c)$ and given $p(\{s\},S,K_i)$ for all subsets $S \subseteq K_i$, we discuss how we compute $p(\{s\},T,K_{i+1})$ for all subsets $T \subseteq K_{i+1}$.

From the definition of left edges, we know that the probability $p(\{s\},S,K_i)$ depends only on the edges in $L(K_i)$. This is because $L(K_i)$ contains all the edges that can lie on a path from s to any node in K_i. Let us denote the set of edges in $L(K_j)\backslash L(K_{j-1})$ with E_j for any $0<j$ (i.e., left edges of K_j which are also right edges of K_{j-1}). Thus, the probability $p(\{s\},T,K_{i+1})$, depends only on the edges in E_{i+1} when $p(\{s\},S,K_i)$ is given for all S. This implies that it is possible to compute the probability $p(\{s\},T,K_{i+1})$ by considering only the edges in E_{i+1} when $p(\{s\},S,K_i)$ is known $\forall S \subseteq K_i$. Below, we compute this probability by transforming the probabilistic network into a collection of polynomials.

Transformation into polynomial space. Assume that the given probabilistic network, $\mathcal{G}=(V, E, P)$, contains n edges and m nodes, denoted with $E = \{e_1,e_2,\ldots e_n\}$ and $V = \{v_1,v_2,\ldots v_m\}$, respectively. As the first step of the transformation, we associate to each edge a polynomial called the *edge polynomial.* More precisely, for edge $e_i \in E$, let $p_i = P(e_i)$ and $q_i = 1- p_i$ denote the existence and absence probability of e_i, respectively. We define the edge polynomial of e_i as the first degree polynomial of two variables, x_i and y_i, $F_i(x_i,y_i) = p_ix_i + q_iy_i$.

Consider a subset E' of the edges in E. We define the *edge aggregation polynomial* for E', denoted with $F(E')$, as the product of all the edge polynomials associated with the edges in E':

$$F(E') = \prod_{e_i\in E'} F_i(x_i, y_i) = \sum_{\mathcal{E}\subseteq E'} \prod_{e_i\in\mathcal{E}} p_i x_i \prod_{e_j\in E'\setminus\mathcal{E}} q_j y_j. \qquad (2)$$

Notice that each term in the summation above corresponds to one of the possible deterministic configurations for the network topology. The coefficient of the term $\prod_{e_i\in\mathcal{E}} x_i \prod_{e_j\in E'\setminus\mathcal{E}} y_j$ in F is the probability of observing all the edges in $\mathcal{E}$ and not observing any edge in $E'\backslash\mathcal{E}$. To understand this better, consider the network in Figure 1 (network on the top). In the edge aggregation polynomial of this network, the term $x_3x_4y_1y_2$ corresponds to the deterministic instance where only edges e_3 and e_4 are present (i.e., bottom left network in Fig. 1). The coefficient of this term is $q_1q_2p_3p_4$ which is the probability of observing that network instance.

Reachability in polynomial space. As we explain in Equation (2), the terms of the edge aggregation polynomial represent different deterministic network configurations. Thus, the probability $p(\{s\},T,K_{i+1})$ is equal to the sum of the coefficients of a specific subset of the polynomial terms: The terms which yield a topology where K_{i+1} is T-reachable from $\{s\}$.

At this point, the polynomial transformation seemingly makes the reachability problem as complicated as exhaustively enumerating all network topologies. This is because, (i) the edge aggregation polynomial has as many terms as the number of network topologies; and (ii) finding the subset of polynomial terms which yield the desired topologies will incur additional computational cost. Below, we build a novel algebra on the edge aggregation polynomial to compute this value by enumerating only a tiny fraction of the polynomial terms.

Algorithm 1 presents a pseudocode that describes our algorithm for constructing the polynomial needed to compute $p(\{s\},T,K_{i+1})$. The algorithm takes the existing edge aggregation polynomial for the edges in $L(K_i)$ as input. At each iteration it grows that polynomial by aggregating it with the edge polynomial of a new edge in E_{i+1} (Step 2). It then reduces the size of the resulting polynomial by collapsing it (Step 3). Briefly, the collapsation step merges all terms which correspond to configurations in which K_{i+1} is T-reachable from s, for each possible subset T of K_{i+1}, into a single term by replacing the variables in these terms with a single variable denoted with z_T. Thus, the coefficient of z_T is the sum of the coefficients of the original terms that were collapsed. In the rest of this section, we elaborate on these steps, particularly the collapsation step.

Algorithm 1 Compute the edge aggregation polynomial for $L(K_{i+1})$

```
Require: Probabilistic graph $\mathcal{G}=(V, E, P)$
Require: Node separators $K_i$ and $K_{i+1}$
Require: Edge aggregation polynomial $F' = F(L(K_i))$.
1: for all $e_j \in E_{i+1}$ do
2:     Aggregate edge polynomial of $e_j$ as $F' = F' \times F_j(x_j,y_j)$
3:     Collapse $F'$
4: end for
```

We start by introducing some notation which will simplify our polynomial algebra below. For a subset of edges $\mathcal{E} \subseteq E$, we denote the set of indices of the edges in $\mathcal{E}$ by $Ind(\mathcal{E})$. For instance, for $\mathcal{E}=\{e_2, e_3, e_8\}$, we have $Ind(\mathcal{E})=\{2, 3, 8\}$

Let us denote the subset of edges of E_{i+1} which have been multiplied into the edge aggregation polynomial so far with $\mathcal{E} \subseteq E_{i+1}$ and its set of indices with $\Theta = Ind(\mathcal{E})$.

Following from Equation (2), since $\mathcal{E}$ and $L(K_i)$ are disjoint, we can write the edge aggregation polynomial of the edge set $\mathcal{E} \cup L(K_i)$ as

$F(\mathcal{E})F(L(K_i))$. To simplify our notation of $F(\mathcal{E})$, for all $I \subseteq \Theta$, we denote $\prod_{i\in I} x_i$ and $\prod_{i\in\Theta\setminus I} y_i$ with $\mathbf{x_I}$ and $\mathbf{y}_{\Theta\setminus\mathbf{I}}$, respectively. We denote the coefficient of $\mathbf{x_I}\mathbf{y}_{\Theta\setminus\mathbf{I}}$ with α_I. Thus, we can write the first polynomial as $F(\mathcal{E}) = \sum_{I\subseteq\Theta} \alpha_I \mathbf{x_I}\mathbf{y}_{\Theta\setminus\mathbf{I}}$.

For each node separator K_i, we define a unique collapsing operator and denote it with $\rho_i()$. This is a linear operator; it acts on the terms of the given edge aggregation polynomial for the edges in $L(K_i)$ independently. Briefly, the collapsed polynomial contains a new variable, z_s, for each subset S of K_i. The form of this polynomial is $\rho_i(F(L(K_i))) = \sum_{S\subseteq K_i} \beta_S z_S$. In this representation, z_S corresponds to the case where K_i is S-reachable from K_0 (i.e., the original source node), and the coefficient β_S is the probability of observing that case. In other words β_S is equal to $p(\{s\},S,K_i)$ in Equation (1). We explain how this operator works and how we compute it in detail later in this section. For the moment, assume that we have already applied it for the edge set $L(K_i)$. Therefore we replace the polynomial, $F(L(K_i))$ in the product $F(\mathcal{E})F(L(K_i))$ with its collapsed version, denoted by $\rho_i(F(L(K_i)))$.

After multiplying the first polynomial and the collapsed version of the second polynomial, we get

$$F(\mathcal{E})\rho_i(F(L(K_i))) = \sum_{I\subseteq\Theta} \alpha_I \mathbf{x_I}\mathbf{y}_{\Theta\setminus\mathbf{I}} \sum_{S\subseteq K_i} \beta_S z_S. \qquad (3)$$

Since this product includes edge polynomials from the edge set $\mathcal{E}$, we further reduce its size by applying the collapsing operator $\rho_{i+1}()$ on it and thus obtain $\rho_{i+1}(F(\mathcal{E})\rho_i(F(L(K_i))))$.

Next, we explain how the collapsing operator works. Given two nodes $u, v \in V$, let π be a path from u to v in the maximal deterministic network $G = (V,E)$ of the given probabilistic network. Here, by path we mean the set of edges traversed to reach from u to v. Let I be a subset of indices, $I \subseteq \{1, \ldots, n\}$. We define two set indicator functions $\chi_{u,v}()$ and $\omega_{u,v}()$ for the node pair (u, v). The first one takes the value $\chi_{u,v}(I) = 1$ if there is a path π from u to v such that $Ind(\pi) \subseteq I$ and 0 otherwise. For instance, in Figure 2, $\chi_{1,8}(\{1,2,5,6,7,10,11\}) = 1$. This is because $\{e_2,e_5,e_6,e_7,e_{10}\}$ forms a path from 1 (source) to 8 (target) and its set of indices $\{2, 5, 6, 7, 10\}$ is a subset of the input set $\{1, 2, 5, 6, 7, 10, 11\}$. Similarly, the second indicator function takes the value $\omega_{u,v}(I) = 1$ if there is a minimal u- v cut κ such that $Ind(\kappa) \subseteq I$ and 0 otherwise. For example, $\omega_{1,8}(\{2,3,4,5\}) = 1$, because $\{e_3,e_5\}$ forms a minimal cut between nodes 1 and 8 and its set of indices $\{3, 5\}$ is a subset of input set $\{2, 3, 4, 5\}$.

Next, we extend the definitions of the set indicator functions χ and ω to multiple source nodes. The extended function $\chi_{S,v}(I)$ evaluates to 1 if there is a path π from at least one node u in S to v such that $Ind(\pi) \subseteq I$ and 0 otherwise. Similarly, $\omega_{S,v}(I)$ evaluates to 1 if for all nodes $u \in S$ there is at least a minimal u- v cut κ such that $Ind(\kappa) \subseteq I$ and 0 otherwise. Formally, we compute these functions as

$$\chi_{S,u}(I) = 1 - \prod_{s\in S}(1 - \chi_{s,u}(I)) \text{ and } \omega_{S,u}(I) = \prod_{s\in S} \omega_{s,u}(I) \qquad (4)$$

Next, we formalize T-reachability of the node separator K_{i+1}. For this purpose, we define a new set indicator function $C_{S,T}()$ which evaluates to 1 only if K_{i+1} is T-reachable from S. Otherwise, it evaluates to 0. We compute this function as

$$\mathcal{C}_{S,T}(I) = \prod_{u\in T} \chi_{S,u}(I) \prod_{v\in K_{i+1}\setminus T} \omega_{S,v}(\Theta\setminus I). \qquad (5)$$

We prove the correctness of Equations (4) and (5) in the Supplementary Materials.

Now we are ready to put all the pieces together and compute the collapsing operator ρ_{i+1}. Recall that each term of the given edge aggregation polynomial indicates a deterministic subnetwork topology for the edges in $\mathcal{E}$, combined with all deterministic topologies of the edges in $L(K_i)$ in which K_i is S-reachable from K_0, for every $S \subseteq K_i$. If that combination ensures that K_{i+1} is T-reachable from K_0, then the collapsing operator ρ_{i+1} replaces all the variables of that term with z_T. More specifically, consider a term in Equation (3) after the product has been expanded, in the form $\gamma_{I,S}\mathbf{x_I}\mathbf{y}_{\Theta\setminus\mathbf{I}} z_S$, where $\gamma_{I,S} = \alpha_I\beta_S$. We compute the collapsing operator $\rho_{i+1}()$ on this term as

$$\begin{aligned}\rho_{i+1}\Big(\gamma_{I,S}\mathbf{x_I}\mathbf{y}_{\Theta\setminus\mathbf{I}} z_S\Big) &= \gamma_{I,S} \sum_{T\subseteq K_{i+1}} \mathcal{C}_{S,T}(I) z_T. \\ &\quad + \gamma_{I,S}\left(\prod_{T\subseteq K_{i+1}} \big(1 - \mathcal{C}_{S,T}(I)\big)\right)\mathbf{x_I}\mathbf{y}_{\Theta\setminus\mathbf{I}} z_S\end{aligned} \qquad (6)$$

The collapsing operator $\rho_{i+1}()$ [see Equation (6)] transforms each term of the polynomial into a single term. The resulting term either contains the variable z_T, where $T \subseteq K_{i+1}$, or remains unchanged. This is because $\mathcal{C}_{S,T}$ either takes the value 0 or 1. Thus, $\rho_{i+1}()$ leaves the term unchanged only if $\mathcal{C}_{S,T} = 0$ for all T. When, $\mathcal{C}_{S,T} = 1$ for some $T \subseteq K_{i+1}$, the coefficient of z_S becomes 0. It returns $\gamma_{I,S} z_T$ in this case. Furthermore, from Equation (5), we know that if $\mathcal{C}_{S,T} = 1$, then for all $T' \neq T$ ($T' \subseteq K_{i+1}$), $\mathcal{C}_{S,T'} = 0$. Thus, the function $\rho_{i+1}()$ returns no other term containing variable z'_T.

Now suppose that a term has collapsed to z_T and a new edge e_j is added in Step 2 of Algorithm 1. From a polynomial point of view, the z_T variable will be multiplied with x_j and y_j, respectively, resulting in two new terms. From the graph reachability point of view, we know that the edges added prior to e_j already ensure T-reachability, so e_j does not make any differece: both its presence and its absence lead to reachable graph configurations. In the polynomial, the coeficients of $z_T x_j$ and $z_T y_j$ have to be added together to obtain the reachability probability. To take advantage of this observation, we introduce a special multiplication rule for the z_T variables: both $z_T x_j$ and $z_T y_j$ are replaced with z_T, for all $e_j \in E_{i+1}$, so that their coefficients are added together.

The collapsing operator is very powerful as it ensures that the size of the edge aggregation polynomial never exceeds $2^{|K_i|+|E_{i+1}|}$ in the worst case (i.e., when the indicator function $C_{S,T}()$ always returns 0 until the last edge in $L(K_{i+1})$ is aggregated). More importantly, it guarantees to reduce the polynomial size down to $2^{|K_{i+1}|}$ once the edges in $L(K_{i+1})$ are all aggregated. This is a significant improvement as without the collapsation function, the size of the edge aggregation polynomial $2^{|L(K_{i+1})|}$ after considering K_{i+1} and it goes up to $2^{|E|}$ after including all the edges.

So what is the reachability probability? After all the edges in E_{i+1} have been added, all the terms will collapse, and the polynomial will be $\rho_{i+1}(F(L(K_{i+1}))) = \sum_{T\subseteq K_{i+1}} \gamma_T z_T$. When $K_{c+1} = \{t\}$ is reached, the polynomial will have only two terms: $\gamma_{\{t\}} z_{\{t\}} + \gamma_\emptyset z_\emptyset$. The coefficient $\gamma_{\{t\}}$ is equal to the probability that the target node is reachable from the source node. We prove the correctness of our method in the Supplementary Material.

2.4 How to choose node separators

Depending on the topology of the maximal deterministic network there can be many alternative sequences of node separators between the source and target nodes. Regardless of how we choose the node separators, our method guarantees to return the correct result. The node separator choice however can affect the size of the intermediate polynomials and thus the running time of our method in two ways. (i) Ideally, each node separator K_i should contain a small number of nodes as it will produce $2^{|K_i|}$ variables of the form z_S. (ii) Each consecutive node separators should contain a small number of edges between them (i.e., E_i should be small). This is because, in the worst case, they yield $2^{|E_i|}$ terms. Finding an optimal sequence of node separators that minimizes the overall computation time is in itself an intriguing area worth investigating. The right balance between the separator size, the size of the edge sets between the separators and the amount of computation we are willing to spend on finding the solution is hard to find. Here, we use a greedy approach to find good node separators.

We consider the first node separator (K_0) to be the source node itself. We determine the next node separator from the current one by considering all nodes that are one edge further from the current node separator. The set of nodes identified in this way is a minimal node separator, but it is not necessarily good, because it may contain nodes with incident backward edges—see Section 2.1. To make it good, we first identify the nodes that have incident backward edges and replace each of them with all the nodes that are reachable from them in one hop. Thus we advance the node separator toward the target keeping it minimal, and stop as soon as we encounter a good minimal node separator. This way, we aim to keep the size of E_i small. We repeat this process to select more good node separators until we reach the target.

3 RESULTS

In this section we experimentally evaluate our method. Section 3.1 presents the datasets and the experimental setup. Section 3.2 examines the running time of our algorithm. Section 3.3 presents the reachability profiles obtained with our method. Section 3.4 evaluates gene centrality based on the reachability profiles. Section 3.5 analyses the stability of the human TRN.

3.1 Datasets and implementation details

We evaluate our method using both synthetic and real biological networks.

Synthetic dataset. We generated the synthetic network dataset using the Barabasi–Albert random network model (Barabasi and Albert, 1999). We chose this model because it is the de facto standard for the scale-free networks, which best describe most biological networks (Jeong *et al.*, 2000; Todor *et al.*, 2012; Yook *et al.*, 2004). We created six sets of random networks. In each set, we created 10 networks with the same number of nodes: 50, 100, 150, 200, 250 and 300, respectively. The number of edges is twice the number of nodes in each network.

Real dataset. For experimentation on real biological networks we used the human regulatory network of Gerstein *et al.* (2012). From this network, we selected only the reliable interactions by taking the intersection with those present in the DIP database (Xenarios *et al.*, 2002). The resulting network has 130 nodes and 172 edges. To assess the interaction confidence for each edge in this intersection, we used the logistic regression method used by Sharan *et al.* (2002). This strategy is used often in the literature to compute interaction confidence (Bader *et al.*, 2004; Ourfali *et al.*, 2007; Sharan *et al.*, 2002; Shlomi *et al.*, 2006). We obtained the gene expression data of 575 leukemia patients from Zhang *et al.* (2012). We obtained control gene expression data in early progenitor cells from Laurenti *et al.* (2013). Both control and leukemia expression datasets are normalized using quantile normalization (Amaratunga and Cabrera, 2001). Each leukemia sample in our dataset belongs to one of eight different subtypes of leukemia: hyperdiploid, TCF3-PBX1, ETV6-RUNX1, MLL, Ph, Hypo, T-ALL and Other, or to non-leukemia sample types CD10CD19 and CD34. We do not include samples from the last two categories in our experiments, since they contain only four samples each. We trained eight different logistic regression models, one for each leukemia subtype to compute interaction probabilities for each subtype separately. Also, we classified the early progenitor cell samples into three categories: primitive (hematopoietic stem cells), lymphoid (ETP, MLP, ProB and B_NKpre) and myeloid (the rest of the samples). We trained a different logistic regression model for each type. Thus, we obtained different probability values for the edges of the human regulatory network, depending on the cancer or control group subtype in which the gene expression levels were measured. This in turn results in different reachability probabilities. We identified all the source and all the target genes in our network using the hierarchical decomposition obtained by HIDEN (Gulsoy *et al.*, 2012). This resulted in 9 source genes and 88 target genes.

We used C + +, Matlab and R for implementation. We ran our experiments on an AMD Opteron processor with 256 GB of memory and 1.9 GHz speed.

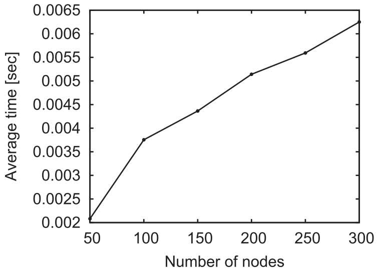

Fig. 4. Average running time of our method on Barabasi–Albert networks for growing network sizes

3.2 Evaluation of the running time

In order to evaluate the performance of our method systematically, we ran it on the synthetic networks of different sizes. We measured the running time for each synthetic network and each source–target pair. We have taken each node, in turn, as a source an then as a target. Thus, computing the reachability profile for the largest network size requires $300 \times 299 = 89700$ reachability probability computations per network. In total, we computed the reachability profile for $10 \times 6 = 60$ networks, for a total of 2264500 reachability probabilities. In Figure 4, we report the average running time to compute the reachability probability for one source–target pair for each set of networks. We report the average running time over all networks in the set and over all source–target pairs.

The figure shows that the running time of our method in a scale-free network grows at most linearly in terms of number of nodes. Even for networks as large as 300 nodes and 600 edges, the average running time of our method per source–target pair remains in milliseconds. *This small running time allows us to compute the entire reachability profiles in practical time for a large number of networks, which was not possible before.*

For comparison, the inclusion/exclusion method (Ourfali *et al.*, 2007) and PReach (Gabr *et al.* 2013) fail to complete execution on the same dataset because they exhausted the 256 GB of memory available in the system even for a single source–target node pair of the smallest network in our dataset.

For the real dataset investigated in this article, we computed 11 reachability profiles, one for each leukemia or control group subtype. For each subtype, we computed 9×88 reachability probabilities (for 9 sources and 88 target nodes), thus 8712 probabilities in total. Our method computed each of these

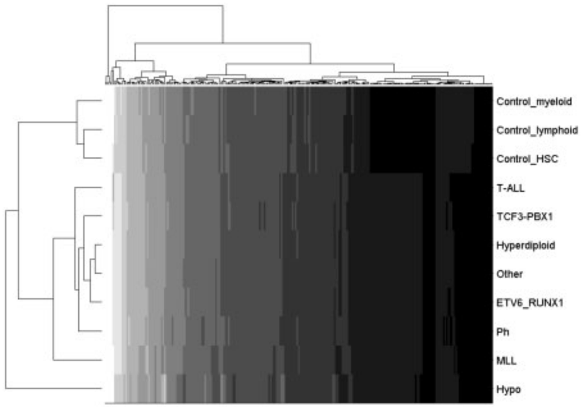

Fig. 5. Reachability profiles in the human regulatory network. Each row represents a cancer type or a control group. Each column represents a source–target pair. The intensity of each cell represents the reachability probability for that source–target pair—lighter color means higher probability

probabilities in only 2.5 s on the average. Both PReach and the inclusion–exclusion method fail to scale to this network size.

3.3 Reachability profiles in the human TRN

For each leukemia or control network, we computed the reachability probability for each pair of source–target nodes. We call this the reachability profile of the network. In Figure 5 we show the reachability profiles for all leukemia subtypes and control groups in a heat map. Each row in the figure represents a leukemia subtype or a control group, and each column represents a source–target pair. The color intensity at a location represents the reachability probability for that pair. We applied hierarchical clustering on both dimensions based on the reachability profiles. Hierarchical clustering correctly clusters the control groups subtypes together, as well as all the leukemias. This shows that the reachability profile can distinguish between healthy and leukemia cases.

Source and target gene groups that show a noticeable gap between their reachability probabilities in control versus leukemia cases include SPI1, POU2F2 as sources and TOPBP1, TFDP1, TFDP2, HDAC1, CDK8, REL, RELA and NFKB2 as targets. While these sources and targets have low a reachability probability for control groups, they exhibit a higher range in leukemia subtypes.

Our findings resonate with earlier observations. Our method clusters the hyperdiploid and the ETV6-RUNX1 subtypes together, while in (Zhang *et al.*, 2012), Supplementary Figure S22, a significant number of genes exhibit similar expression levels in these subtypes. They are frequently studied together, as they are both related to a favorable prognosis in children (Liang *et al.*, 2010; Paulsson *et al.*, 2010). On the contrary, the Hypo subtype, which is least similar to Hyperdiploid and ETV6-RUNX1 in our results, is associated with poor outcome (Holmfeldt *et al.*, 2013).

To further appreciate the value added by the reachability profiles to our results, we performed another experiment based solely on gene expression data, without taking the regulatory network into account. In this experiment, we clustered the gene expression samples using k-means clustering. We set $k = 11$, as there are totally 11 subtypes in our dataset. Then, within each cluster, we examined the distribution of each leukemia type. The results are shown in Figure 6. Our results demonstrate that, with the exception of cluster 10, consisting primarily of T-ALL samples, all the clusters are a heterogeneous mix and do not have a definitive dominant leukemia type. Although one cluster consists only of control samples, the control subtypes are mixed together. Furthermore, the myeloid subtype samples are spread out through the rest of the clusters. *We conclude that clustering based on gene expression alone is insufficient for classifying leukemia types.*

In light of these experimental observations, reachability profiles prove to be a reliable and valuable tool for assessing the viability of TRNs working as a whole.

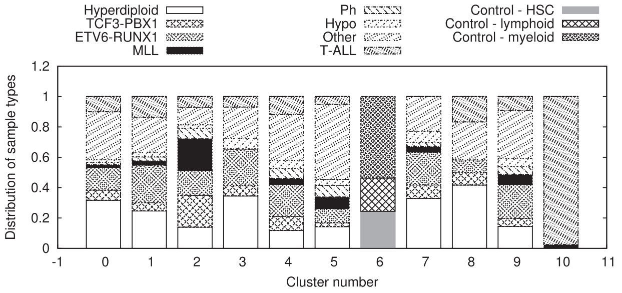

Fig. 6. Distribution of leukemia subtype and control group samples within clusters obtained from transcription data alone. Each cluster is normalized by the number of samples it contains

3.4 Gene centrality using reachability profiles

We further illustrate the usefulness of reachability profiles by analysing the centrality of genes based on their contribution toward the reachability profile (Gabr and Kahveci, 2013). For this experiment, we compare the reachability profiles for the original network with the reachability profiles obtained by eliminating one gene from the network. Thus, for each gene, we compute its centrality by comparing the reachability profile for the original network with the reachability profile obtained when the gene is missing. For a given gene g, whose centrality is under consideration, and a given source–target pair, the difference in reachability probability can be seen as the probability that the source–target pair is indispensable for connecting the source to the target; in other words, $\{g\}$ is a node separator. Then the sum of this value over all source–target pairs is the average number of source–target pairs for which g is indispensable. To formalize this description, let us denote the set of source and target genes with S and T, respectively. We also denote the probability that gene t is reachable from gene s in the original network with $p(s,t)$ and the same probability for the network where gene g is removed with $p_{\overline{g}}(s, t)$. The centrality of gene g is defined as $\sum_{s\in S}\sum_{t\in T} p(s, t) - p_{\overline{g}}(s, t)$.

Figure 7 plots the centrality values for each leukemia type and each gene. We excluded from the plot the genes having centrality smaller than 1. As expected, only a few genes have a high centrality, which is a characteristic of scale-free networks. We also

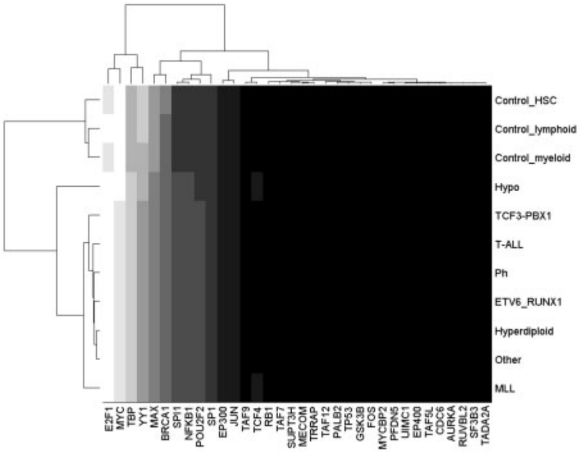

Fig. 7. Centrality of genes in the human regulatory network for different leukemia subtypes. Light color denotes high centrality. We only show the genes with centrality value >1 for at least one network

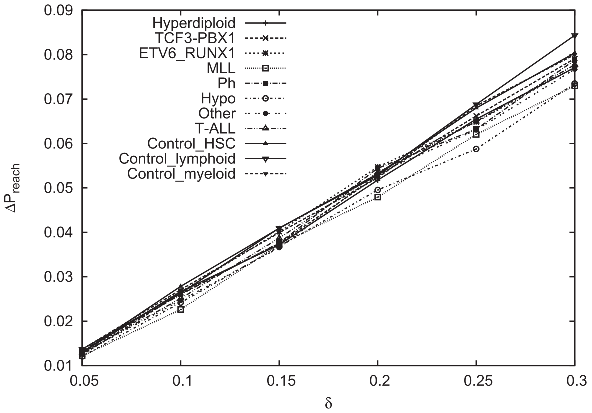

Fig. 8. Effect of random perturbation as a measure of network stability: average change in reachability probability (ΔP_{reach}) when each interaction probability p is altered to a random value in the window $p \pm \delta \cap [0,1]$

performed hierarchical clustering of the leukemia subtypes and of the genes based on their centrality. We observe that the most similar subtypes of leukemias are T-ALL and Ph. The Ph subtype is a chromosomal abnormality resulting from the same translocation found in ALL (Talpaz *et al.*, 2006). The least similar to the first two is Hypo, like in the reachability profiles experiment. TP53 and RB1 are two of the most central genes identified by our method. They are both characterized by alterations in Hypodiploid ALL (Holmfeldt *et al.*, 2013). We see that the most central gene is E2F1, which a transcription factor known to have a crucial role in cell cycle and tumor suppression (Neuman *et al.*, 1996). Thus, malfunctioning of this gene severely affects many pathways in the regulatory network. Likewise, the following two reachable genes, MYC and TBP are known hubs regulating important functions. MYC is involved in cell proliferation and its persistent expression is common to many cancers (Nesbit *et al.*, 1999), while TBP is related to RNA polymerase II, an essential element of DNA transcription initiation (Kornberg, 2007). Among the top genes we identified based on their centrality is also EP300, a histone-modifying gene which was reported to inactivate lesions disrupting hematopoietic development in ETP ALL (Zhang *et al.*, 2012).

3.5 Assessment of network stability

Beside characterization of single genes using centrality, we also performed and experiment to characterize the entire human TRN. In this experiment, we assess the level of stability of each of the studied networks. We measure the stability of the network as the average change in reachability probability when edge probabilities are randomly perturbed.

Consider the given probabilistic network $\mathcal{G} = (V, E, P)$ and the sets of source and target nodes S and T. Also consider a parameter δ that denotes the maximum change in edge probabilities. We defined a perturbed edge probability function $P^{\delta} \rightarrow [0,1]$ that, for each edge $e \in E$, returns a value drawn uniformly at random from the range $P(e) \pm \delta \cap [0, 1]$. We constructed a perturbed network $G^{\delta} = (V, E, P^{\delta})$. For every pair of $s \in S$ and $t \in T$, we measured the reachability probability in G as $p(s,t)$, as well as that in G^{δ} as $p^{\delta}(s,t)$. We then computed the absolute difference $|p^{\delta}(s, t) - p(s, t)|$. We repeated this experiment 20 times. We computed the average of the resulting values over all $s \in S$ and $t \in T$, as well as over the 20 experiments.

We plotted the results for different values of δ for the leukemia networks as well as for the control networks. Figure 8 shows the results. The first observation we draw from the figure is that the change in reachability probability for all networks is linear. We also observe that even by perturbing the edge probabilities in the range of ±0.3, the change in reachability probability does not exceed 0.1. From these two observations we can judge transcription-factor regulation in homo sapiens as highly stable and insensitive to random perturbation. This conclusion holds for both healthy people and leukemia patients. However, we also observe that the gap between the networks is not constant; it slightly increases with the increase of perturbation level. At the extreme case of ±0.3, the gap is maximum. There, T-ALL show the most sensitivity to this level of perturbation, while Hypo and MLL show the least sensitivity.

4 CONCLUSION

In this article we have characterized different types of leukemias based on the state of the regulatory networks in patients affected by this disease. The state is evaluated through reachability profiles. The reachability profile describes the ability of regulator genes to affect the transcription factors. For this we developed a fast, exact method for computing the probability for a signal to reach from a source node to a destination node in a probabilistic network. The rigorous mathematical apparatus, which involves polynomials and polynomial collapsing operators, allows fast execution time, demonstrated in the performance evaluation experiments. Valuable uses of the reachability profiles illustrated in this article include leukemia subtype classification, gene centrality evaluation and regulatory network stability analysis. All these are valuable tools for evaluating the viability of the TRN under varying conditions as a whole, not just limited to individual gene expressions levels. An interesting parallel can be drawn between our solution and Bayesian Network inference. However, as we mentioned in Section 1, this alternative is limited to acyclic networks. We see a possible application of the Bayesian Network alternative in combination with the reduction of strongly connected components to single nodes, but this solution deserves a careful examination by itself.

Funding: NSF CCF-1251599, NSF DBI-1262451, NSF IIS-084-5439.

Conflict of Interest: none declared.

REFERENCES

Aggarwal,K.K. *et al.* (1975) Reliability evaluation: a comparative study of different techniques. *Microelectron. Reliab.*, **14**, 49–56.

Amaratunga,D. and Cabrera,J. (2001) Analysis of data from viral DNA microchips. *J. Am. Stat. Assoc.*, **96**, 1161–1170.

Armstrong,S.A. *et al.* (2002) MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nat. Genet.*, **30**, 41–47.

Bader,J.S. *et al.* (2004) Gaining confidence in high-throughput protein interaction networks. *Nat. Biotechnol.*, **22**, 78–85.

Barabasi,A.-L. and Albert,R. (1999) Emergence of scaling in random networks. *Science*, **286**, 509–512.

Brown,J.I. and Colbourn,C.J. (1996) Non-stanley bounds for network reliability. *Journal of Algebraic Combinatorics*, **5**, 13–36.

Bu,Z. and Callaway,D.J. (2011) Proteins move! Protein dynamics and long-range allostery in cell signaling. *Adv. Protein Chem. Struct. Biol.*, **83**, 163–221.

Ceol,A. *et al.* (2010) MINT, the Molecular INTeraction database: 2009 update. *Nucleic Acids Res.*, **38**, D532–D539.

Deng,M. *et al.* (2003) Assessment of the reliability of protein-protein interactions and protein function prediction. *Pac. Symp. Biocomp.*, 140–151.

Gabr,H. and Kahveci,T. (2013) Characterization of probabilistic signaling networks through signal propagation. In: *CISE Technical Report REP-2013-567*. University of Florida.

Gabr,H. *et al.* (2013) PReach: reachability in probabilistic signaling networks. *ACM-BCB*.

Gerstein,M.B. *et al.* (2012) Architecture of the human regulatory network derived from ENCODE data. *Nature*, **489**, 91–100.

Gulsoy,G. *et al.* (2012) HIDEN: hierarchical decomposition of regulatory networks. *BMC Bioinformatics*, **13**, 250.

Holmfeldt,L. *et al.* (2013) The genomic landscape of hypodiploid acute lymphoblastic leukemia. *Nat. Genet.*, **45**, 242–252.

Husfeldt,T. and Taslaman,N. (2010) The exponential time complexity of computing the probability that a graph is connected. *International Symposium on Parameterized and Exact Computation*, **6478**, 192–203.

Hwang,G.L. *et al.* (1981) System-reliability evaluation techniques for complex/large systems – a review. *IEEE Trans. Reliab.*, **R-30**, 416–423.

Jeong,H. *et al.* (2000) The large-scale organization of metabolic networks. *Nature*, **407**, 651–654.

Kornberg,R.D. (2007) The molecular basis of eukariotic transcription. *Proc. Natl Acad. Sci. USA*, **104**, 12955–12961.

Krivtsov,A.V. *et al.* (2009) Gene expression profiling of leukemia stem cells. *Methods Mol. Biol.*, **538**, 231–246.

Laurenti,E. *et al.* (2013) The transcriptional architecture of early human hematopoiesis identifies multilevel control of lymphoid commitment. *Nat. Immunol.*, **14**, 756–763.

Liang,D.C. *et al.* (2010) Frequencies of ETV6-RUNX1 fusion and hyperdiploidy in pediatric acute lymphoblastic leukemia are lower in far east than west. *Proc. Natl Acad. Sci. USA*, **55**, 430–433.

Los,M. *et al.* (2009) Switching Akt: from survival signaling to deadly response. *Bio Essays*, **31**, 492–495.

Mattick,J.S. *et al.* (2009) RNA regulation of epigenetic processes. *Bio Essays*, **31**, 51–59.

Nesbit,C.E. *et al.* (1999) Prochownic. MYC oncogenes and human neoplastic disease. *Oncogene*, **18**, 3004–3016.

Neuman,E.S. *et al.* (1996) Structure and partial genomic sequence of the human E2F1 gene. *Gene*, **173**, 163–169.

Ourfali,O. *et al.* (2007) SPINE: a framework for signaling-regulatory pathway inference from cause-effect experiments. *Bioinformatics*, **23**, i359–i366.

Paulsson,K. *et al.* (2010) Genetic landscape of high hyperdiploid childhood acute lymphoblastic leukemia. *Proc. Natl Acad. Sci. USA*, **107**, 21719–21724.

Provan,J.S. and Ball,M.O. (1983) The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM J. Comput.*, **12**, 777–788.

Sharan,R. *et al.* (2002) Conserved patterns of protein interaction in multiple species. *PNAS*, **102**, 1974–1979.

Shlomi,T. *et al.* (2006) QPath: a method for querying pathways in a protein-protein interaction network. *BMC Bioinformatics*, **7**, 199.

Suthram,S. *et al.* (2006) A direct comparison of protein interaction confidence assignment schemes. *BMC Bioinformatics*, **7**, 360.

Szklarczyk,D. *et al.* (2011) The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored. *Nucleic Acids Res.*, **39**, D561–D568.

Talpaz,M. *et al.* (2006) Dasatinib in imatinib-resistant Philadelphia chromosome-positive leukemias. *N. Engl. J. Med.*, **354**, 2531–2541.

Todor,A. *et al.* (2012) Uncertain interactions affect degree distribution of biological networks. *BIBM*.

Todor,A. *et al.* (2013) Probabilistic biological network alignment. *TCBB*, **10**, 109–121.

Valk,P.J. *et al.* (2004) Prognostically useful gene-expression profiles in acute myeloid leukemia. *N. Engl. J. Med.*, **350**, 1617–1628.

van Lint,J.H. and Wilson,R.M. (1992) *A Course in Combinatorics*. Cambridge University Press, Cambridge, UK.

von Mering,C. *et al.* (2002) Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, **417**, 399–403.

Xenarios,I. *et al.* (2002) DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Res.* **30**, 303–305.

Yook,S.H. *et al.* (2004) Functional and topological characterization of protein interaction networks. *Proteomics*, **4**, 928–942.

Zhang,J. *et al.* (2012) The genetic basis of early T-cell precursor acute lymphoblastic leukaemia. *Nature*, **481**, 157–163.