

Common Attractors in Multiple Boolean Networks

Yu Cao, Wenya Pi, Chun-Yu Lin, Ulrike Münzner, Masahiro Ohtomo, Tatsuya Akutsu, *Senior Member, IEEE*

Abstract—Analyzing multiple networks is important to understand relevant features among different networks. Although many studies have been conducted for that purpose, not much attention has been paid to the analysis of attractors (i.e., steady states) in multiple networks. Therefore, we study common attractors and similar attractors in multiple networks to uncover hidden similarities and differences among networks using Boolean networks (BNs), where BNs have been used as a mathematical model of genetic networks and neural networks. We define three problems on detecting common attractors and similar attractors, and theoretically analyze the expected number of such objects for random BNs, where we assume that given networks have the same set of nodes (i.e., genes). We also present four methods for solving these problems. Computational experiments on randomly generated BNs are performed to demonstrate the efficiency of our proposed methods. In addition, experiments on a practical biological system, a BN model of the TGF- β signaling pathway, are performed. The result suggests that common attractors and similar attractors are useful for exploring tumor heterogeneity and homogeneity in eight cancers.

Index Terms—Boolean networks, attractors, multi-layer networks, TGF- β signaling pathway.

1 INTRODUCTION

Analysis of biological networks plays a critical role in understanding the underlying mechanisms in many fields including bioinformatics, systems biology, and neuroscience. To execute computational analysis of biological networks in a holistic manner, various mathematical models have been proposed and utilized, among which the Boolean network (BN) has been one of the most well-studied models. The BN is a discrete mathematical model of genetic networks [1], [2], and neural networks [3], in which each node takes either 0 (inactive) or 1 (active) at each time step, and the states of nodes change synchronously or asynchronously according to regulation rules given as Boolean functions. Although BN is a simple model with automated generation tools available [4], it captures significant dynamic properties of the modeled system [5], and can contribute to our understanding of disease mechanisms [6], [7]. Therefore, various studies have been done on BNs. In particular, extensive studies have recently been done on identification and control of BNs and related models [8], [9], [10] using the semi-tensor product [11], [12] and similar approaches. Among various studies, attractors gathered special interest of researchers. An *attractor* is a set of the system's states satisfying that any two states in this set can be reached from each other. An attractor is called a *singleton attractor* if it consists of only one state, otherwise it is called a *periodic attractor*. As stable

states of the system, attractors in gene regulatory networks are hypothesized to characterize cellular phenotypes [1]. Therefore, extensive studies have been done for detecting and analyzing attractors in BNs [13], [14], [15], [16], [17], [18], [19].

Compared to extensively studied analysis of characteristic properties on BNs and other network models, comparison of multiple networks has not yet been addressed in the literature so far except the work on control (but not detection) of singleton attractors in two BNs [20]. Since there exist differences in the structure of genetic networks not only between individuals but also between different cells [21], research on similarities and differences among networks could provide important insights of biological systems [22]. Of course, there exists a tool to examine the robustness of attractors against random perturbations (e.g., see Section 3.5 of [23]). However, our purpose is not to examine the robustness but to examine the similarities and differences among specific cells (e.g., normal and cancer cells). Therefore, we study the detection of common and similar attractors in multiple BNs focusing on singleton attractors. Since we consider networks in different types of cells in the same organism, we assume that the sets of nodes in the input networks are the same. Even if there exist some genes not appearing in other networks, the corresponding nodes in those networks can be regarded as isolated ones (i.e., nodes with self loops only).

In this paper, we propose three problems, ComAttBN, SimAttBN, and SimAttWBN, to detect common attractors in BNs, similar attractors in BNs, and similar attractors in node-weighted BNs, respectively. Next, we theoretically analyze the expected number of common and similar attractors. Then, we propose four methods for solving these three problems using recursive algorithms and integer linear programming. To evaluate the efficiency of our proposed

Y. Cao, W. Pi, M. Ohtomo, T. Akutsu are with Bioinformatics Center, Institute for Chemical Research, Kyoto University, Kyoto 611-0011, Japan. (e-mail: {caoyu, piwenya, ohtomo, takutsu}@kuicr.kyoto-u.ac.jp).

Y. Cao and W. Pi are the joint first authors.

C-Y. Lin is with Institute of Bioinformatics and Systems Biology, National Yang Ming Chiao Tung University, Hsinchu 300, Taiwan, and Center for Intelligent Drug Systems and Smart Bio-devices, National Yang Ming Chiao Tung University, Hsinchu 300, Taiwan. (e-mail: chunyu.lin@nctu.edu.tw).

U. Münzner was with the Laboratory of Cell Systems, Institute for Protein Research, Osaka University, Suita, Japan. (e-mail: available upon request).

methods as well as to demonstrate the usefulness of the proposed approach for casting new light on the understanding of similarities or differences among multiple networks across eight cancer types/subtypes, we perform computational experiments on random BNs and on a BN model of the TGF- β signaling pathway. Note that some figures are given as Supplemental Material (PDF), and the main codes and data are also given as Supplemental Material (tar file).

2 PRELIMINARIES

A Boolean network (BN) $G(V, F)$ consists of a set of nodes $V = \{v_1, \dots, v_n\}$ and a set of Boolean functions $F = \{f_1, \dots, f_n\}$, where each node takes the value of either 0 (inactive) or 1 (active) at each discrete time step t . In this paper, we consider the synchronous BN, in which the state of each node at time $t + 1$ is determined by the state of other nodes (and/or itself) at time t . We use $v_i(t)$ to denote the state of node v_i at time t , and k_i to denote the number of input nodes of v_i , which is called the *indegree* of v_i . We use $\mathbf{v}(t) = [v_1(t), \dots, v_n(t)]$ to denote the state of the whole network at time t . A Boolean function f_i is assigned to each v_i and the state of v_i at time $t + 1$ is determined by $v_i(t + 1) = f_i(v_{i_1}(t), \dots, v_{i_{k_i}}(t))$, where $\{v_{i_1}, \dots, v_{i_{k_i}}\}$ is the set of input nodes to v_i . We also use $v_i(t + 1) = f_i(\mathbf{v}(t))$ to denote this function. Then, the state of the whole BN at time $t + 1$ is determined by $\mathbf{v}(t + 1) = \mathbf{f}(\mathbf{v}(t))$, where $\mathbf{f}$ is a vector of Boolean functions $f_1, \dots, f_n$.

The network structure of $G(V, F)$ is represented by a directed graph on the set node nodes V in which there exists a directed edge (v_j, v_i) if and only if v_j is an input node to v_i . The *maximum indegree* over all nodes is denoted by K , which has a strong correlation with the complexity of BN and significantly affects the computational complexity in many BN problems [13]. The dynamics of a BN is well-described by a *state transition diagram* in which a node corresponds to a state of the BN and there exists an edge from a node corresponding to $\mathbf{x}$ to a node corresponding to $\mathbf{y}$ if and only if $\mathbf{y} = \mathbf{f}(\mathbf{x})$. If a state $\mathbf{x}$ satisfies $\mathbf{x} = \mathbf{f}(\mathbf{x})$, it is called a *singleton attractor*. If a sequence of states $(\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^p)$ satisfies $\mathbf{x}^{i+1} = \mathbf{f}(\mathbf{x}^i)$ for $i = 1, \dots, p - 1$ and $\mathbf{x}^1 = \mathbf{f}(\mathbf{x}^p)$ for $p > 1$, it is called a *periodic attractor*.

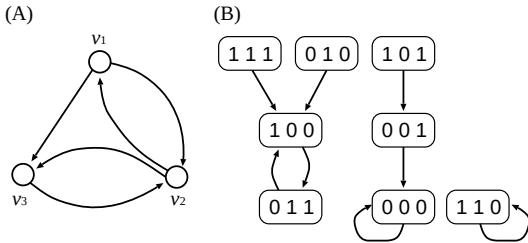


Fig. 1. (A) Network structure of BN $G_1(V, F_1)$. (B) State transition diagram of G_1 .

Example 1. The network structure and the state transition diagram of the following BN ($G_1(V, F_1)$) are given in Fig. 1 (A) and Fig. 1 (B), respectively, where $V = \{v_1, v_2, v_3\}$ and $F = \{v_1(t + 1) = v_2(t), v_2(t + 1) = v_1(t) \wedge v_3(t), v_3(t + 1) = v_1(t) \wedge v_2(t)\}$. This BN has two singleton attractors $[0, 0, 0]$ and $[1, 1, 0]$ and one periodic attractor $([0, 1, 1], [1, 0, 0])$.

3 PROBLEM DEFINITIONS

As mentioned in Section 1, we consider three problems, ComAttBN, SimAttBN, and SimAttWBN. First, we define a basic problem, ComAttBN.

Definition 1. [ComAttBN]

Instance: BNs $G_1, \dots, G_N$ with the same set of nodes $V = \{v_1, \dots, v_n\}$,

Problem: decide whether or not there exists a singleton attractor common to $G_1, \dots, G_N$ and output one if it exists.

Example 2. Consider BN $G_1(V, F_1)$ given in Example 2.1. Suppose that two more BNs $G_2(V, F_2)$ and $G_3(V, F_3)$ are given as $F_2 = \{v_1(t + 1) = v_2(t) \vee v_3(t), v_2(t + 1) = v_1(t) \vee v_3(t), v_3(t + 1) = v_2(t) \wedge v_3(t)\}$, and $F_3 = \{v_1(t + 1) = v_2(t), v_2(t + 1) = v_1(t) \wedge v_3(t), v_3(t + 1) = v_1(t)\}$. G_1 has two singleton attractors: $[0, 0, 0]$ and $[1, 1, 0]$, G_2 has three singleton attractors: $[0, 0, 0]$, $[1, 1, 0]$, and $[1, 1, 1]$, and G_3 has two singleton attractors: $[0, 0, 1]$ and $[1, 1, 0]$. Then, the solution of ComAttBN(G_1, G_2, G_3) is $[1, 1, 0]$.

It is not plausible that many BNs share exactly the same attractors. Furthermore, to be shown in the next section, the expected number of common singleton attractors among random BNs is quite few. Therefore, it is reasonable to consider similar attractors among multiple BNs, which might also be useful to reveal hidden features in multiple BNs. Before defining the detection problem, we need to define the *similarity* among multiple attractors. Hereafter, $\mathbf{x}_i^j$ denotes the i -th element (i.e., i -th bit) of a vector $\mathbf{x}^j$.

Definition 2. [Similarity of attractors]

The similarity of N singleton attractors $\mathbf{x}^1, \dots, \mathbf{x}^N$ is defined as $s(\mathbf{x}^1, \dots, \mathbf{x}^N) = |\{i | \mathbf{x}_i^1 = \dots = \mathbf{x}_i^N, i = 1, \dots, n\}|$.

In other words, the similarity is defined as the number of bits that are common to all given attractors. A tuple of singleton attractors $(\mathbf{x}^1, \dots, \mathbf{x}^N)$ is called a *similar attractor tuple* if its similarity is no less than the given similarity measure. Now we define the detection problem of similar attractors.

Definition 3. [SimAttBN]

Instance: N BNs $G_1, \dots, G_N$ with the same set of nodes $V = \{v_1, \dots, v_n\}$, and an integer S ,

Problem: decide whether or not there exists a similar attractor tuple with similarity $s \geq S$ for $G_1, \dots, G_N$ and output one if it exists.

Example 3. Consider a BN $G_4(V, F_4)$ with singleton attractors $[0, 0, 0, 1]$ and $[1, 1, 0, 1]$, and a BN $G_5(V, F_5)$ with singleton attractors $[0, 0, 1, 0]$ and $[0, 1, 0, 0]$. Suppose that $S = 2$ is given as the similarity measure. Then, there exist three similar attractor tuples for G_4 and G_5 : $([0, 0, 0, 1], [0, 0, 1, 0])$, $([0, 0, 0, 1], [0, 1, 0, 0])$, $([1, 1, 0, 1], [0, 1, 0, 0])$.

In SimAttBN, every node in a BN is equivalently considered in the measurement of similarity among singleton attractors. However, distinct nodes (e.g., distinct genes) in practical networks usually play distinct roles thus have different significance. For instance, the output nodes in the Boolean model of TGF- β (see Section 6.2) are considered to be more critical compared to other nodes for deciding

the cellular phenotype. Therefore, we give a definition of a weighted BN to reflect the importance of respective nodes.

Definition 4. [Weighted Boolean network]

Let $w = [w_1, \dots, w_n]$ be a vector of weights. A weighted Boolean network $G(V, F, w)$ is a Boolean network in which each node v_i has a weight w_i assigned to it.

According, the similarity for attractors in weighted BNs is re-defined as below.

Definition 5. [Weighted similarity of attractors]

The weighted similarity of N singleton attractors $\mathbf{x}^1, \dots, \mathbf{x}^N$ is defined by $s_w(\mathbf{x}^1, \dots, \mathbf{x}^N) = \sum_{\{i | \mathbf{x}_i^1 = \dots = \mathbf{x}_i^N, i=1, \dots, n\}} w_i$.

Based on this similarity measure, the detection problem of similar singleton attractors in weighted BNs is defined as follows.

Definition 6. [SimAttWBN]

Instance: N weighted Boolean networks $G_1, \dots, G_N$ with the same set of nodes $V = \{v_1, \dots, v_n\}$ and the same weight vector $w = [w_1, \dots, w_n]$, and a positive real number W ,

Problem: decide whether or not there exists a singleton attractor tuple with weighted similarity $s_w \geq W$ for $G_1, \dots, G_N$ and output one if it exists.

Example 4. Consider weighted BNs, $G_6(V, F_6, w)$ with singleton attractors $\{[0, 0, 0], [1, 1, 0], [1, 1, 1]\}$ and $G_7(V, F_7, w)$ with one singleton attractor $[0, 1, 0]$, where $w = [0.3, 0.1, 0.5]$. Suppose $W = 0.7$ is given. Then, $\text{SimAttWBN}(G_6, G_7, W)$ has a unique solution $([0, 0, 0], [0, 1, 0])$ with $s_w = 0.8$.

4 EXPECTED NUMBER OF COMMON ATTRACTORS

In this section, we theoretically analyze the expected number of common attractors and similar singleton attractors with assuming random BNs. The first result is on the expected number of common attractors, which is an extension of the result on the expected number of singleton attractors in a random BN [24], [25]. Note that in a random BN with the maximum indegree K , for each node, K input nodes are randomly selected and then a Boolean function is randomly selected from 2^{2^K} possible functions, where this assignment procedure is done independently for all nodes. The results in this section hold for random BNs with any fixed K , including the case of $K = n$.

Proposition 1. The expected number of common singleton attractors for N random BNs is $\frac{1}{2^{n(N-1)}}$.

Proof. Suppose that N BNs with the same set of nodes $G_1, \dots, G_N$ are given. Here, we define the state of i -th node of G_j at time t by $v_i^j(t)$, and the state of G_j at time t by $\mathbf{v}^j(t)$. Assume without loss of generality (w.l.o.g.) that the initial state is $\mathbf{v}^1(0) = \dots = \mathbf{v}^N(0) = [0, \dots, 0]$. Therefore in G_1 , the probability that $v_i^1(1) = 0$ is $\frac{1}{2}$ because the Boolean function assigned to v_i^1 is randomly selected thus the probability that $v_i^1(1) = 0$ (resp., $v_i^1(1) = 1$) is $\frac{1}{2}$ regardless of input node values. Since the Boolean function assigned to each node of G_1 is independently selected, the probability that $\mathbf{v}^1(1) = [0, \dots, 0]$ is $\frac{1}{2^n}$, thus the probability that $[0, \dots, 0]$ is a singleton attractor of G_1 is $\frac{1}{2^n}$. Similarly, for each BN, the probability that $[0, \dots, 0]$ is a singleton

attractor of G_j is $\frac{1}{2^n}$. Therefore, the probability that $[0, \dots, 0]$ is a common singleton attractor is $\frac{1}{2^{nN}}$. Since there are 2^n possible initial states, from the linearity of expectation, the expected number of common singleton attractors is given by $\frac{1}{2^{nN}} \cdot 2^n = \frac{1}{2^{n(N-1)}}$. $\square$

The next result is on the expected number of similar attractor tuples and is based on novel and non-trivial ideas.

Theorem 1. The expected number of similar attractor tuples with similarity S for N random BNs is $\frac{\binom{n}{S}(2^{N-1}-1)^{n-S}}{2^{n(N-1)}}$.

Proof. Suppose that N BNs with the same set of nodes $G_1, \dots, G_N$ are given. Assume G_j has a singleton attractor $\mathbf{x}^j$. Consider an n -dimensional vector $\mathbf{x} = [0, \epsilon, \dots, \epsilon]$ where ϵ represents a bit that can be either 0 or 1, and thus $\mathbf{x}$ has 2^{n-1} different forms. Since we are considering random BNs, the probability that $[0, 0, \dots, 0]$ is a singleton attractor of G_1 is $\frac{1}{2^n}$. From the linearity of expectation, the probability that $\mathbf{x}$ is a singleton attractor of G_1 is $\frac{1}{2^n} \cdot 2^{n-1} = \frac{1}{2}$, thus the probability that $\mathbf{x}_1^1 = 0$ is $\frac{1}{2}$. Similarly, for each BN G_j , the probability that $\mathbf{x}_1^j = 0$ is $\frac{1}{2}$. Hence the probability that $\mathbf{x}_1^1 = \dots = \mathbf{x}_1^N = 0$ is $\frac{1}{2^N}$. Then as well, the probability that $\mathbf{x}_1^1 = \dots = \mathbf{x}_1^N = 1$ is $\frac{1}{2^N}$. Therefore,

$$\begin{aligned} \text{Prob}(\mathbf{x}_1^1 = \dots = \mathbf{x}_1^N) &= \text{Prob}(\mathbf{x}_1^1 = \dots = \mathbf{x}_1^N = 0) \\ &\quad + \text{Prob}(\mathbf{x}_1^1 = \dots = \mathbf{x}_1^N = 1) \\ &= \frac{1}{2^N} + \frac{1}{2^N} = \frac{1}{2^{N-1}}. \end{aligned}$$

From the complement rule for probability,

$$\text{Prob}(\overline{\mathbf{x}_1^1 = \dots = \mathbf{x}_1^N}) = 1 - \frac{1}{2^{N-1}}.$$

Similarly, for each node v_i , $\text{Prob}(\mathbf{x}_i^1 = \dots = \mathbf{x}_i^N) = \frac{1}{2^{N-1}}$ and $\text{Prob}(\overline{\mathbf{x}_i^1 = \dots = \mathbf{x}_i^N}) = 1 - \frac{1}{2^{N-1}}$. Assume $I = \{1, 2, \dots, S\}$ and $\bar{I} = \{S+1, S+2, \dots, n\}$ where $S < n$. Then,

$$\begin{aligned} \text{Prob}(\mathbf{x}_i^1 = \dots = \mathbf{x}_i^N \mid i \in I) &= \left(\frac{1}{2^{N-1}}\right)^S, \\ \text{Prob}(\overline{\mathbf{x}_i^1 = \dots = \mathbf{x}_i^N} \mid i \in \bar{I}) &= \left(1 - \frac{1}{2^{N-1}}\right)^{n-S}. \end{aligned}$$

Thus, the probability that $s(\mathbf{x}^1, \dots, \mathbf{x}^N) = S$ where $\mathbf{x}_i^1 = \dots = \mathbf{x}_i^N$ for $i \in I$ is

$$\left(\frac{1}{2^{N-1}}\right)^S \cdot \left(1 - \frac{1}{2^{N-1}}\right)^{n-S} = \frac{(2^{N-1}-1)^{n-S}}{2^{n(N-1)}}.$$

Since there are $\binom{n}{S}$ possible combinations of choosing S nodes from n nodes, from the linearity of expectation,

$$\begin{aligned} \text{Prob}(s(\mathbf{x}^1, \dots, \mathbf{x}^N) = S) &= \frac{(2^{N-1}-1)^{n-S}}{2^{n(N-1)}} \cdot \binom{n}{S} \\ &= \frac{\binom{n}{S}(2^{N-1}-1)^{n-S}}{2^{n(N-1)}}. \end{aligned}$$

It has been proved that the expected number of singleton attractors in a random Boolean network is 1. Therefore, the expected number of similar attractor tuples is given by

$$1 \cdot 1 \cdot \dots \cdot 1 \cdot \frac{\binom{n}{S}(2^{N-1}-1)^{n-S}}{2^{n(N-1)}} = \frac{\binom{n}{S}(2^{N-1}-1)^{n-S}}{2^{n(N-1)}}.$$

$\square$

5 ALGORITHMS

The problems defined in Section 3 can be solved by a trivial approach: Separately enumerating singleton attractors for given BNs. However, the number of singleton attractors is $O(2^n)$ in the worst case and may also be large in practical networks, although the expected number for a random BN is small [24], [25]. Furthermore, enumeration of singleton attractors is known to be computationally difficult (#P-hard) [26]. Therefore, we need to develop efficient algorithms for solving the three problems defined in Section 3. Indeed, to be shown in theoretical analysis of the average case time complexity for random BNs for the first problem, the algorithm for two BNs is faster than that for one BN. This result suggests that it is worthy to develop algorithms for detecting common and similar attractors for multiple BNs because the trivial approach cannot provide such an interesting theoretical result.

5.1 RecComAttBN

For ComAttBN, we develop two algorithms, RecComAttBN and IlpComAttBN. In this subsection, we present RecComAttBN (Algorithm 1), which is an extension of the recursive algorithm for enumeration of singleton attractors for a single BN [18].

Algorithm 1 RecComAttBN: Recursive algorithm for common attractor enumeration

Input: N Boolean Networks $G_1(V, F_1), \dots, G_N(V, F_N)$

Output: all common singleton attractors

Initialize $i := 1$ and $\mathbf{x} := [\epsilon, \dots, \epsilon]$

procedure RecComAttBN($\mathbf{x}, i$)

if $i = n + 1$ **then**

 output $\mathbf{x}$

return

end if

for $b = 0$ **to** 1 **do**

$x_i \leftarrow b$

if it is found that $f_k^j(\mathbf{x}) \neq x_k$ for some $k \leq i$ and $j \in \{1, \dots, N\}$ **then**

continue

else

 RecComAttBN($\mathbf{x}, i + 1$)

end if

end for

return

The basic idea of RecComAttBN is to build a system state in a node-by-node manner by checking that each node's update function is consistent with the current state in every one of the models. For example, consider the case of $n = 3$. Then, it examines the system states in the following order: $[0, \epsilon, \epsilon]$, $[0, 0, \epsilon]$, $[0, 0, 0]$, $[0, 0, 1]$, $[0, 1, \epsilon]$, $[0, 1, 0]$, $[0, 1, 1]$, $[1, \epsilon, \epsilon]$, $\dots$, where ϵ means that the value of this element is not yet determined. Once it is found that some function outputs the value different from the current state, the states obtained by replacing (some of) ϵ with 0 and 1 are skipped. For example, if inconsistency is found at $[0, 0, \epsilon]$, it next examines $[0, 1, \epsilon]$ with skipping $[0, 0, 0]$ and $[0, 0, 1]$.

As explained above, RecComAttBN outputs all common singleton attractors in multiple BNs by extending a partial

global state $\mathbf{x} = [x_1, \dots, x_n]$ one by one in a recursive manner. Here, x_i denotes the value of node v_i in a common singleton attractor and $x_i \in \{0, 1, \epsilon\}$. Denote the Boolean function assigned to node v_i in G_j by f_i^j . At the i -th recursion level of RecComAttBN where $i \leq n$, values of the first $i-1$ nodes (i.e., $x_1, x_2, \dots, x_{i-1}$) are already determined and the algorithm extends $\mathbf{x}$ by letting $x_i = 0$ first. If either $f_k^j(\mathbf{x}) = x_k$ holds or its value is not yet determined for each $k = 1, \dots, i$ and each $j = 1, \dots, N$, then the algorithm proceeds to the next recursion level. Otherwise, the algorithm lets $x_i = 1$ and executes a similar procedure. After examining both $x_i = 0$ and $x_i = 1$, the algorithm returns to the previous recursion level. If the recursion level reaches $n + 1$, the algorithm outputs the current vector and then returns to the previous recursion level.

Here, we approximately analyze the average case time complexity of RecComAttBN (Algorithm 1) when two random BNs with the maximum indegree K are given. To this end, we extend the approximate method developed for analysis of the recursive algorithm for enumeration of singleton attractors of a single BN [18].

Assume that we have tested the first m out of n nodes, where $m \geq K$. The probability that $f_k^1(\mathbf{x}) \neq x_k$ or $f_k^2(\mathbf{x}) \neq x_k$ holds is given by

$$\begin{aligned} & \text{Prob}(f_k^1(\mathbf{x}) \neq x_k \text{ or } f_k^2(\mathbf{x}) \neq x_k) \\ &= \text{Prob}(f_k^1(\mathbf{x}) \neq x_k) + \text{Prob}(f_k^2(\mathbf{x}) \neq x_k) \\ &\quad - \text{Prob}(f_k^1(\mathbf{x}) \neq x_k \text{ and } f_k^2(\mathbf{x}) \neq x_k) \\ &= \frac{1}{2} \binom{m}{k_1} / \binom{n}{k_1} + \frac{1}{2} \binom{m}{k_2} / \binom{n}{k_2} \\ &\quad - \frac{1}{4} \binom{m}{k_2} / \binom{n}{k_2} \binom{m}{k_1} / \binom{n}{k_1} \\ &\approx \frac{1}{2} \left(\frac{m}{n}\right)^{k_1} + \frac{1}{2} \left(\frac{m}{n}\right)^{k_2} - \frac{1}{4} \left(\frac{m}{n}\right)^{k_1+k_2} \\ &\approx \frac{1}{2} \left(\frac{m}{n}\right)^K + \frac{1}{2} \left(\frac{m}{n}\right)^K - \frac{1}{4} \left(\frac{m}{n}\right)^{2K} \\ &= \left(\frac{m}{n}\right)^K - \frac{1}{4} \left(\frac{m}{n}\right)^{2K}. \end{aligned}$$

If $f_k^1(\mathbf{x}) \neq x_k$ or $f_k^2(\mathbf{x}) \neq x_k$ does not hold, the algorithm can proceed to the next recursion level. Therefore, the probability that the algorithm examines the $(m+1)$ -th node is no more than

$$\left[1 - \left(\left(\frac{m}{n}\right)^K - \frac{1}{4} \left(\frac{m}{n}\right)^{2K}\right)\right]^m = \left[1 - \left(\frac{m}{n}\right)^K + \frac{1}{4} \left(\frac{m}{n}\right)^{2K}\right]^m.$$

Hence the number of recursive calls executed for the first m nodes is at most

$$f(m) = 2^m \cdot \left[1 - \left(\frac{m}{n}\right)^K + \frac{1}{4} \left(\frac{m}{n}\right)^{2K}\right]^m,$$

Let $s = m/n$. Then, $f(m)$ is written as $f(m) = (g(s))^n$ using $g(s)$ defined by

$$g(s) = [(2 - 2s^K + \frac{1}{2}s^{2K})^s]^n.$$

Since the other factors are polynomially bounded, the order of the average case time complexity is estimated by computing $\max_{s \in [0,1]} g(s)$.

Table 1 compares the average case time complexity between the recursive algorithm for a single BN [10] and RecComAttBN. It is interesting that the complexity of RecComAttBN is lower than the algorithm in [10]. However,

it is reasonable because RecComAttBN puts a more strict condition to proceed to the next recursion level.

TABLE 1
Comparison of the order of the average case time complexity.

K	2	3	4	5	6
single BN [18]	1.35^n	1.43^n	1.49^n	1.54^n	1.57^n
two BNs	1.25^n	1.34^n	1.40^n	1.46^n	1.50^n

5.2 IlpComAttBN

IlpComAttBN (Algorithm 2) is based on integer linear programming (ILP) and is an extension of an ILP-based algorithm for detection of a singleton attractor for a single BN [27]. Although detection of an attractor is already a computational difficult problem (NP-hard) [26], ILP has been widely used to solve NP-hard problems and thus it is reasonable to apply ILP to detection of a common attractor. Different from RecComAttBN, IlpComAttBN does not enumerate all common attractors but outputs one common attractor (if any).

The basic idea of IlpComAttBN is to represent each Boolean function using a set of linear constraints on binary variables (i.e., 0-1 variables). Recall that the purpose of ILP is to maximize (or minimize) the objective linear function subject to linear constraints (linear equalities and inequalities) under the condition that each variable takes an integer value. For example, $z = x \wedge y$ can be represented by the following set of constraints:

$$\begin{aligned} z &\geq x + y - 1, \\ z &\leq \frac{1}{2}(x + y), \\ x &\in \{0, 1\}, \\ y &\in \{0, 1\}, \\ z &\in \{0, 1\}. \end{aligned}$$

The main difference between IlpComAttBN and the ILP-based algorithm in [27] is that variables $x_{i,b_{i_1} \dots b_{i_{k_i}}}$ in [27] are replaced with $x_{i,b_{i_1}^j \dots b_{i_{k_i}^j}^j}$ to reflect intermediate states of the j -th network.

Before presenting details of IlpComAttBN, we need some definitions. A *literal* is a variable (x) or its negation ($\bar{x}$), and a *term* is AND of literals (e.g., $x \wedge y$). It is well-known that every Boolean function can be represented in a *disjunctive normal form* (DNF), which is OR of terms. Here, we define $\sigma_b(x)$ by $\sigma_b(x) = \begin{cases} x, & \text{if } b = 1, \\ \bar{x}, & \text{otherwise.} \end{cases}$ Then, any Boolean function $f(x_1, \dots, x_k)$ with k inputs can be represented as a DNF formula given by

$$f(x_1, \dots, x_k) = \bigvee_{\{b_1, \dots, b_k\} \in \{0,1\}^k} \left(f(b_1, \dots, b_k) \wedge \sigma_{b_1}(x_1) \wedge \dots \wedge \sigma_{b_k}(x_k) \right).$$

Recall that an ILP problem is a mathematical optimization problem in which the objective function and constraints are linear and all the variables are restricted to be integers. In order to represent Boolean functions by linear constraints

(i.e., linear equalities and/or linear inequalities), we introduce $\tau_b(x)$ defined by $\tau_b(x) = \begin{cases} x, & \text{if } b = 1, \\ 1 - x, & \text{otherwise.} \end{cases}$

Here, we use v_i^j to denote the i -th node of G_j , k_i^j to denote the indegree of node v_i^j and $f_i^j(x_{i_1}^j, \dots, x_{i_{k_i^j}^j}^j)$ to denote the Boolean function assigned to v_i^j . For each $[b_{i_1}^j, \dots, b_{i_{k_i^j}^j}^j] \in \{0,1\}^{k_i^j}$, if $f_i^j(b_{i_1}^j, \dots, b_{i_{k_i^j}^j}^j) = 1$, we add constraints

$$x_{i,b_{i_1}^j \dots b_{i_{k_i^j}^j}^j}^j \geq \left(\sum_{l=1}^{k_i^j} \tau_{b_{i_l}^j}(x_{i_l}^j) \right) - (k_i^j - 1), \quad (1)$$

$$x_{i,b_{i_1}^j \dots b_{i_{k_i^j}^j}^j}^j \leq \frac{1}{k_i^j} \sum_{l=1}^{k_i^j} \tau_{b_{i_l}^j}(x_{i_l}^j)$$

to force $x_{i,b_{i_1}^j \dots b_{i_{k_i^j}^j}^j}^j$ to be 1 if and only if $\sigma_{b_{i_1}^j}(x_{i_1}^j) \wedge \dots \wedge \sigma_{b_{i_{k_i^j}^j}^j}(x_{i_{k_i^j}^j}^j)$ is satisfied, otherwise we simply add a constraint

$$x_{i,b_{i_1}^j \dots b_{i_{k_i^j}^j}^j}^j = 0. \quad (2)$$

Constraints (1) and (2) ensure that $x_{i,b_{i_1}^j \dots b_{i_{k_i^j}^j}^j}^j = 1$ if and only if $f_i^j(b_{i_1}^j, \dots, b_{i_{k_i^j}^j}^j) \wedge \sigma_{b_{i_1}^j}(x_{i_1}^j) \wedge \dots \wedge \sigma_{b_{i_{k_i^j}^j}^j}(x_{i_{k_i^j}^j}^j)$ is satisfied.

To determine the state x_i of a node v_i in a common attractor, we add the following constraints for each x_i .

$$\begin{aligned} x_i &\geq \frac{1}{2^{k_i^j}} \sum_{[b_{i_1}^j, \dots, b_{i_{k_i^j}^j}^j] \in \{0,1\}^{k_i^j}} x_{i,b_{i_1}^j \dots b_{i_{k_i^j}^j}^j}^j, \\ x_i &\leq \sum_{[b_{i_1}^j, \dots, b_{i_{k_i^j}^j}^j] \in \{0,1\}^{k_i^j}} x_{i,b_{i_1}^j \dots b_{i_{k_i^j}^j}^j}^j. \end{aligned} \quad (3)$$

Putting together constraints (1)-(3), we obtain IlpComAttBN as Algorithm 2 shown below. It is to be noted that ComAttBN does not need an objective function. However "Maximize x_1 " is given in IlpComAttBN since an objective is required to utilize any ILP solver.

As for the time complexity of IlpComAttBN, it is quite difficult to analyze because analysis of the detailed time complexity of ILP-based methods is almost impossible in general. Therefore, instead of the time complexity, we analyze the numbers of variables and constraints in the ILP instance generated by IlpComAttBN. Suppose that the maximum indegree of a given BN is K . Then, the number of variables (in particular, $x_{i,b_{i_1}^j \dots b_{i_{k_i^j}^j}^j}^j$) is clearly $O(2^K nN)$. Since

at most two constraints with length $O(K)$ are generated per $x_{i,b_{i_1}^j \dots b_{i_{k_i^j}^j}^j}^j$, the number of constraints with length $O(K)$ is $O(2^K nN)$. In addition, since $2N$ constraints with length $O(2^K)$ are generated for each x_i , the number of constraints with length $O(2^K)$ is $O(nN)$. Note that ILP solvers need exponential time of the size of the ILP instance in the worst case.

Algorithm 2 IlpComAttBN: ILP-based algorithm for common attractor detection

Maximize x_1 ,
Subject to

$$\begin{aligned}
x_{i,b_{i_1}^j \dots b_{i_{k_i^j}}^j} &\geq \left(\sum_{l=1}^{k_i^j} \tau_{b_{i_l}^j}(x_{i_l}^j) \right) - (k_i^j - 1), \\
x_{i,b_{i_1}^j \dots b_{i_{k_i^j}}^j} &\leq \frac{1}{k_i^j} \sum_{l=1}^{k_i^j} \tau_{b_{i_l}^j}(x_{i_l}^j), \\
&\text{for all } i \in \{1, \dots, n\}, j \in \{1, \dots, N\} \\
&\text{and } [b_{i_1}^j, \dots, b_{i_{k_i^j}}^j] \in \{0, 1\}^{k_i^j} \\
&\text{such that } f_i^j(b_{i_1}^j, \dots, b_{i_{k_i^j}}^j) = 1, \\
x_{i,b_{i_1}^j \dots b_{i_{k_i^j}}^j} &= 0, \\
&\text{for all } i \in \{1, \dots, n\}, j \in \{1, \dots, N\} \\
&\text{and } [b_{i_1}^j, \dots, b_{i_{k_i^j}}^j] \in \{0, 1\}^{k_i^j} \\
&\text{such that } f_i^j(b_{i_1}^j, \dots, b_{i_{k_i^j}}^j) = 0, \\
x_i &\geq \frac{1}{2^{k_i^j}} \sum_{[b_{i_1}^j, \dots, b_{i_{k_i^j}}^j] \in \{0, 1\}^{k_i^j}} x_{i,b_{i_1}^j \dots b_{i_{k_i^j}}^j}, \\
x_i &\leq \sum_{[b_{i_1}^j, \dots, b_{i_{k_i^j}}^j] \in \{0, 1\}^{k_i^j}} x_{i,b_{i_1}^j \dots b_{i_{k_i^j}}^j}, \\
&\text{for all } i \in \{1, \dots, n\} \text{ and } j \in \{1, \dots, N\}, \\
x_{i,b_{i_1}^j \dots b_{i_{k_i^j}}^j} &\in \{0, 1\}, \\
&\text{for all } i \in \{1, \dots, n\}, j \in \{1, \dots, N\} \\
&\text{and } [b_{i_1}^j, \dots, b_{i_{k_i^j}}^j] \in \{0, 1\}^{k_i^j}, \\
x_i &\in \{0, 1\}, \\
&\text{for all } i \in \{1, \dots, n\}.
\end{aligned}$$

5.3 RecSimAttBN

For both of problems SimAttBN and SimAttWBN, we modify the recursive algorithm RecComAttBN. The basic idea of RecSimAttBN is to examine system states for all networks in a node-to-node manner with checking whether the similarity among partial system states is not smaller than the given threshold. For SimAttBN, the pseudocode of the algorithm, RecSimAttBN, is given as Algorithm 3. For the simplicity, we present the code only for the case in which two BNs are given, where the extension to a general case is straightforward.

For SimAttWBN, it is enough to replace “if $s(\mathbf{x}^1, \mathbf{x}^2) \geq S$ ” with “if $s_w(\mathbf{x}^1, \mathbf{x}^2) \geq W$ ” and remove “ $s(\mathbf{x}^1, \mathbf{x}^2) < S$ or” of RecSimAttBN, where W is a specified lower bound of the weighted similarity. The resulting algorithm is referred to as RecSimAttWBN.

Algorithm 3 RecSimAttBN: Algorithm for SimAttBN problem

Input: Two BNs $G_1(V, F_1), G_2(V, F_2)$
similarity S

Output: all similar singleton attractor tuples

Initialize $i := 1$ and $\mathbf{x}^j := [\epsilon, \dots, \epsilon]$ for $j = 1, 2$

procedure RecSimAttBN($\mathbf{x}^1, \mathbf{x}^2, i$)

if $i = n + 1$ **then**

if $s(\mathbf{x}^1, \mathbf{x}^2) \geq S$ **then**

 output $(\mathbf{x}^1, \mathbf{x}^2)$

end if

return

end if

for $b_1 = 0$ **to** 1 **do**

$\mathbf{x}_i^1 \leftarrow b_1$

for $b_2 = 0$ **to** 1 **do**

$\mathbf{x}_i^2 \leftarrow b_2$

if it is found that $s(\mathbf{x}^1, \mathbf{x}^2) < S$ **or** $f_k^j(\mathbf{x}^j) \neq \mathbf{x}_k^j$ for some $k \leq i$ and $j \in \{1, 2\}$ **then**

continue

else

 RecSimAttBN($\mathbf{x}^1, \mathbf{x}^2, i + 1$)

end if

end for

end for

return

6 COMPUTATIONAL EXPERIMENTS AND DISCUSSIONS

In order to evaluate the efficiency of the algorithms proposed in Section 5 as well as to demonstrate the significance of common attractors and similar attractors in BNs, computational experiments were performed on both simulated BNs and a biological model. We employed random BNs for the former one and a Boolean model of TGF- β signaling pathway [28] for the latter one. Computational experiments on random BNs were done on a server with an AMD EPYC 7742 3.4GHz CPU and 512 GB RAM running under the Ubuntu 20.04.3 LTS operating system, where ILOG CPLEX (version 22.1) was utilized to solve integer linear programs with the single thread mode. Computational experiments on TGF- β pathway were done on a PC with an Intel Core i5 2.7GHz CPU and 8 GB RAM running under the macOS Mojave (version 10.14.5) operating system, where ILOG CPLEX (version 12.9) was utilized to solve integer linear programs.

6.1 Comparison of RecComAttBN and IlpComAttBN

For solving ComAttBN, we proposed a recursive algorithm, RecComAttBN, and an ILP-based algorithm, IlpComAttBN. We compared their performances to see which one is superior, using all combinations of $N = 2, 3, 4$, $K = 2, 3, 4$, and $n = 10, 15, 20, 25, 30, 35, 40, 45, 50, 55$. The results are shown in Fig. 2 (for the case of $(N = 3, K = 2)$) and Figs. S1-S8 in Supplemental Material (PDF). Furthermore, in order to examine the efficiency of IlpComAttBN for larger n , we examined cases up to $n = 100$ on IlpComAttBN. The results are shown in Figs. S9-S11 in Supplemental Material

(PDF). For each case, the average CPU time over 100 trails is shown.

We can see the following tendencies from the results.

- RecComAttBN performs better when n is small, while the advantage rapidly narrows down as n increases. This tendency is reasonable because RecComAttBN is based on a simple recursive procedure whereas IlpComAttBN utilizes a state-of-the-art ILP solver.
- CPU time of RecComAttBN increases exponentially with the size of BNs (n). Although CPU time increases as K increases, it decreases as N decreases. As discussed in Section 5.3, the latter tendency is reasonable because inconsistency would be detected at earlier steps as the number of BNs increases.
- The results suggest that IlpComAttBN works near linearly for the case of $K = 2$ with large n , but it needs exponential time for $K \geq 4$. For the case of $K = 3$, it seems that IlpComAttBN takes more than linear time although it is unclear whether IlpComAttBN needs exponential time. A similar tendency was also observed in a previous study [27] on an ILP-based method for detecting a singleton attractor for a single BN. It is a bit strange because the singleton attractor detection problem remains NP-hard even if $K = 2$ [13]. As discussed in [27], one of the reasons might be that 6 among 16 possible Boolean functions of $K = 2$ are constant or unary functions and thus the values of many nodes are determined quickly. Another possible reason is that randomly generated BNs for $K = 2$ may not have large connected components or may have small connectivities and thus detection of a singleton attractor may be easy.

The comparison suggests the potential of IlpComAttBN for being applicable to practical instances with large size. However, RecComAttBN has an advantage when n is small and/or enumeration of all common singleton attractors is required.

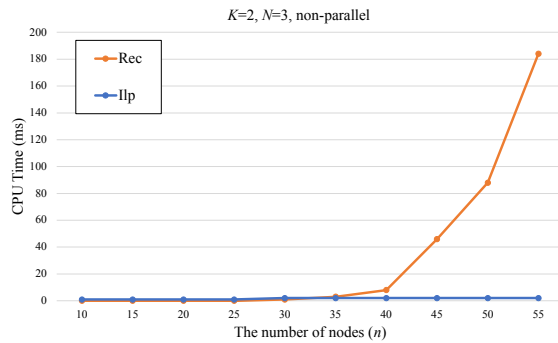


Fig. 2. CPU time (ms) of RecComAttBN (orange lines) and IlpComAttBN (blue lines) for ($N = 3, K = 2$).

6.2 Experiments on TGF- β Pathway

The transforming growth factor-beta (TGF- β) pathway is involved in the regulation of cell growth and differentiation [28], and plays an essential role in maintaining tissue homeostasis through its ability to induce cell cycle arrest,

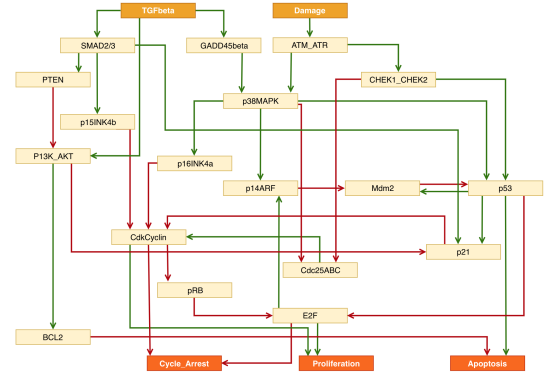


Fig. 3. BN model of the TGF- β pathway proposed in [28].

differentiation and apoptosis, and to preserve genomic stability [29]. The TGF- β signaling pathway is correlated with activation or inhibition processes of many proteins and thus of great significance in human diseases including cancers [30].

6.2.1 BN Model of TGF- β Pathway

A Boolean model of the crosstalk between the TGF- β , p38 MAPK and cell cycle checkpoint pathways was proposed in [28] with taking into account the effect of dysfunction (e.g. mutation and abnormal expression) that lead TGF- β to present its dual behavior and was validated by being consistent with several related experimental works. This model focuses on describing the dual role of TGF- β in cancer that while being able to block proliferation for the wild-type case, its deleterious dysfunction can impair cell cycle arrest promoting cellular proliferation.

The model is given in Fig. 3, where each node represents a molecular component such as a protein family comprising the pathways, except for the input and output nodes. Damage to the DNA (i.e. node “Damage”) and TGF- β signaling (i.e. node “TGFbeta”) are treated as input signals which are transmitted and processed in the network and eventually connected to the output proteins. This model has three output nodes corresponding to Apoptosis, Cycle_Arrest, and Proliferation. Besides, green lines represent activation interactions while red lines represent inhibitions in this directed interaction network.

This BN model has three singleton attractors as listed in Table 2. Each singleton attractor (i.e. a stable state of the system) indicates one possible cell fate under a different input combination of TGF- β and DNA damage.

Here we performed several experiments on the BN model of TGF- β pathway in order to demonstrate the potential of ComAttBN and SimAttWBN problems as well as our proposed algorithms in exploring tumor heterogeneity and homogeneity. Eight cancers with cancer subtypes were considered in our experiments, including Breast invasive carcinoma (BRCA), Colon adenocarcinoma (COAD), Rectum adenocarcinoma (READ), Kidney Chromophobe (KICH), Kidney renal clear cell carcinoma (KIRC), Kidney renal papillary cell carcinoma (KIRP), Lung adenocarcinoma (LUAD), and Lung squamous cell carcinoma (LUSC). The RNA-seq profiling data in these eight cancers containing 3,360 tumor and 371 normal samples were assembled by

TABLE 2
Singleton attractors in the BN model of TGF- β signaling pathway.

Input																		Output			stable states		
TGFbeta	Damage	SMAD2/3	p15INK4b	p16INK4a	p14ARF	PTEN	BCL2	Mdm2	p53	p21	pRB	p38MAPK	CdkCyclin	Cdc25ABC	E2F	CHEK1_CHEK2	ATM_ATR	P13K_AKT	GADD45beta	Apoptosis	Cycle_Arrest	Proliferation	
0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	1	Proliferative state
1	0	1	1	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0	1	0	1	0	Cycle Arrest state
1	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0	1	1	0	1	1	1	0	Apoptotic state

TABLE 3
Weights for the BN model of TGF- β signaling pathway.

node	TGFbeta	Damage	SMAD2/3	GADD45beta	ATM_ATR	PTEN	CHEK1_CHEK2	p38MAPK	p15INK4b	P13K_AKT	p16INK4a	p14ARF	Mdm2	p53	CdkCyclin	p21	pRB	Cdc25ABC	E2F	BCL2	Cycle_Arrest	Proliferation	Apoptosis
dist	4	5	4	4	4	4	3	3	3	3	3	4	3	2	2	3	3	3	2	2	1	1	1
w_i	0.4	0.2	0.4	0.4	0.4	0.4	0.6	0.6	0.6	0.6	0.6	0.4	0.6	0.8	0.8	0.6	0.6	0.6	0.8	0.8	1.0	1.0	1.0

the TCGA Data Coordinating Center [31]. The counts were log2-transformed before being used for further analysis.

To obtain the modified BN of a cancer, we firstly recovered the simplified model by identifying all involved molecular components in the network. For example, node “SMAD2/3” consists of two proteins, *Mothers against decapentaplegic homolog 2* and *Mothers against decapentaplegic homolog 3*.

Then, a modified t-statistic was utilized to measure up- and down-regulated genes between tumors and corresponding normal samples in each cancer type for TCGA RNA-seq sets. The adjusted P value was used for multiple hypothesis testing using Benjamini and Hochberg’s method [32], and the false discovery rate was controlled at 5%. Here, we used adjusted P values ≤ 0.05 and log2 (fold change) ≥ 1 or ≤ -1 to identify three groups, in which 1, 0, x denote that the gene with up- (≥ 1) or down- (≤ -1) regulation or no differential expression as described in [33], [34]. In other words, up- and down-regulated genes could be regarded as a gene with overexpression (i.e. status=1) and knockdown (i.e. status=0) in each cancer, respectively; whereas other ones belong to unchanged genes (i.e. status=x) or genes without expression data (i.e. status=-; Fig. 4). Recall that multiple genes may be associated with a single node. We further hypothesized that each gene associated with node v_i equally contributes to the Boolean function of v_i . Therefore, the Boolean function assigned to v_i is modified depending on the status of genes associated with v_i . Intuitively, if the majority of genes associated with v_i are overexpressed (resp., knocked down), the Boolean function f_i assigned to v_i is modified as $f_i(t) = 1$ (resp., $f_i(t) = 0$) for all t , otherwise f_i is not modified. The detailed rules are given below.

For node v_i having m ($m \geq m_o + m_k$) associated genes, we assume w.l.o.g. that v_i has m_o overexpressed genes, $g_1, g_2, \dots, g_{m_o}$, and m_k knockdown genes,

$g_{m_o+1}, g_{m_o+2}, \dots, g_{m_o+m_k}$, in cancer I based on differential expression between tumor and normal samples. Here, the differential expression for each gene is hypothesized to continuously occur during signaling transduction [i.e., $f_i^I(t) = f_i^I(t+1)$ for all t] since it has been known that the timescale of tumor progression (or gene expression) is longer than signaling transduction. Then, the modified Boolean function of v_i for cancer I (denoted by f_i^I) is defined as follows.

- the node v_i could be regarded as overexpression and knockdown when all its genes belong to overexpressed and knockdown ones, respectively; thus, $f_i^I(t) = 0$ if $m_o = 0$ and $m_k \neq 0$, while $f_i^I(t) = 1$ if $m_o \neq 0$ and $m_k = 0$;
- the genes of node v_i lack consistently differential expression; thus, when $m_o \neq 0$ and $m_k \neq 0$, $f_i^I(t) = 1$ if $m_o > m_k$, while $f_i^I(t) = 0$ if $m_o < m_k$;
- the genes of node v_i lack consistently differential expression and the numbers of overexpressed and knockdown genes are the same; thus, when $m_o = m_k \neq 0$, f_i^I has two modification forms, $f_i^I(t) = 1$ and $f_i^I(t) = 0$, and would be respectively used in the experiment;
- otherwise $f_i^I = f_i$ (i.e., the Boolean function is not modified).

Based on the above rules, modifications on the BN for each cancer are shown in Fig. 4. It can be seen that many nodes have multiple genes. For each gene and each cancer type, the differential expression result (0,1,x,-) obtained from gene expression data is shown in the ‘Status’ column. For the column f_i , 0 and 1 (in red color) mean that the Boolean function assigned to a node v_i should be modified to $f_i(t) = 0$ and $f_i(t) = 1$ (for all t), respectively. Note that the third rule is applied only to the case of $I = \text{READ}$, for which there exist two f_i columns.

Node, v_i	i	Gene (UniProtKB)	BRCA	COAD	READ	KICH	KIRC	KIRP	LUAD	LUSC
			Status f_i	Status f_i	Status f_i	Status f_i	Status f_i	Status f_i	Status f_i	Status f_i
SMAD2/3	2	Q15796	X	X	X	X	X	X	X	X
	2	P84022	X	X	X	X	X	X	X	X
GADD45beta	3	O75293	X	X	X	X	X	0	0	0
ATM_ATR	4	Q13315	X	X	X	X	X	X	X	X
	4	Q13535	X	X	X	X	X	X	X	X
PTEN	5	P60484	X	X	X	X	X	X	X	X
CHEK1_CHEK2	6	O14757	1	1	X	X	X	1	1	1
	6	O96017	X	1	X	X	1	X	1	1
p38MAPK	7	Q16539	X	X	X	X	X	X	X	X
p15INK4b	8	P42772	X	0	0	0	0	X	1	1
	9	B0LPE5	X	X	X	X	X	X	X	X
	9	P42336	X	X	X	X	X	X	X	X
	9	Q4LE51	-	-	-	-	-	-	-	-
	9	P42338	X	X	X	X	X	X	X	X
	9	O00329	X	0	0	X	X	X	X	X
	9	A0A2K8FKV1	-	-	-	-	-	-	-	-
	9	P27986	0	X	X	0	X	X	0	X
	9	A0A2X0SFG1	-	-	-	-	-	-	-	-
	9	O00459	1	X	X	X	X	X	X	1
P13K_AKT	9	Q92569	X	X	X	X	X	X	X	X
	9	Q8N381	X	0	X	0	0	X	1	X
	9	Q5UE93	X	0	X	1	1	1	X	0
	9	B3KRK9	X	0	X	1	1	1	X	0
	9	Q8WYR1	X	0	0	X	1	1	0	0
	9	L7RT34	X	0	0	X	1	1	0	0
	9	P48736	X	0	0	X	1	X	0	0
	9	A0A024R720	X	0	0	X	1	X	0	0
	9	A8K9G9	X	0	0	X	1	X	0	0
	9	P31749	X	X	X	X	X	X	X	X
	9	P31751	X	X	X	X	X	X	X	X
	9	Q9Y243	0	0	0	X	X	X	0	0
p16INK4a	10	P42771	1	1	1	1	1	1	1	1
	10	Q8N726	1	1	1	1	1	1	1	1
p14ARF	11	P42771	1	1	1	1	1	1	1	1
Mdm2	12	Q00987	X	X	X	X	X	X	X	X
p53	13	P04637	X	X	X	X	X	X	X	X
	14	P06493	1	1	1	X	X	X	1	1
	14	P78396	X	0	0	X	X	X	0	X
CdkCyclin	14	P20248	1	1	1	X	0	1	1	1
	14	Q8WWL7	X	X	X	X	X	X	X	0
	14	P14635	1	1	X	X	X	X	1	1
	14	O95067	1	X	X	1	1	1	1	1
p21	15	P38936	X	0	0	0	0	X	1	X
pRB	16	P06400	X	X	X	X	X	X	X	X
	17	P30304	1	1	1	X	X	X	1	1
Cdc25ABC	17	P30305	X	1	1	1	1	X	1	X
	17	P30307	1	1	1	1	1	1	1	1
	18	Q01094	1	1	1	1	1	1	1	1
E2F	18	Q14209	1	X	1	X	1	1	1	1
	18	O00716	X	1	1	1	X	X	1	1
BCL2	19	P10415	X	0	0	0	0	X	X	X

Fig. 4. Modification on BN for eight cancers (BRCA, COAD, ..., LUSC). For each cancer type, 'Status' means whether the corresponding gene is regarded as overexpressed (i.e., status=1), knockdown (i.e., status=0), unchanged (i.e., status=x), or unclear (status=-). For the column f_i , 0 and 1 (red color) mean that the Boolean function assigned to a node v_i is modified to $f_i(t) = 0$ and $f_i(t) = 1$ (for all t), respectively, whereas the blank item means that the Boolean function is not modified.

For the SimAttWBN problem, a weight vector $w = [w_1, \dots, w_n]$ is required. We define w_i as the length of the shortest path from node v_i to three output nodes. The weight of an output node is set to be 1. Weight of each node is shown in Table 3.

6.2.2 Common attractors in cancer subtypes

We first applied IlpComAttBN. Note that since there were two modifications of v_{14} in $I = \text{READ}$, we respectively performed the experiment twice for $f_{14}^{\text{READ}}(t) = 0$ and $f_{14}^{\text{READ}}(t) = 1$ (red bold letter, Fig. 4). The results are shown in Fig. 5, where only half of the results are shown for each case since the matrix is symmetric.

Each light blue box denotes that a common attractor was detected between a cancer cell and itself, which is obvious as long as the BN has singleton attractors. Each navy blue box denotes that a common attractor was detected between two different cancers including (BRCA, LUAD), (BRCA, LUSC), (KICH, KIRC) and (LUAD, LUSC) when v_{14} was knocked down in READ, and (BRCA, LUAD), (BRCA, LUSC), (COAD, READ), (KICH, KIRC) and (LUAD, LUSC) when v_{14} was overexpressed in READ. The result (shown by thick red boxes) that common attractors were found in (COAD, READ), (KICH, KIRC), and (LUAD, LUSC), indicates that high similarities are shared between cancer

subtypes. Another observation from Fig. 5 is that it shows high similarity between breast cancer and two lung cancer subtypes, as common attractors were found in (BRCA, LUAD) and (BRCA, LUSC). This result is reasonable because breast cancer is known as second cancer that develops in lung carcinoid cancer survivors [35]. A second cancer is a new cancer that is irrelevant to any previous cancer diagnosis and also not the same as a cancer recurrence.

knockdown of node 14										
	BRCA	COAD	READ	KICH	KIRC	KIRP	LUAD	LUSC		
BRCA	1							1	1	
COAD		1								
READ			1							
KICH				1	1					
KIRC					1					
KIRP						1				
LUAD							1	1		
LUSC								1		

overexpression of node 14										
	BRCA	COAD	READ	KICH	KIRC	KIRP	LUAD	LUSC		
BRCA	1							1	1	
COAD		1	1							
READ			1							
KICH				1	1					
KIRC					1					
KIRP						1				
LUAD							1	1		
LUSC								1		

Fig. 5. Result of common attractor detection for eight cancers, where blue boxes mean that common attractors were detected.

	BRCA	COAD	READ	KICH	KIRC	KIRP	LUAD	LUSC
BRCA		0.96	0.84	0.79	0.79	0.83	1	1
COAD			0.88	0.83	0.79	0.79	0.96	0.96
READ				0.78	0.67	0.67	0.84	0.84
KICH					1	0.96	0.79	0.79
KIRC						0.93	0.75	0.75
KIRP							0.83	0.83
LUAD								1
LUSC								

Fig. 6. Result of similar attractor detection for eight cancers.

6.2.3 Similar attractors in cancer subtypes

From Fig. 5, the result that no common attractors were found in (KICH, KIRP) or (KIRC, KIRP) is a bit surprising because these two pairs are subtypes of kidney cancer (highlighted by thick black boxes). One possible reason is

that the expected number of common singleton attractors is quite few as we discussed in Section 4. Hence we applied RecSimAttWBN to pairs of eight cancers, utilizing the weights given in Table 3. Here, the result is represented by a score defined as $\frac{s_w}{\sum_i w_i}$ to intuitively describe how similar two attractors from different BNs are. Results are shown in Fig. 6 where $\frac{s_w}{\sum_i w_i} = 1$ if a common attractor exists between two BNs. Note again that the result is the average of two trials where the only difference is on f_{14}^{READ} . For example, $s_w(G_{COAD}, G_{READ}) = 1$ when v_{14} is overexpressed and $s_w(G_{COAD}, G_{READ}) = 0.76$ when v_{14} is knocked down, thus we get $s_w(G_{COAD}, G_{READ}) = \frac{1+0.76}{2} = 0.88$.

It is seen from Fig. 6 that all pairs of two cancer subtypes including (COAD, READ), (KICH, KIRC), (KICH, KIRP), (KIRC, KIRP), and (LUAD, LUSC) got higher scores (≥ 0.88) than other pairs except the five pairs surrounded by two boxes. This result indicates high similarity in (KICH, KIRP) and (KIRC, KIRP) even though no common attractors were found. Combined with the result in Section 6.2.2, we can observe that BNs of two cancer subtypes usually share common attractors or at least have higher similarity compared to pairs not belonging to the same cancer type.

Interestingly, we further found that five cancer type pairs showed high similarities (highlighted by thick black boxes in Fig. 6). As a possible reason for common attractors found in (BRCA, LUAD) and (BRCA, LUSC) that we discussed in Section 6.2.2, the high similarity in (BRCA, COAD), (COAD, LUAD) and (COAD, LUSC) is also understandable because colon cancer is known as the second cancer after both breast cancer [36] and lung cancer [37].

7 CONCLUSION

In this paper, we studied the problems of finding common attractors and similar attractors in multiple Boolean networks (BNs), and proposed four methods, RecComAttBN, IlpComAttBN, RecSimAttBN, and RecSimAttWBN. Theoretical results on random BNs show that the number of common attractors and similar attractors decreases as the number of BNs grows. This result is reasonable because different BNs tend to have different attractors. Interestingly, the theoretical result on the average case analysis of RecComAttBN suggests that it may be easier to find a common attractor for multiple BNs than for a single BN. This observation gives a justification of the proposed approach, compared to the trivial approach that computes attractors separately for respective BNs and then finds common or similar attractors. The results on computational experiments on random BNs show that the integer linear programming (ILP)-based algorithm (IlpComAttBN) is much faster than the recursive algorithm (RecComAttBN) for large size BNs, although RecComAttBN has an advantage that it can enumerate all common singleton attractors. Therefore, development of ILP-based methods for finding similar attractors for multiple BNs is important future work.

We also demonstrated the potential usefulness of the proposed approach by applications to the analysis of a BN model of the TGF- β pathway. The results suggest that networks corresponding to subtypes of same cancer or closely related cancers tend to have common or similar attractors,

implying our models could further provide clues to explore tumor homogeneity and heterogeneity.

Tumor heterogeneity is often observed at the phenotypic level. That means that a tumor appears to be different across types and patients, suggesting heterogeneity also at the genetic level, and *vice versa* for tumor homogeneity. In our case, an attractor refers to a computational phenotype which directly translates to cellular behavior: Proliferation, apoptosis, and cell cycle arrest. We can detect similar or homogeneous attractors (phenotypes), using the approach presented in this paper. Although these phenotypes appear to be the same or similar, our method offers the possibility to study similarities and differences at the level of the underlying network, corresponding to the genotype. For example, the four, respectively five pairs of BNs with a common attractor (Fig. 5) suggest homogeneity on the phenotypic level. The underlying structure of the BNs however is not always the same: The BRCA BN is different from the LUAD and LUSC BNs. Hence, our method can assist in detecting genotypic heterogeneity based on phenotypic homogeneity. These insights can furthermore assist in hypothesis-generation for the selection of drug targets.

In this paper, we have focused on singleton attractors. However, periodic attractors also play important roles. Unfortunately, no scalable method is known for detection/enumeration of periodic attractors even for a single BN [13]. Therefore, development of efficient methods for detection and/or enumeration of common and/or similar periodic attractors in multiple BNs is a challenging future task.

Another important issue is to consider the effects of noise and uncertainty because biological systems often contain noise and show probabilistic behaviors. In order to cope with such effects, probabilistic Boolean networks (PBNs) have been extensively studied [8], [38], [39]. In a PBN, multiple Boolean functions can be assigned to each node and one function is selected per node at each time step according to a specified probability distribution. Therefore, a PBN can be considered as a set of constituent BNs. Although there is no strong consensus on the definition of attractors of a PBN, the most common one would be that the set of attractors of a PBN is the union of the sets of attractors of the constituent BNs [40], [39]. However, it is difficult to extend the theoretical results presented in Section 4 to PBNs by the following reasons. The proofs of the results heavily depend on the random BN model. However, no consensus model is known for random PBNs. The analyses also heavily depend on the fact that the next system state will be uniquely determined from 2^n possible states with probability $\frac{1}{2^n}$ for each BN. However, we cannot utilize such property for PBNs. Of course, we can consider all possible combinations of the constituent BNs. Then, the upper bounds of the expected numbers of common and similar singleton attractors might be given by multiplying the number of combinations to the numbers given in Proposition 1 and Theorem 1, respectively. However, the number of combinations might be too large. For example, suppose that two Boolean functions are assigned to each node. Then, there would be 2^n constituent BNs and the number of combination would be 2^{nN} . To multiply 2^{nN} might give too loose estimates and thus would be meaningless. Therefore,

theoretical analysis of the expected number of common and similar attractors for PBNs is left as future work.

In dynamical evolution of genes, it may encounter perturbations like knock-out or knock-in perturbations. Therefore, it is important to extend the results of this paper to include the effect of such events. However, it is difficult to extend the theoretical results for that purpose because we need to consider probabilistic models of knock-out and knock-in perturbation, for which difficulties similar to those for PBNs would arise. On the other hand, if all occurred knock-out events are known, it might be possible to modify the algorithms to take the effect of such events into account. For example, if the i th gene of BN G_j is knocked out, the resulting common attractors may be obtained by adding a new BN obtained from G_j by letting $v_i(t) = 0$ for all t to the set of input BNs. For the case of knocked-in events, the situation would be different and more difficult because we need to know the effect of knocked-in events to other genes (e.g., how Boolean functions are affected by introduction of knocked-in genes). Therefore, in order to study the effect of perturbations, we need to develop appropriate models as well as novel analysis and computational methods.

ACKNOWLEDGMENT

Tatsuya Akutsu was partially supported by Grants-in-Aid #22H00532 and #22K19830 from JSPS, Japan.

REFERENCES

- [1] S. A. Kauffman, "Homeostasis and differentiation in random genetic control networks," *Nature*, vol. 224, no. 5215, pp. 177–178, 1969.
- [2] —, *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, 1993.
- [3] M. Anthony, *Discrete Mathematics of Neural Networks, Selected Topics*. SIAM, 2001.
- [4] S. S. Aghamiri, V. Singh, A. Naldi, T. Helikar, S. Soliman, and A. Niarakis, "Automated inference of boolean models from molecular interaction maps using casq," *Bioinform.*, vol. 36, no. 16, pp. 4473–4482, 2020.
- [5] S. Watterson, S. Marshall, and P. Ghazal, "Logic models of pathway biology," *Drug Discov. Today*, vol. 13, no. 9–10, pp. 447–456, 2008.
- [6] J. Béal, L. Pantolini, V. Noël, E. Barillot, and L. Calzone, "Personalized logical models to investigate cancer response to braf treatments in melanomas and colorectal cancers," *PLOS Comput. Biol.*, vol. 17, no. 1, pp. 1–23, 2021.
- [7] G. Selvaggio, S. Canato, A. Pawar, P. T. Monteiro, P. S. Guerreiro, M. M. Brás, F. Janody, and C. Chaouiya, "Hybrid epithelial-mesenchymal phenotypes are controlled by microenvironmental factors," *Cancer Res.*, vol. 80, no. 11, pp. 2407–2420, 2020.
- [8] K. Kobayashi and K. Hiraishi, "Design of probabilistic boolean networks based on network structure and steady-state probabilities," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 8, pp. 1966–1971, 2017.
- [9] Y. We and T. Shen, "Policy iteration algorithm for optimal control of stochastic logical dynamical systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 2031–2036, 2018.
- [10] X. Zhang, H. Han, and W. Zhang, "Identification of boolean networks using premined network topology," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 2, pp. 464–469, 2017.
- [11] D. Cheng and H. Qi, "Model construction of boolean network via observed data," *IEEE Trans. Neural Netw.*, vol. 21, no. 4, pp. 584–594, 2010.
- [12] D. Cheng, H. Qi, and Z. Li, *Analysis and Control of Boolean Networks: A Semi-tensor Product Approach*. Springer-Verlag, 2011.
- [13] T. Akutsu, *Algorithms for Analysis, Inference, and Control of Boolean Networks*. World Scientific Publishing Co Pte Ltd, 2018.
- [14] J. Aracena, J. Demongeot, and E. Goles, "Positive and negative circuits in discrete neural networks," *IEEE Trans. Neural Netw.*, vol. 15, no. 1, pp. 77–83, 2004.
- [15] E. Dubrova and M. Tesienko, "A sat-based algorithm for finding attractors in synchronous boolean networks," *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 8, no. 5, pp. 1393–1399, 2011.
- [16] A. Garg, A. Di Cara, I. Xenarios, L. Mendoza, and G. De Micheli, "Synchronous versus asynchronous modeling of gene regulatory networks," *Bioinform.*, vol. 24, no. 17, pp. 1917–1925, 2008.
- [17] Q. Yuan, H. Qu, J. Pang, and A. Mizera, "Improving bdd-based attractor detection for synchronous boolean networks," *Science China Information Sciences*, vol. 59, no. 8, p. Art no. 080101, 2016.
- [18] S.-Q. Zhang, M. Hayashida, T. Akutsu, W.-K. Ching, and M. K. Ng, "Algorithms for finding small attractors in boolean networks," *EURASIP J. Bioinform. Syst. Biol.*, vol. 2007, p. 20180, 2007.
- [19] J. Zhong, Y. Liu, J. Lu, and W. Gui, "Pinning control for stabilization of boolean networks under knock-out perturbation," *IEEE Trans. Automat. Contr.*, vol. 67, no. 3, pp. 1550–1557, 2021.
- [20] Y. Qiu, T. Tamura, W.-K. Ching, and T. Akutsu, "On control of singleton attractors in multiple boolean networks: integer programming-based method," *BMC Syst. Biol.*, vol. 8, no. Suppl-1, p. S7, 2014.
- [21] H. Dai, L. Li, and C. Launan, "Cell-specific network constructed by single-cell rna sequencing data," *Nucleic Acids Res.*, vol. 47, no. 11, p. e62, 2019.
- [22] A. Roli, S. Benedettini, R. Serra, and M. Villani, "Analysis of attractor distances in random boolean networks," in *Neural Nets WIRN10: Proceedings of the 20th Italian Workshop on Neural Nets*, vol. 226, 2011, p. 201.
- [23] C. Müssel, M. Hopfensitz, and H. A. Kestler, "Boolnet package vignette," 2019. [Online]. Available: https://cran.r-project.org/web/packages/BoolNet/vignettes/BoolNet_package_vignette.Snw.pdf
- [24] I. Harvey and T. Bossomaier, "Time out of joint: attractors in asynchronous random boolean networks," in *Proceedings of the 4th European Conference on Artificial Life (ECAL '97)*, 1997, pp. 67–75.
- [25] A. Mochizuki, "An analytical study of the number of steady states in gene regulatory networks," *J. Theoret. Biol.*, vol. 236, no. 3, pp. 291–310, 2005.
- [26] T. Akutsu, S. Kuhara, O. Maruyama, and S. Miyano, "A system for identifying genetic networks from gene expression patterns produced by gene disruptions and overexpressions," *Genome Inform.*, vol. 9, pp. 151–160, 1998.
- [27] T. Akutsu, Y. Zhao, M. Hayashida, and T. Tamura, "Integer programming-based approach to attractor detection and control of boolean networks," *IEICE Trans. Inform. Syst.*, vol. 95, no. 12, pp. 2960–2970, 2012.
- [28] V. V. Rossato, D. A. Silveira, S. Gupta, and J. C. M. Mombach, "Towards the contribution of the p38mapk pathway to the dual role of $\text{tgf}\beta$ in cancer: A boolean model approach," *Comput. Biol. Med.*, vol. 104, pp. 235–240, 2019.
- [29] M. Tian and W. P. Schiemann, "The $\text{tgf}\beta$ paradox in human cancer: an update," *Future Oncol.*, vol. 5, no. 2, pp. 259–271, 2009.
- [30] J. Massagué, S. W. Blain, and R. S. Lo, "Tgf β signaling in growth control, cancer, and heritable disorders," *Cell*, vol. 103, no. 2, pp. 295–309, 2000.
- [31] The Cancer Genome Atlas Research Network, J. N. Weinstein, E. A. Collisson, G. B. Mills, K. R. M. Shaw, B. A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, and J. M. Stuart, "The cancer genome atlas pan-cancer analysis project," *Nat. Genet.*, vol. 45, no. 10, pp. 1113–1120, 2013.
- [32] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate - a practical and powerful approach to multiple testing," *J. Royal Stat. Soc. B*, vol. 57, no. 1, pp. 289–300, 1995.
- [33] C.-Y. Lin, C.-H. Lee, Y.-H. Chuang, J.-Y. Lee, Y.-Y. Chiu, Y.-H. W. Lee, Y.-J. Jong, J.-K. Hwang, S.-H. Huang, L.-C. Chen, C.-H. Wu, S.-H. Tu, Y.-S. Ho, and J.-M. Yang, "Membrane protein-regulated networks across human cancers," *Nat. Commun.*, vol. 10, p. 3131, 2019.
- [34] C.-Y. Lin, P. Ruan, R. Li, J.-M. Yang, S. See, J. Song, and T. Akutsu, "Deep learning with evolutionary and genomic profiles for identifying cancer subtypes," *J. Bioinform. Comput. Biol.*, vol. 17, no. 3, p. 1940005, 2019.
- [35] American Cancer Society, "Second Cancers After Lung Carcinoid Tumor," <https://www.cancer.org/cancer/lung-carcinoid-tumor/after-treatment/no-longer-working.html>, 2019.

- [36] —, “Second Cancers After Breast Cancer,” <https://www.cancer.org/cancer/breast-cancer/living-as-a-breast-cancer-survivor/second-cancers-after-breast-cancer.html>, 2019.
- [37] —, “Second Cancers After Non-Small Cell Lung Cancer,” <https://www.cancer.org/cancer/non-small-cell-lung-cancer/after-treatment/second-cancers.html>, 2019.
- [38] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, “Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks,” *Bioinform.*, vol. 18, no. 2, pp. 261–274, 2002.
- [39] P. Trairatphisan, A. Mizera, J. Pang, A. A. Tantar, J. Schneider, and T. Sauter, “Recent development and biomedical applications of probabilistic boolean networks,” *Cell Commun. Signal.*, vol. 11, no. 1, p. 46, 2013.
- [40] M. Brun, E. R. Dougherty, and I. Shmulevich, “Steady-state probabilities for attractors in probabilistic boolean networks,” *Signal Process.*, vol. 85, no. 10, pp. 1993–2013, 2005.



Masahiro Ohtomo received the B.E. and M.E. degrees in computer science and engineering from the Toyohashi University of Technology in 2016 and 2018, respectively. He is currently enrolled in the doctoral program of informatics at Kyoto University. His research interests include motif extraction and analysis of proteins and genes.



Cao Yu received the B.E. degree in software engineering from Dalian University of Technology and the M.E. degree in informatics from Kyoto University, 2016 and 2019, respectively. She has been a systems engineer in Astellas Pharma Inc. Her research interests focused on Boolean networks during the master course.



Pi Wenya received B.E. degree in software engineering from the University of Central South, China, in 2016, and the degree of master of informatics from the University of Kyoto, Japan, in 2019. Her research interest include computational analysis of biological networks and she is currently a product manager of an internet company in China.



Tatsuya Akutsu received the B.E. and M.E. degrees in aeronautics and the D.E. degree in information engineering from the University of Tokyo, 1984, 1986, and 1989, respectively. He has been a professor in the Bioinformatics Center, Institute for Chemical Research, Kyoto University since 2001. discrete algorithms. His research interests include bioinformatics and discrete algorithms. He is a senior member of the IEEE.



Chun-Yu Lin received his M.Sc. and Ph.D. degrees both in Bioinformatics and Systems Biology from National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 2010 and 2014, respectively. He has been an assistant professor in Bioinformatics and Systems Biology, National Yang Ming Chiao Tung University, since 2019. His primary research interests are integrating multi-omics data with the biological networks for establishing disease-focused regulatory network models.



Münzner Ulrike received a a doctorate in Theoretical Biophysics in 2017 from Humboldt-Universität zu Berlin, Germany. She has been a postdoctoral researcher at Kyoto University from 2018 to 2020, and a project assistant professor at Osaka University from 2020 to 2022. Her research interests include signaling pathway reconstruction and mathematical modeling of biological processes, especially using Boolean networks. She is currently a bioinformatician at a biopharmaceutical company in Norway.