Received: 31 December 2020 | Revised: 28 April 2021 | Accepted: 26 May 2021

DOI: 10.1002/cso2.1027

ORIGINAL ARTICLE

COMPUTATIONAL AND SYSTEMS ONCOLOGY Open Access

WILEY

Oncogenetic network estimation with disjunctive Bayesian networks

Phillip B. Nicol[1] | **Kevin R. Coombes**[2] | **Courtney Deaver**[3] | **Oksana Chkrebtii**[4] | **Subhadeep Paul**[4] | **Amanda E. Toland**[5] | **Amir Asiaee**[6]

[1] Harvard College, Cambridge, Massachusetts

[2] Department of Biomedical Informatics, Ohio State University, Columbus, Ohio

[3] Natural Sciences Division, Pepperdine University, Malibu, California

[4] Department of Statistics, Ohio State University, Columbus, Ohio

[5] Department of Cancer Biology and Genetics and Department of Internal Medicine, Division of Human Genetics, Comprehensive Cancer Center, Ohio State University, Columbus, Ohio

[6] Mathematical Biosciences Institute, Ohio State University, Columbus, Ohio

Correspondence
Amir Asiaee, Mathematical Biosciences Institute, Ohio State University, Columbus, OH 43210.
Email: asiaeetaheri.1@osu.edu

Funding information
National Science Foundation, Grant/Award Numbers: DMS 1440386, DMS 1757423; National Human Genome Research Institute, Grant/Award Number: K99HG011367

Abstract

Motivation: Cancer is the process of accumulating genetic alterations that confer selective advantages to tumor cells. The order in which aberrations occur is not arbitrary, and inferring the order of events is challenging due to the lack of longitudinal samples from tumors. Moreover, a network model of oncogenesis should capture biological facts such as distinct progression trajectories of cancer subtypes and patterns of mutual exclusivity of alterations in the same pathways. In this paper, we present the disjunctive Bayesian network (DBN), a novel oncogenetic model with a phylogenetic interpretation. DBN is expressive enough to capture cancer subtypes' trajectories and mutually exclusive relations between alterations from unstratified data.

Results: In cases where the number of studied alterations is small (< 30), we provide an efficient dynamic programming implementation of an exact structure learning method that finds a best DBN in the superexponential search space of networks. In rare cases that the number of alterations is large, we provided an efficient genetic algorithm in our software package, OncoBN. Through numerous synthetic and real data experiments, we show OncoBN's ability in inferring ground truth networks and recovering biologically meaningful progression networks.

Availability: OncoBN is implemented in R and is available at https://github.com/phillipnicol/OncoBN.

KEYWORDS

Bayesian network, cancer progression, oncogenetic model, tumor phylogenetic

1 | INTRODUCTION

Cancer is the process of accumulating molecular alterations that over time lead to cancer hallmarks [1]. A natural question to ask is whether the order of alterations follows a particular pattern. Phylogenetic tree reconstruction methods answer this problem for individual tumors [2]. However, historically, due to the lack of high-resolution multiregion data of individual tumors, oncogenetic models were considered first. Oncogenetic models of tumorigenesis utilize many samples from the population of patients to estimate the order of alterations occur at the *disease* level, but are silent about the order of events at the *individual tumor* and *cell* levels. Recent technologies have enabled researchers to delineate various modes of evolution [3] and depict tumors' evolutionary history in an unprecedented resolution [4]. Although high-resolution data from individual tumors help infer the tumor's history, they do not provide the big picture of how a specific cancer type evolves. *In this work, we are taking first steps to reconciling these two levels of cancer progression modeling*.

The first oncogenetic model of tumorigenesis by Fearon and Vogelstein [5] was developed for colon cancer and suggested that a *chain* of aberrations is required to transform normal cells into carcinoma. Desper's *Oncogenetic trees* [6] modeled progression as a rooted directed tree. Mixtures of oncogenetic trees [7, 8] were proposed to capture the presence of an aberration in multiple progression paths. *Directed acyclic graphs* (DAGs) are the next straightforward generalization of tree-based models, as they allow multiple alterations (parents) to set up the clonal stage for the appearance of a new aberration (the child). *Bayesian networks* (BNs), which are DAGs equipped with a joint probability distribution [9], lend themselves naturally to representing such models. Perhaps, the most famous BN model of cancer progression is the *conjunctive Bayesian network* (CBN) [10, 11] that assumes all parent aberrations must be present in order for a child to occur.

The evolutionary interpretation of oncogenetic graphs is challenging. The most concrete biological way of thinking about an edge $e = (v, u)$ in such DAGs is to assume mutation v fixates in the cell population and prepares the tumor for the next selective sweep by u [11]. In other words, all mutations are assumed to be clonal, which is not accurate because of the observed intratumor heterogeneity in many cancer types [12]. *Our proposed tumorigenesis model has a phylogenetic interpretation and accommodates the presence of subclonal alterations*.

At its core, inferring cancer progression networks is the BN structure learning problem, which is NP-hard [13]. Various approximation and search algorithms have been proposed for cancer progression inference [11, 14, 15]. These algorithms' objective is to find a network structure that maximizes a (regularized) likelihood. The optimal network learned by any approximation method may be far from the ground truth and iterative search methods can get trapped in local maximums. *Here, we show that for the number of driver alterations that we often encounter in tumors* (< 30), *one can use an efficient dynamic programming (DP) implementation of an exact structure learning algorithm* [16].

1.1 | Related work

Mutual exclusivity of alterations is another phenomenon that was considered in learning cancer progression networks. Two sets of alterations are mutually exclusive if they (almost) never co-occur in a tumor [17]. Two potential explanations for this observation are functional redundancy and synthetic lethality [18]. Existing approaches considering pathways and their effects on cancer progression either assume that the pathways are inputs of the progression inference algorithm [19, 20] or learn them along with the progression network [21, 22].

The CBN progression rule dictates that all parent alterations need to be present in the tumor for the child to occur, under which mutually exclusive genes cannot share any descendant alterations. CBN's inability to capture mutual exclusivity of alterations has motivated a line of work in which the mutual exclusivity restriction and pathway information are introduced artificially to the CBN [19, 20]. Moreover, as each cancer subtype has distinct molecular characteristics and progression paths, one must first stratify samples to disjoint subtypes and then learn each subtype's progression network separately. This extra step is required for all of the above models mainly because they cannot naturally capture subtypes' mutual exclusivity. PICNIC [23] is the state-of-the-art pipeline that clusters samples to subtypes, detects driver events, checks for statistically significant mutual exclusivity hypotheses, or takes pathway information as an input, and infers the progression network.

Several recent works attempt to model the accumulation of alterations by Suppes' probability raising causal framework [23–27]. Farahani and Lagergren [14] proposed (semi-)monotone progression networks without any biological interpretation. The class of monotone BNs is a superset of our proposed model that makes it more flexible but prone to overfitting due to lack of enough samples in many real-world scenarios.

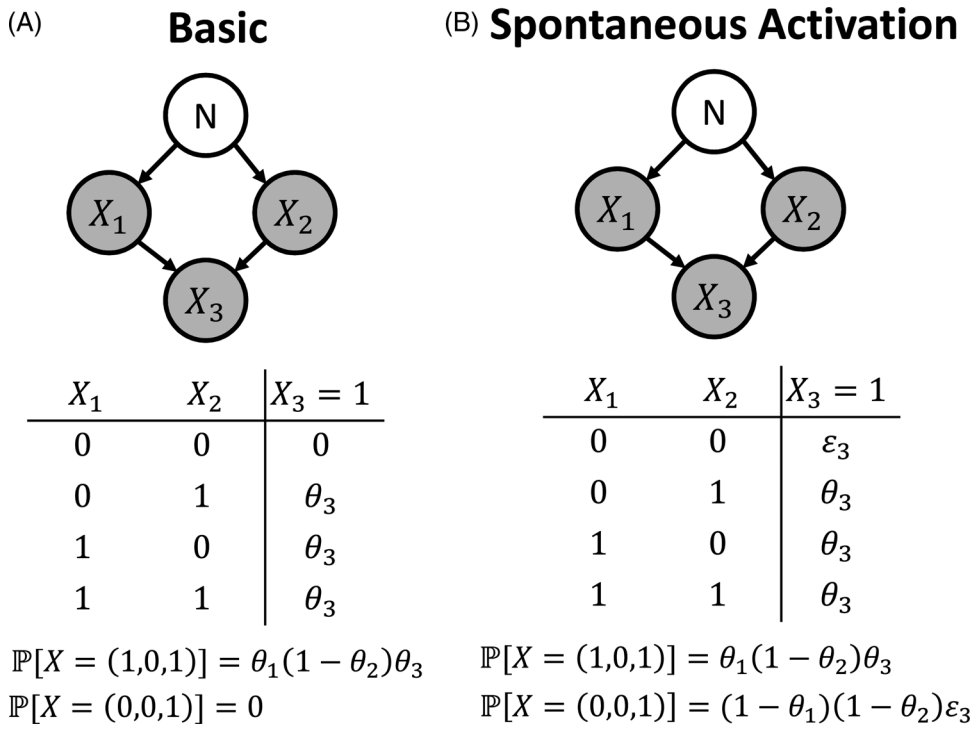

FIGURE 1 Bayesian networks of the cancer progression models investigated. Node N represents normal cell state, each random variable X_j is an observed alteration, and the corresponding progression probability parameter is θ_j. In all models, the conditional probability table of X_3 is shown, and probabilities of instance observations are computed. (A) Basic DBN model where further progression is impossible if none of the parent alterations have occurred. (B) Spontaneous activation model where there is a nonzero chance of a child occurring even if none of its parents are active

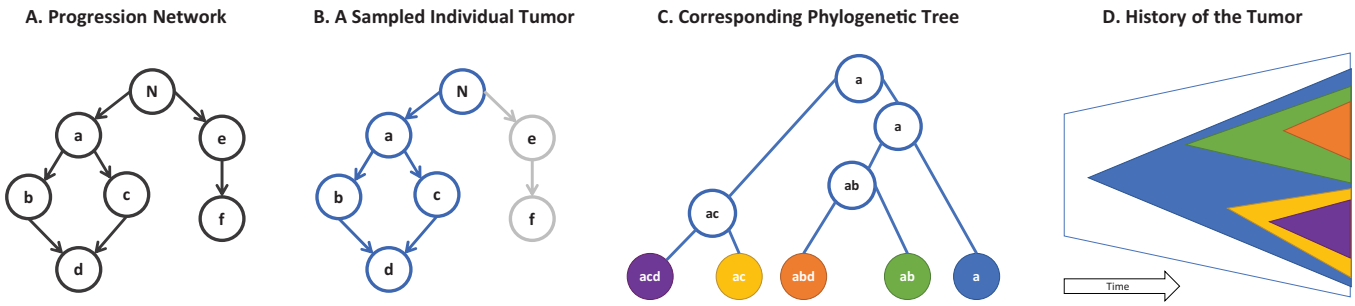

FIGURE 2 Phylogenetic interpretation of the DBN model. (A) A DBN progression network that models a cancer type at the population level (disease level). Root N represents the wild-type state (Normal) and there are six known driver alterations. (B) A sample from the network (blue nodes) that represents an individual tumor. (C) The corresponding phylogenetic tree of the sample. Each path of sampled graph forms a subclone living on the leaves of the phylogenetic tree that are distinguished by various colors. (D) Visualization of the tumor history and the subclonal relationships through time

1.2 | Our contribution

1.2.1 | Biological modeling

We propose the disjunctive Bayesian network (DBN), a population-level progression models (oncogenetic models), which has a phylogenetic interpretation. From the oncogenetic perspective, DBN relaxes the CBN progression assumption by allowing progression even if one of parents has occurred, see Figure 1A. From the phylogenetic perspective, each directed path starting from the wild-type root in a DBN graph may be interpreted as a (sub)clone (Figure 2C), and each sample from the DBN graph can be thought of as an individual tumor consisting of (sub)clones, see Figure 2B. Overall, the DBN itself is the overlay of all of the pos-

sible subclones corresponding to the modeled cancer (Figure 2A).

The DBN can naturally accommodate distinct progression paths for subtypes and is expressive enough to capture the mutual exclusivity of alterations present in the data. Therefore, one can skip two preprocessing steps necessary for the state-of-the-art models: stratifying samples by subtype and mutual exclusivity detection. We consider two extensions of DBN. The first extension relaxes the strict disjunction assumption and allows spontaneous (parent-less) alteration (Figure 1B). The second extension directly models measurement error of alterations. To have an uncluttered presentation, we present the measurement error model only in Supplement A. Note that each of N's children (Figure 1) should be considered as a root (parentless) mutation that happens early with (usually) substantial probability. Therefore, the progression probability θ_i of root mutations can be think of as spontaneous activation.

Although inherited predisposition and environmental mutagens can cause mutations and contribute to tumorigenesis, unavoidable errors associated with DNA replication have been suggested to be responsible for more than two-thirds of mutations in human cancers [28, 29]. The random nature of replication-caused mutations suggests that they can occur anywhere in the evolutionary history of the tumor and hitchhike in subclones or become clonal and fixate in the population. The spontaneous activation model (Figure 1B) is sufficiently flexible to accommodate both evolutionary constrained mutations that occur in a specific order and the occurrence of those alterations that are completely random.

1.2.2 | Computational efficiency

We provide an efficient dynamic programming (DP) implementation of an exact structure learning method [16] that learns the optimal DBN (in terms of a regularized likelihood). Additionally, this algorithm can be incorporated into existing cancer progression frameworks such as conjunctive BNs [11] or CAPRI [26], which will likely improve their accuracies. For rare cases that the studied driver alterations are numerous, we provided an efficient genetic algorithm (GA) in our software package. To speed up the GA's global search, we characterize a likelihood-equivalence relation over DBNs and only search through the representative DAGs of each class.

1.2.3 | Experimental performance

Through numerous synthetic and real data experiments, we show the ability of our algorithms in reconstructing ground truth progression networks from simulated samples and inferring biologically interpretable progression networks for cutaneous melanoma, lung adenocarcinoma, and bladder cancer. Our scalable **Onco**genetic **B**ayesian **N**etwork R package, **OncoBN**, provides two easy to use routines (approximate and exact) for estimation of oncogenetic BNs including DBN and CBN.

2 | METHODS

We model the observation of alterations as a binary random vector $(X_1, \dots, X_p)$, where $X_j = 1$ if the jth alteration is detected in the sample and $\mathbf{x} = (x_1, \dots, x_p)$ is an observed sample. We assume that a BN governs the order in which the events can occur. The BN consists of a DAG G and local *conditional probability distributions* (CPDs) $\mathbb{P}(x_j|\mathbf{x}(\mathcal{P}_j); \theta)$ where $\mathcal{P}_j$ is the set of parents of event j in G and θ parameterizes the distribution. Local CPDs form the joint distribution as $\mathbb{P}(\mathbf{x}; G, \theta) = \prod_{j=1}^{p} \mathbb{P}(x_j|\mathbf{x}(\mathcal{P}_j); \theta)$.

2.1 | Progression rule and parameter estimation

2.1.1 | Basic DBN

The DBN progression rule asserts that an event j occurs with probability θ_j if and only if at least one of its parents have occurred. Therefore, $\mathbb{P}(x_j = 1|\mathbf{x}(\mathcal{P}_j); \theta) = 0$ if parents are inactive and θ_j otherwise (Figure 1A).

2.1.2 | Spontaneous activation model

The deviation from the DBN progression rule may be the results of *spontaneous activation* caused by unknown sources. To capture that, we add a nonzero spontaneous activation probability $\varepsilon_j > 0$ for each node (Figure 1B).

Given n cross-sectional samples and the network G, we wish to find $\hat{\theta}_G$, the maximum likelihood estimator (MLE) for θ. We focus on the spontaneous activation model, where the likelihood is:

$$\mathcal{L}(\theta, G) = \mathbb{P}(\mathbf{x}; \theta, G) = \prod_{j=1}^{p} [\theta_j{}^{x_j}(1-\theta_j)^{1-x_j}]^{\mathbf{1}(\mathbf{x}(\mathcal{P}_j)\neq\mathbf{0})} \varepsilon_j^{\mathbf{1}(\mathbf{x}(\mathcal{P}_j)=\mathbf{0})}. \quad (1)$$

Maximizing the log-likelihood results in $\hat{\theta}_j^G = \frac{\sum_{i=1}^{n} \mathbf{1}(x_{ij}=1, \mathbf{x}_i(\mathcal{P}_j)\neq\mathbf{0})}{\sum_{i=1}^{n} \mathbf{1}(\mathbf{x}_i(\mathcal{P}_j)\neq\mathbf{0})}$ and where x_{ij} is the realization

of the jth event in the ith sample. From now on, to reduce the number of inferred parameters, we assume $\forall j : \varepsilon_j = \varepsilon$ and we fix it throughout the experiments. Details of parameter estimation for the three models (basic, spontaneous, measurement error) are presented in Supplement B.

2.2 | Exact structure learning

Although for a fixed network G, the MLE parameters have closed form, finding the best G is NP-hard. We present an efficient DP method for $p < 30$ that finds a best graph with maximum likelihood. We use "a best" instead of "the best" graph to emphasize on the fact that the graph with the maximum likelihood is not unique.

To have more interpretability and avoid overfitting, we restrict our search space to the space of p-node DAGs with an in-degree bound of k, $\mathcal{G}_{p,k}$. To further penalize dense graphs, we follow Ramazzotti et al. [26] and use the Bayesian information criterion (BIC) as our graph fitness score. The final optimization objective takes the following form:

$$\max_{G \in \mathcal{G}_{p,k}} \text{BIC}(G, \hat{\theta}^G), \quad \text{BIC}(G, \hat{\theta}^G) \triangleq \ell(G, \hat{\theta}^G) - \frac{\log(N)}{2}|E|. \tag{2}$$

2.2.1 | Dynamic programming algorithm

An exhaustive search of $\mathcal{G}_{p,k}$ takes superexponential time. Silander and Myllymäki [16] introduced a DP algorithm that can find the optimal network in exponential time. Their algorithm assumes that each graph G can be assigned a decomposable score $\text{Score}(G)$ such that $\text{Score}(G) = \sum_{j=1}^{p} \text{Score}_j(\mathcal{P}_j)$ where $\text{Score}_j(\mathcal{P}_j)$ is the score of the subgraph consisting of only vertex j and its parents $\mathcal{P}_j$. $\text{Score}_j(\mathcal{P}_j)$ is called the local score of j. For us, $\text{Score}(G) = \text{BIC}(G, \hat{\theta}^G)$ is our decomposable score. The rest of this section is devoted to a high-level summary of the algorithm.

Optimal substructure

First, note that each DAG has at least one sink node, which is a node with no outgoing edges. The score of a best graph $G^*(V)$ can be broken down to the best parents of any of its sinks s and a best subgraph obtained by removing s and its incoming edges. More formally, for s, an arbitrary sink of G^*, $\mathcal{P}_s^*$, should be a best set of parents, i.e., has highest local score $\mathcal{P}_s^* = \text{argmax}_{\mathcal{P}_s} \text{Score}_s(\mathcal{P}_s)$. In addition, for $G^*(V)$ to be optimal, $\text{Score}(G^*(V \backslash \{s\}))$ should also be optimal. This optimal substructure suggests the following recursive formula for finding a best sink for set of nodes $W \subseteq V$:

$$\text{Sink}^*(W) = \underset{s \in W}{\text{argmax}}\, \text{Score}_s(\mathcal{P}_s^*(W)) + \text{Score}(G^*(W \backslash \{s\})), \tag{3}$$

where $\mathcal{P}_s^*(W) = \text{argmax}_{\mathcal{P}_s \in W} \text{Score}_s(\mathcal{P}_s)$ is the precomputed best parents of s in W. Best sinks can be computed in $O(n2^{n-1})$ time using memorization.

Reconstructing an optimal solution

Best sinks immediately result in a best ordering of nodes in reverse order. By having an optimal order and the best set of parents for all nodes, it is straightforward to build an optimal graph. Starting from an empty graph, we add a node according to the optimal order and add incoming edges from its optimal parents that preexist in the graph.

Computational complexity

The most intensive portion of the algorithm is computing the set of best parents $\mathcal{P}_s^*(W)$ for every $W \subseteq V \backslash \{s\}$. This step requires $O(n^2 2^{n-1})$ time and $O(n2^{n-1})$ space. By leveraging disk space, it is possible to implement the algorithm such that at most 2^{n+2} bytes of RAM are occupied at any given time.

2.2.2 | Pruning spurious edges

When the data are corrupted by noise, the estimated graph is likely to contain spurious edges. To remove low confidence edges, we perform statistical tests on the estimated graph. In DBNs, if $e = (u, v)$ is an edge in the ground-truth graph, we have $\mathbb{P}(X_u = 1 \mid X_v = 1) > \mathbb{P}(X_u = 1 \mid X_v = 0)$. Thus, we use the Fisher's exact test to check the inequality and retain edges for which the inequality holds with high confidence.

2.3 | Approximate structure learning

For large number of mutations, the exhaustive search is infeasible. Here we propose a GA to approximate the global maximum to the log-likelihood function l for $p > 30$. The pseudocode of this part is summarized in Algorithm 1.

2.3.1 | Genetic algorithm

GAs search for a global optimum using a "survival of the fittest" strategy. We begin with a population of $2C$ candidate solutions known as *chromosomes* and evolve them

Algorithm 1 Genetic Algorithm of OncoBN Package

1: **input**: Data set $\mathcal{D}$, parameters C, T, and $r \geq 0$.
2: **output**: Inferred graph $\hat{G}$
3: Generate population of random trees: $S_0 = \{G_i^0\}_{i=1}^{2C}$.
4: **for** $t = 1$ to T **do**
5: Compute fitness score of each DAG as: $\upsilon_i^t = \ell(G_i^t; \hat{\theta}_{G_i^t}^{\text{MLE}}, \mathcal{D})$
6: **if** $r = 0$ **then** ▷ MDL penalty
7: $\upsilon_i^t = \upsilon_i^t + \log n \log p \sum_{j \in G_i^t} |\mathcal{P}_j|$
8: **end if**
9: $\mathbf{v}^t = \frac{(\upsilon_1^t, \upsilon_2^t, \ldots, \upsilon_{2C}^t)}{\sum_j^{2C} \upsilon_j^t}$ ▷ Selection probabilities
10: **for** $i = 1$ to S **do**
11: $(G_i^t, G_{i+1}^t) \leftarrow \text{Selection}(\mathbf{v}^t, 2)$ ▷ Select DAGs
12: $(G_i^{t+1}, G_{i+1}^{t+1}) \leftarrow \text{Crossover}(G_i^t, G_{i+1}^t)$
13: $G_i^{t+1} \leftarrow \text{Mutate}(G_i^{t+1}, r)$
14: $G_{i+1}^{t+1} \leftarrow \text{Mutate}(G_{i+1}^{t+1}, r)$
15: $G_i^{t+1} \leftarrow \Pi_{\sim}(G_i^{t+1}); G_{i+1}^{t+1} \leftarrow \Pi_{\sim}(G_{i+1}^{t+1})$
16: **end for**
17: **end for**
18: Return the $\hat{G}$ corresponding to $\upsilon_{\max} = \max_{t \in [T], j \in [2C]} \upsilon_j^t$

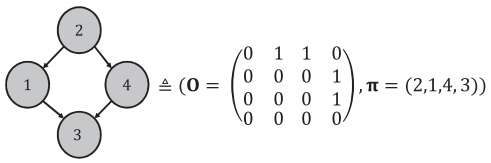

FIGURE 3 DAG representation. The DAG can be decomposed into an upper triangular matrix **O** along with a permutation $\boldsymbol{\pi}$

for T *generations*. Each chromosome is assigned a *fitness value* υ that determines its quality. Then, S chromosome pairs are selected preferentially according to their fitness for reproduction. The next generation forms by performing a *crossover operation* on chromosome pairs. In each generation, there is a chance that a *mutation operation* changes chromosomes. In the setting of our model, chromosomes at generation t are $2C$ DAGs, $\{G_i^t\}_{i=1}^{2C}$ and the fitness of each DAG is its maximum likelihood value.

Representation

The most natural way to encode a DAG G is by using its adjacency matrix **A**. However, perturbing the entries in **A** may unintentionally introduce directed cycles into the resulting graph. To avoid this problem, following Carvalho [30], we represent G with a pair $(\mathbf{O}, \boldsymbol{\pi})$, where **O** is the adjacency matrix for the *topological ordering* of G (i.e., a strictly upper triangular matrix), and $\boldsymbol{\pi}$ is a permutation vector describing how the vertices of **O** should be relabeled to generate **A** (Figure 3). We consider the ordering **O** and permutation $\boldsymbol{\pi}$ as separate chromosomes and evolve each of them individually. We can avoid introducing directed cycles by ensuring that our genetic operators always return an upper triangular matrix.

Operations

Each crossover operation is defined to take in two DAGs and produce two offspring to keep the generation size constant. The orderings and permutations are crossed over separately (Supplement C). To maintain diversity in the population, we also define three mutation operators: edge, branch, and permutation (Supplement C).

2.3.2 | Speeding up the GA with DAG equivalence classes

As mutation i activates with probability θ_i *irrespective* of which parent mutations are active, many different network structures induce the same probability distribution over $\{0,1\}^p$. We say that $G \sim G'$ if, for every $\boldsymbol{\theta}$ and $\mathbf{x}$, $\mathbb{P}(\mathbf{x}; G, \boldsymbol{\theta}) = \mathbb{P}(\mathbf{x}; G', \boldsymbol{\theta})$. It is clear that $\sim$ defines an equivalence relation over DAGs. To make the GA more efficient, we search only one DAG per equivalence class by defining a *canonical form* for each graph. Figure 4 gives an example of equivalent networks. Algorithmically, we project back new solution graphs to the state space of canonical forms by removing redundant edges and uniquely labeling similar vertices in function $\Pi_{\sim}(\cdot)$ (line 12 of Algorithm 1). More details on mathematical properties of DBNs is presented in Supplement D.

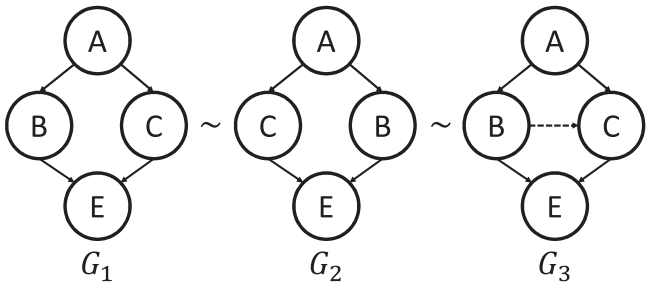

FIGURE 4 Examples of DAGs from the same equivalence class and their canonical form. For all θ and $\mathbf{x}$, $\mathbb{P}(\mathbf{x};\theta)$ is the same for all of the three network structures shown above. B and C are similar vertices in G_1 and G_2. Edge $B \to C$ is redundant in G_3. By uniquely labeling similar vertices and removing redundant edges, we reach G_1 as the canonical form of the other two DAGs

2.3.3 | Controlling complexity

To prevent overfitting, we consider two types of penalty to control the complexity of the learned BN. First, if $r = 0$ in Algorithm 1, we perform regularized MLE by using the minimum description length penalty introduced in Lam and Bacchus [31] that simplifies to $\log n \log p \sum_{j=1}^{p} |\mathcal{P}_j|$ for the DBN. In another approach represented by $r > 0$ in Algorithm 1, we limit the number of parents of each node to r, i.e., $\max_j |\mathcal{P}_j| \leq r$.

3 | RESULTS

3.1 | Inferring simulated ground truths

To test the DP method against existing cancer progression algorithms, we generate datasets from simulated networks. Random graphs G are created using the PCALG R package [32], which allows the user to specify the number of vertices and the average degree. For network parameters, we sample $\theta_j \sim \text{Unif}(0.25, 0.75)$. Once θ_js and G are known, a simulated dataset can be created by iterating over a topological sort of G. For tests on simulated data, we fix the number of observations n to be 400 and the number of alterations p to be 20 (this is similar to the size of existing cancer datasets). Unless specified otherwise, the average degree is set to 3. To simulate the noise that is likely present in real data, we flip the binary value of each entry with probability η.

If $\hat{G} = (V, \hat{E})$ is the estimated network with ground truth $G = (V, E)$, one can define a false-positive edge to be an edge $e \in \hat{E}$ with $e \notin E$ and false-negative edges similarly. As the number of possible false positives is likely much larger than the number of possible false negatives, we assess performance using Matthew's correlation coefficient (MCC), which is robust under uneven class sizes [33]. The MCC can be computed as

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}, \tag{4}$$

where TP (FP) is the number of true (false) positives and TN (FN) is the number of true (false) negatives. An MCC of 1 corresponds to perfect reconstruction, whereas an MCC of 0 means that the algorithm is outputting a random network.

The DP algorithm requires that the spontaneous activation rate ε and in-degree bound k are chosen in advance. We suggest (and use) the following heuristic to set ε: set $\varepsilon = f_m/2$, where f_m is the frequency of the *least* frequent alteration. One should always select $\varepsilon < f_m$, as otherwise there may be incentive to misplace the node corresponding to this alteration. In the interest of efficiency, we set $k = 5$, although in theory, one could test every possible k to select the one that best trades expressivity for complexity. For pruning spurious edges, Fisher's exact test with significance level of 10^{-5} is used.

First, we compare the DP algorithm to CBN. The original approach of Gerstung et al. [11] uses simulated annealing to approximate the network structure alongside a computationally expensive expectation-maximization (EM) algorithm for parameter estimation. As a result, their method is only applicable when the number of mutations is less than 12. Montazeri et al. [34] address this issue by developing an efficient Monte Carlo algorithm, named MC-CBN, to estimate the parameters and structure of a CBN. Figure 5A compares MC-CBN and the DP algorithm for various choices of $\eta \in [0, 0.2]$. In the case of low error ($\eta \approx 0$), both methods are extremely accurate. However, as η becomes larger, the MCC for MC-CBN drops to 0 at a faster rate.

Next, we compare the DP algorithm to CAPRI [26]. CAPRI is a flexible framework for inferring cancer progression networks that can account for many types of interactions between nodes. CAPRI first applies a constraint-based algorithm to obtain a *prima facie* network, and then applies a local search algorithm to prune spurious edges. CAPRI is available through TRONCO [24]. Figure 5A compares the ability of CAPRI and the DP algorithm to recover networks with various levels of noise. We also compare the methods in terms of normalized hamming distance between the adjacency matrices of inferred networks and the ground truth, which is equivalent to (1 − accuracy) of reconstruction. We report the comparison results in Supplement E. The relative relationship of methods performance in terms of accuracy is very similar to their MCC counterparts. To understand how the algorithms perform as network complexity increases, Figure 5B

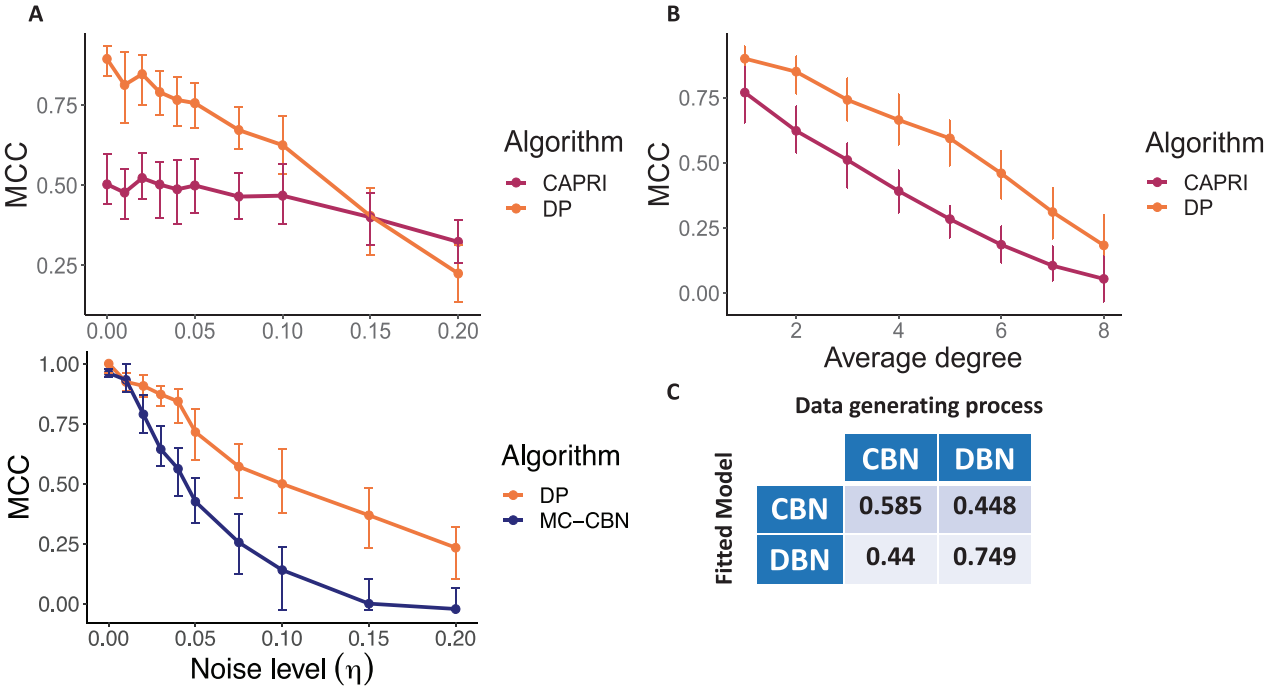

FIGURE 5 Comparison to existing cancer progression algorithms on simulated data. (A) Comparing the DP algorithm to CAPRI and MC-CBN for various noise rates. For each value of η, 100 datasets were generated. Points represent the median MCC over all trials and error bars give interquartile range. (B) Comparing the DP algorithm to CAPRI for various levels of network complexity. Network complexity was quantified by varying the average degree of random graphs from 1 to 8. One hundred datasets for each degree were generated. (C) Cross-comparison between DBN and CBN. One hundred datasets were generated from the CBN model and 100 datasets were generated from the DBN model. Then the DBN and CBN models were fitted to all of the datasets, and the mean MCC in each class is reported

varies the average degree while keeping the noise constant at $\eta = 0.05$.

We perform a cross-comparison of the CBN and DBN models. To do this, we simulate 100 datasets from the CBN model and 100 datasets from the DBN model. We fit the DBN model to the CBN datasets and vice versa. Figure 5C reports the mean MCC in each category.

We benchmarked the GA with 2C=100 solutions, which are initialized as random trees, and compared the results with CAPRI for various noise levels. The results are shown in Figure S3 of the Supplement. With our simple random initialization, CAPRI outperformed the GA on all noise levels.

3.2 | Real data experiment

We use our method to recover the order of *driver* mutations in three cancer types from The Cancer Genome Atlas (TCGA) program [35]. We selected Skin Cutaneous Melanoma (SKCM) and Lung Adenocarcinoma (LUAD) because there are known molecular subtypes and mutual exclusivity relationships characterized for them. To determine the driver mutations, we used results from Bailey et al. [36] where 26 computational methods had been applied to the TCGA data. The number of resulted driver mutations for SKCM and LUAD are below 30 and therefore exact DP method of Section 2.2 is applicable. We chose the Bladder Cancer (BLCA) for our third experiment because it has the highest driver mutation rate per sample in the TCGA dataset [36] and therefore is suitable to check the scalability of our proposed GA. Number of samples, driver mutations, and frequent driver mutations (5% frequency cutoff threshold) for each cancer type are listed in Table 1. The last column of Table 1 shows the median number of frequent driver mutations per patient that reveals that each tumor needs only a few driver hits. The small number of per patient drivers in the selected cancer types is aligned with the fact that typically, three driver mutations are required to convert normal cells into cancer cells [37].

To quantify our uncertainty in the estimated progression network, we run the algorithm on 100 bootstrapped datasets. We form the *mean graph* by only reporting the edges that are present in a sufficiently large number of networks estimated from the bootstrapped datasets (this cutoff will be 25 or 50). Finally, to get a better sense of how cancer of each individual patient progress, we map a random sample of patients' mutations onto their corresponding inferred cancer progression network. These results are presented in Section F of the Supplement.

TABLE 1 Number of samples (n), number of driver mutations, and number of frequent driver mutations (5% frequency cutoff threshold) (p) for the three used TCGA cancer types. The rightmost column shows the median number of frequent driver mutation per patient

	# samples	# drivers	# freq. drivers	median # of freq. drivers / patient
SKCM	467	20	15	3
LUAD	567	24	14	3
BLCA	414	45	31	5

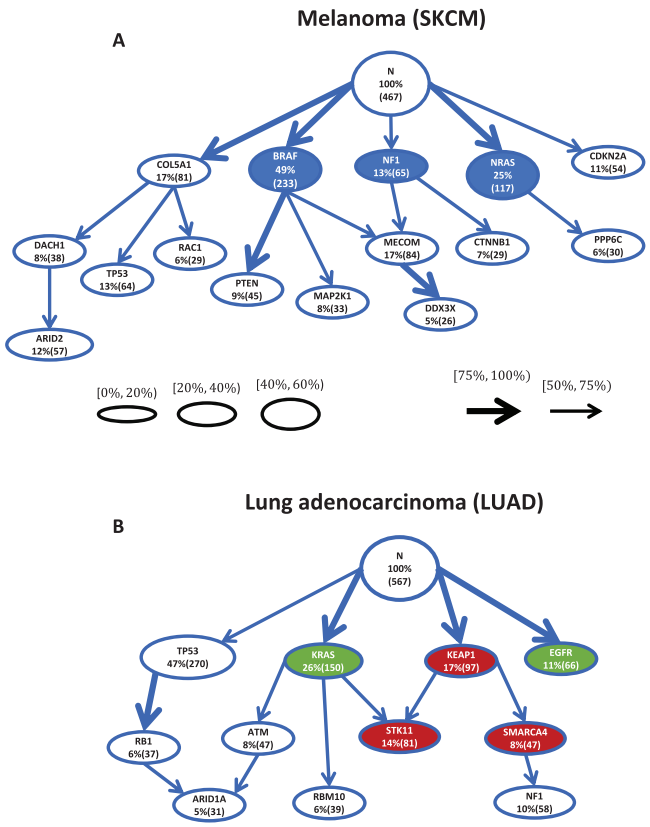

FIGURE 6 Mutation progression networks of melanoma and lung adenocarcinoma inferred by the exact dynamic programming learning method. (A) We recover three known subtypes of melanoma (*BRAF*, *NRAS*, and *NF1* in blue) as separate roots. Mutations linked to metastasis such as *PTEN* and *DDX3X* are captured as late events. (B) Synthetically lethal mutations of LUAD, *KRAS* and *EGFR* (green nodes), appear in disjoint branches. Frequently co-occurred mutations *STK11*, *KEAP1*, and *SMARCA4* occupy a branch of the inferred network (red nodes). Subtype defining mutations *TP53* and *RB1* are ordered with high confident

3.2.1 | Progression of mutations in cutaneous melanoma and lung adenocarcinoma

We run the DP method of the OncoBN package on 100 bootstrapped datasets with the in-degree bound of $k = 3$ and fixed universal spontaneous activation probability of $\varepsilon = 0.025$. Refer to Section 3.1. for more information on how we select these values for the constants. The mean progression network is illustrated in Figure 6. Note that out of 24 LUAD mutations, only 11 of them are present in the mean progression network. This is because the rest of mutation are not connected with enough confident to the other nodes or to each others.

For SKCM, we recovered three root mutations with a high mean presence: *BRAF*, *NRAS*, and *COL5A1*. In the rest of the graph, two connections have highest confident: *BRAF*→*PTEN* and *MECOM*→*DDX3X*. The only mutation with multiple parents is *MECOM*. For LUAD, three high confident roots have been recovered: *KRAS*, *KEAP1*, and *EGFR* plus a high confident edge *TP53*→*RB1*. *STK11* and *ARID1A* each have two parents.

3.2.2 | Progression of mutations in bladder cancer

We run the GA of OncoBN package with $2S = 100$ solutions for $T = 300$ generations on 100 bootstrap datasets. The mean progression network is illustrated in Figure 7. Out of $p = 31$ nodes, only 18 are inferred in the mean progression network because the remaining 13 are not connected with enough confident to the rest or to each others.

We recover three root mutations with a high mean presence for the progression of bladder cancer: *TP53*, *KDM6A*, and *KMT2D*. From the several children of these roots, three have a mean presence greater than 50%: *RB1*, *STAG2*, and *KMT2C*. Finally, roots with meager mean presence (*ELF3*, *ATM*, and *CREBBP*) and childless *PIK3CA* are mutations for which OncoBN cannot find enough supporting evidence to place them in the main progression graph. Note that these placements are possible because of the flexibility of the spontaneous activation model.

3.2.3 | Comparison with CAPRI on real data

We run CAPRI on the real data and present the inferred networks in the Supplement Figure S8. CAPRI uses AIC and BIC methods to penalize the complexity of the inferred progression network. As the penalty of BIC is larger than AIC, its corresponding graph is sparser. Qualitatively, we find the AIC network to be very dense and hard to interpret and the BIC network very sparse such that it fails to capture

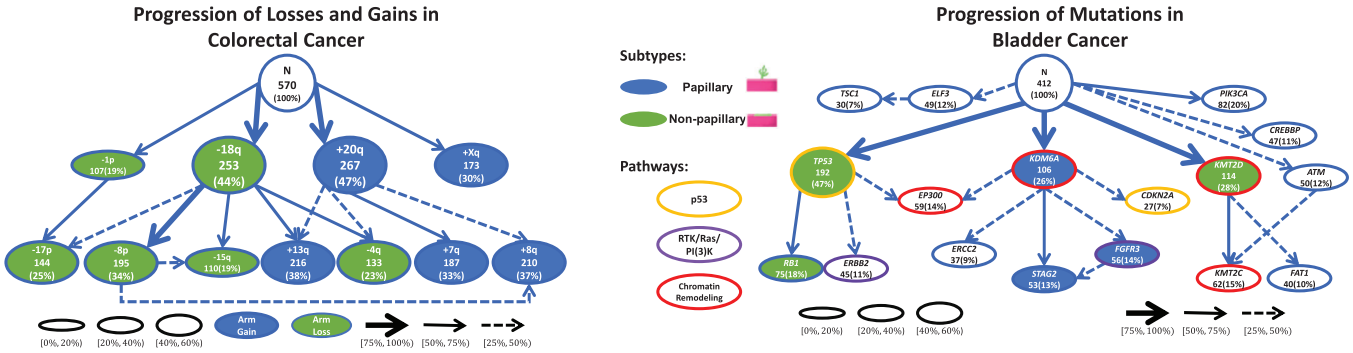

FIGURE 7 Mutation progression network of bladder cancer inferred by OncoBN. Focusing on the three high confidence roots (*TP53*, *KDM6A*, and *KMT2D*), the two subtypes of bladder cancer are clearly separated. The middle subgraph (rooted in *KDM6A*) is enriched for hallmark aberrations of the papillary subtype (blue nodes), and the other two subgraphs correspond to flat tumors (green nodes). Known mutual exclusive alteration pairs such as (*KDM6A*, *KMT2D*) and (*TP53*, *CDKN2A*) are occurring in different subgraphs. Four established highly perturbed pathways of bladder cancer are represented with varying outline colors. Each subtype has at least one mutated gene from these pathways is its subgraphs; therefore in both subtypes, all of the four pathways are perturbed

many known biological facts. More discussion on CAPRI results is presented in Supplement Section H.

4 | DISCUSSION

4.1 | Simulation study

Figure 5 shows that the DP algorithm outperforms existing cancer progression algorithms when the noise rate is small. For high noise rates ($\eta \approx 0.2$), CAPRI is slightly more accurate. A future improvement to the method could be to integrate some of CAPRI's regularization steps to improve robustness to noise. When η is small and fixed, DP uniformly outperforms CAPRI at different levels of network complexity (Figure 5B).

The cross-comparison (Figure 5C) shows that the DBN model is adequate even when the underlying data generating process assumes the CBN model. Although the CBN model performs slightly better when the data-generating process assumes the CBN model (MCC 0.585 vs. MCC 0.44), the DBN model is significantly better when the data-generating process assumes the DBN model (MCC 0.749 vs. MCC 0.448).

The GA is an approximate algorithm, and its reconstruction ability largely depends on the number of starting solutions (BN structures) and their quality. In comparison with CAPRI in the simulation study (Figure S3), the poor performance of the GA suggests that for the large graphs, one needs to initialize the solution population better and increase the population size substantially to improve the quality of the final inferred network. One future direction for better initialization of the GA is to use the heuristics provided in other methods [6–8], which reconstruct tree networks based on the nodes' frequencies and co-occurrence.

4.2 | Melanoma and lung adenocarcinoma

Inferred melanoma progression network captures multiple known characteristics of melanoma. Namely, there are three distinct known molecular subtypes for cutaneous melanoma with *BRAF*, *NRAS*, and *NF1* as biomarkers [38]. All three of these mutations are roots of our inferred progression network, which suggests that they are important early occurring events. Strong metastasis inducing cooperation of *PTEN* with *BRAF* [39] is captured with *BRAF*→*PTEN*. *DDX3X* that is linked with metastasis in melanoma is captured as a late-stage event [40].

BRAF exhibits mutual exclusivity relationship with *NRAS* and *NF1*. More specifically, hotspot mutations in the V600 codon of *BRAF* and Q61 codon of *NRAS* have been observed to be mutually exclusive even in single-cell level [41] and are suggested to be synthetically lethal [42, 43]. On the other hand, nonhotspot mutations of each of the three subtype-determining genes can co-occur with hotspot mutations of other ones [38]. Due to the limited number of samples, we do not separate hotspot and nonhotspot mutations but because in the inferred network, they are separate roots of the progression, their co-occurrence probability is extremely small. For example, in our inferred progression network, the co-occurrence probability of *BRAF* and *NF1* is less than 0.05, and in our dataset of 467 samples, they co-occur nine times ($\approx 2\%$). More interestingly, the *BRAF* and *NRAS* branches in Figure 6A do not share any child and therefore based on the phyloge-

netic interpretation of DBN (Figure 2C) although they can co-exist in a tumor, they may not co-occur at the cell level. This finding is aligned with the previous single-cell analysis of melanoma cell lines harboring both *NRAS* and *BRAF* [41].

In the inferred progression network of lung adenocarcinoma, synthetically lethal mutations *KRAS* and *EGFR* [44] appear as distinct roots. Moreover, *KRAS*, *KEAP1*, *STK11*, *SMARCA4*, and *NF1* form a subgraph. It is known that *KRAS*, *KEAP1*, *STK11*, and *SMARCA4* co-occur in nonsmall cell lung cancers [45] and our algorithm suggests that *KRAS* and *KEAP1* are early events in those tumors.

4.3 | Bladder cancer

The recovered progression network for bladder cancer reflects existing biological research. First, bladder cancer is known to have two histologically different subtypes known as papillary and nonpapillary [46]. Papillary tumors are finger-like, which start in the lining and grow toward the center of the bladder. Nonpapillary tumors also initiate in the lining but are flat in shape. Both types can be muscle-invasive, which means the tumor has grown outward, escaped the lining, and infiltrated bladder muscles, or non-muscle invasive [46]. All of the bladder cases in TCGA are muscle-invasive, but papillary and nonpapillary cases are not known.

There are known molecular signatures for papillary and nonpapillary bladder cancers. Mutations in *TP53*, *RB1*, and *KMT2D* (green nodes in Figure 7) are very frequent in nonpapillary subtype, whereas *KDM6A*, *STAG2*, and *FGFR3* (blue nodes in Figure 7) are hallmarks of papillary tumors [47–50]. Focusing on the high confident recovered roots (*TP53*, *KDM6A*, and *KMT2D*) and their descendants, our inferred network of Figure 7 shows separate progression paths for papillary and nonpapillary subtypes. The middle subgraph rooted at *KDM6A* contains *KDM6A*, *STAG2*, and *FGFR3* mutations and is mostly separated from the rest of the network. Therefore, we can match it to the progression of the papillary subtype. Subgraphs on the right and left of the figure (rooted at *TP53* and *KMT2D*) are enriched with molecular hallmarks of nonpapillary subtype. Our result shows the ability of OncoBN to infer the cancer progression network while maintaining subtype-specific biology.

In addition, we know that usually, single perturbation of a pathway is enough for the manifestation of a cancer hallmark. Therefore, another mutated gene in the same pathway does not confer a selective advantage. Thus, patterns of mutual exclusivity of cancer events arise among genes in the same pathways. In bladder cancer, high rate of alteration of p53/Rb, RTK/Ras/PI(3)K, and histone modification pathways are observed [47]. Figure 7 highlights the corresponding pathways of genes with different outline color for each pathway. It confirms that the two subtypes (papillary and nonpapillary) have perturbation in p53, RTK/Ras/PI(3)K, methylation, and acetylation pathways. The only mutation that is shared between the two subtypes is *EP300*, which is a chromatin remodeling protein and more specifically a histone-acetyltransferase. Mutation in the histone-acetyltransferase domain of *EP300* is associated with a higher tumor mutation burden and promotes antitumor immunity in bladder cancer and indicated a favorable clinical prognosis [51, 52].

5 | CONCLUSION

We presented the DBN, a flexible cancer progression model, which has a phylogenetic interpretation. Through synthetic and real experiments, we showed that DBN is flexible enough to capture trajectories of cancer subtypes and mutual exclusivity patterns of alterations. In our R package, OncoBN, we provided two inference methods for learning cancer progression networks. Our exact DP method learns the DBN efficiently when the number of alterations is small, whereas our supplemented GA approximates the most likely structure for larger networks. Our model provides an avenue to integrate oncogenetic (population-level cancer progression) findings to improve tumor phylogenetic inference (patient-level progression) in the future.

ORCID

Amir Asiaee https://orcid.org/0000-0002-5317-9820

References

1. D. Hanahan and R. A. Weinberg, *Hallmarks of cancer: the next generation*, Cell **144** (2011), no. 5, 646–674. ISSN 0092-8674, 1097-4172. https://doi.org/10.1016/j.cell.2011.02.013.
2. P. M. Altrock, L. L. Liu, and F. Michor, *The mathematics of cancer: integrating quantitative models*, Nat. Rev. Cancer **15** (2015), no. 12, 730–745.
3. A. Davis, R. Gao, and N. Navin, *Tumor evolution: Linear, branching, neutral or punctuated?*, Biochim. Biophys. Acta. Rev. Cancer **1867** (2017), no. 2, 151–161.
4. M. Gerstung et al., *The evolutionary history of 2,658 cancers*, Nature **578** (2020), no. 7793, 122–128.
5. E. R. Fearon and B. Vogelstein, *A genetic model for colorectal tumorigenesis*, Cell **61** (1990), no. 5, 759–767.
6. R. Desper et al., *Inferring tree models for oncogenesis from comparative genome hybridization data*, J. Comput. Biol. **6** (1999), no. 1, 37–51.
7. N. Beerenwinkel et al., *Learning multiple evolutionary pathways from cross-sectional data*, J. Comput. Biol. **12** (2005a), no. 6, 584–598.

8. N. Beerenwinkel et al., *Mtreemix: a software package for learning and using mixture models of mutagenetic trees*, Bioinformatics **21** (2005b), no. 9, 2106–2107.
9. D. Barber, *Bayesian reasoning and machine learning*, Cambridge University Press, 2012.
10. N. Beerenwinkel, N. Eriksson, and B. Sturmfels, *Conjunctive Bayesian networks*, Bernoulli **13** (2007), no. 4, 893–909.
11. M. Gerstung et al., *Quantifying cancer progression with conjunctive Bayesian networks*, Bioinformatics **25** (2009), no. 21, 2809–2815.
12. I. Dagogo-Jack and A. T. Shaw, *Tumour heterogeneity and resistance to cancer therapies*, Nat. Rev. Clin. Oncol. **15** (2018), no. 2, 81–94.
13. D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*, MIT Press, 2009.
14. H. S. Farahani and J. Lagergren, *Learning Oncogenetic Networks by Reducing to Mixed Integer Linear Programming*, PLoS One **8** (2013), no. 6, e65773.
15. H. Montazeri et al., *Large-scale inference of conjunctive Bayesian networks*, Bioinformatics **32** (2016b), no. 17, i727–i735.
16. T. Silander and P. Myllymäki, *A simple approach for finding the globally optimal Bayesian network structure*, Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence, 2006, pp. 445–452.
17. M. D. M. Leiserson et al., *CoMEt: a statistical approach to identify combinations of mutually exclusive alterations in cancer*, Genome Biol. **16** (2015), 160.
18. Y. Deng et al., *Identifying mutual exclusivity across cancer genomes: Computational approaches to discover genetic interaction and reveal tumor vulnerability*, Briefings Bioinf. **20** (2019), no. 1, 254–266.
19. Y.-K. Cheng et al., *A mathematical methodology for determining the temporal order of pathway alterations arising during gliomagenesis*, PLoS Comput. Biol. **8** (2012), no. 1, e1002337.
20. M. Gerstung et al., *The temporal order of genetic and pathway alterations in tumorigenesis*, PLoS One **6** (2011), no. 11, e27136.
21. S. Cristea, J. Kuipers, and N. Beerenwinkel, *pathTiMEx: joint inference of mutually exclusive cancer pathways and their progression dynamics*, J. Comput. Biol. **24** (2017), no. 6, 603–615.
22. B. J. Raphael and F. Vandin, *Simultaneous inference of cancer pathways and tumor progression from cross-sectional mutation data*, J. Comput. Biol. **22** (2015), no. 6, 510–527.
23. G. Caravagna et al., *Algorithmic methods to infer the evolutionary trajectories in cancer progression*, Proc. Natl. Acad. Sci. **113** (2016), no. 28, E4025–34.
24. L. De Sano et al., *TRONCO: an R package for the inference of cancer progression models from heterogeneous genomic data*, Bioinformatics **32** (2016), no. 12, 1911–1913.
25. L. O. Loohuis et al., *Inferring tree causal models of cancer progression with probability raising*, PLoS One **9** (2014), no. 10, e108358.
26. D. Ramazzotti et al., *CAPRI: efficient inference of cancer progression models from cross-sectional data*, Bioinformatics **31** (2015), no. 18, 3016–3026.
27. D. Ramazzotti et al., *Modeling cumulative biological phenomena with Suppes-Bayes causal networks*, Evol. Bioinform. Online **14** (2018), 1176934318785167.
28. C. Tomasetti and B. Vogelstein, *Variation in cancer risk among tissues can be explained by the number of stem cell divisions*, Science **347** (2015), no. 6217, 78–81.
29. C. Tomasetti, L. Li, and B. Vogelstein, *Stem cell divisions, somatic mutations, cancer etiology, and cancer prevention*. Science **355** (2017), no. 6331, 1330–1334.
30. A. Carvalho, A cooperative coevolutionary genetic algorithm for learning Bayesian network structures, May 2013.
31. W. Lam and F. Bacchus, *Learning bayesian belief networks: An approach based on the mdl principle*, Comput. Intell. **10** (1994), no. 3, 269–293.
32. M. Kalisch et al., *Causal inference using graphical models with the r package pcalg*, J. Stat. Softw. **47** (2012), no. 11, 1–26.
33. B. W. Matthews, *Comparison of the predicted and observed secondary structure of t4 phage lysozyme*, Biochim. Biophys. Acta (BBA)-Protein Struct. **405** (1975), no. 2, 442–451.
34. H. Montazeri et al., *Large-scale inference of conjunctive Bayesian networks*, Bioinformatics **32** (2016a), no. 17, i727–i735.
35. Cancer Genome Atlas Research Network et al., *The cancer genome atlas Pan-Cancer analysis project*, Nat. Genet. **45** (2013), no. 10, 1113–1120.
36. M. H. Bailey et al., *Comprehensive characterization of cancer driver genes and mutations*, Cell **173** (2018), no. 2, 371–385.e18.
37. J. G. Reiter et al., *An analysis of genetic heterogeneity in untreated cancers*, Nat. Rev. Cancer, **19** August (2019), no. 11, 639–650.
38. The Cancer Genome Atlas Network, *Genomic classification of cutaneous melanoma*, Cell **161** (2015), no. 7, 1681–1696.
39. D. Dankort et al., *Braf(V600E) cooperates with Pten loss to induce metastatic melanoma*, Nat. Genet. **41** (2009), no. 5, 544–552.
40. B. Phung et al., *The X-linked DDX3X RNA helicase dictates translation reprogramming and metastasis in melanoma*, Cell Rep. **27** (2019), no. 12, 3573–3586.e7.
41. M. Sensi et al., *Mutually exclusive NRAS Q61R and BRAF V600E mutations at the single-cell level in the same human melanoma*, Oncogene **25** (2006), no. 24, 3357–3364. ISSN 1476-5594.
42. R. Kumar et al., *Growth suppression by dual BRAF(V600E) and NRAS(Q61) oncogene expression is mediated by SPRY4 in melanoma*, Oncogene **38** (2019), no. 18, 3504–3520.
43. C. Petti et al., *Coexpression of NRASQ61R and BRAFV600E in human melanoma cells activates senescence and increases susceptibility to cell-mediated cytotoxicity*, Cancer Res. **66** (2006), no. 13, 6503–6511.
44. A. M. Unni et al., *Evidence that synthetic lethality underlies the mutual exclusivity of oncogenic KRAS and EGFR mutations in lung adenocarcinoma*, eLife **4** (2015), e06907.
45. A. J. Schoenfeld et al., *The genomic landscape of SMARCA4 alterations and associations with outcomes in patients with lung cancer*, Clin. Cancer Res. **26** (2020), no. 21, 5701–5708.
46. A. M. Kamat et al., *Bladder cancer*, The Lancet **388** (2016), no. 10061, 2796–2810.
47. Cancer Genome Atlas Research Network, *Comprehensive molecular characterization of urothelial bladder carcinoma*, Nature **507** (2014), no. 7492, 315–322.
48. C. P. N. Dinney et al., *Focus on bladder cancer*, Cancer Cell **6** (2004), no. 2, 111–116.
49. Y. Gui et al., *Frequent mutations of chromatin remodeling genes in transitional cell carcinoma of the bladder*, Nat. Genet. **43** (2011), no. 9, 875–878.
50. D. A. Solomon et al., *Frequent truncating mutations of STAG2 in bladder cancer*, Nat. Genet. **45** (2013), no. 12, 1428–1430.
51. R. Krupar et al., *In silico analysis reveals EP300 as a panCancer inhibitor of anti-tumor immune response via metabolic modulation*, Sci. Rep. **10** (2020), no. 1, 9389.

52. G. Zhu et al., *EP300 mutation is associated with tumor mutation burden and promotes antitumor immunity in bladder cancer patients*, Aging **12** (2020), no. 3, 2132–2141. ISSN 1945-4589.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of the article.

How to cite this article: Nicol PB, Coombes KR, Deaver C, Chkrebtii O, Paul S, Toland AE, Asiaee A. *Oncogenetic network estimation with disjunctive Bayesian networks*. Comp. Sys. Onco. 2021;**1**:e1027. https://doi.org/10.1002/cso.21027