Inference of Cancer Progression Models with Biological Noise

Ilya Korsunsky
Department of Computer Science
Courant Institute for Mathematical Sciences, NYU
715 Broadway Room 1012
New York, NY 10003

Daniele Ramazzotti
Dipartimento di Informatica, Sistemistica e Comunicazione
Universitá degli Studi di Milano-Bicocca,
Viale Sarca 336, 20126 Milano

Giulio Caravagna
Dipartimento di Informatica, Sistemistica e Comunicazione
Universitá degli Studi di Milano-Bicocca,
Viale Sarca 336, U14
20126 Milano

Bud Mishra
Department of Computer Science, Department of Mathematics
Courant Institute for Mathematical Sciences, NYU
715 Broadway Room 1002
New York, NY 10003

Abstract

Many applications in translational medicine require the understanding of how diseases progress through the accumulation of persistent events. Specialized Bayesian networks called monotonic progression networks offer a statistical framework for modeling this sort of phenomenon. Current machine learning tools to reconstruct Bayesian networks from data are powerful but not suited to progression models. We combine the technological advances in machine learning with a rigorous philosophical theory of causation to produce Polaris, a scalable algorithm for learning progression networks that accounts for causal or biological noise as well as logical relations among genetic events, making the resulting models easy to interpret qualitatively. We tested Polaris on synthetically generated data and showed that it outperforms a widely used machine learning algorithm and approaches the performance of the competing special-purpose, albeit clairvoyant algorithm that is given a priori information about the model parameters. We also prove that under certain rather mild conditions, Polaris is guaranteed to converge for sufficiently large sample sizes. Finally, we applied Polaris to point mutation and copy number variation data in Prostate cancer from The Cancer Genome Atlas (TCGA) and found that there are likely three distinct progressions, one major androgen driven progression, one major non-androgen driven progression, and one novel minor androgen driven progression.

1. Introduction

Modern data science focuses on scientific problems that are replete with high dimensional data, with the data-dimension approaching the sample size. This situation has become often too common in biology, biomedicine and social sciences. Typically, data are collected and summarized in a joint distribution and some useful patterns are extracted from the distribution. Graphical models, in particular Bayesian networks [13, 16, 21], succinctly represent these joint distributions and extract the statistical dependencies between the variables, effectively filtering out indirect relationships to expose the underlying structure of interactions in the system. While this approach is widely applicable, it fails to provide the kind of information needed for many clinical problems, such as survival prediction, therapy design and drug resistance in cancer. These problems would benefit greatly from models that describe a temporal ordering of events describing a progressive process.

Here, the useful information lies in asymmetric relationships, such as causality and precedence, and not necessarily symmetric ones such as correlation. Several research groups have produced statistical progression models of varying complexities but remain disconnected from the technological advances made in the machine learning community, particularly in structure learning and conditional inference in graphical models.

In this paper, we present a novel framework to synthesize recent advances in graphical models with a sound and rigorous theory of causality. Namely, we consider the set of probabilistic logical conditions underlying Suppes's *probabilistic causation* theory [19] to identify *positive prima facie* causes. In other words, we look for a cause C modifying the effect E, positively, by C being temporally prior to E and C raising the probability of E; not all positive prima-facie causes are *genuine*. In this paper, we show how to translate these probabilistic logical conditions into the standard regularized, maximum likelihood score of Bayesian networks, and devise a score based machine learning algorithm, Polaris (Progression mOdel LeARnIng Score), to extract the underlying progression and causal structures although the data are non-temporal and cross sectional.

The second novelty for Polaris is in the way it handles what one may choose to describe as a *causal noise*, which accounts for the net effect of unmodeled (usually, minor and/or rare) causes on an event in the absence of the event's canonical causes. This model thus differs from most statistical models of progression, which focus on observational noise, or the effects of mislabeling the occurrence of an event, in either direction. Last but not least, Polaris tackles a wider range of causal relations, naturally including all that can be described with a probabilistic boolean logic. This capability makes the resulting model easily interpretable with phenomenological statements such as "*the presence of EGFR and MYC mutations causes a mutation in P53 but a mutation in either gene alone does not*."

The rest of the paper is structured as follows. It starts with a technical description of graphical models and progression models, some approaches to structure learning for both types of models, and a perspective on the limitations of existing structure learning algorithms for progression models. It then describes the development of our algorithm, Polaris, grounded on its philosophical roots which lead to its mathematical definitions and ultimately, to its practical implementation. It follows this section with theoretical

convergence results in the case of sufficiently large samples and a demonstration of its practical performance across many realistic data sizes. The next section illustrates how Polaris works, with an application to a practical example in Prostate cancer, while producing some novel hypotheses for its progression. The paper concludes with a discussion of various related issues.

2. Model Descriptions and structure learning

2.1. **Bayesian Networks.** A *Bayesian network* (BN) is a statistical model that provides a sparse and succinct representation of a multivariate probability distribution over n random variables and encodes it into a sparse *directed acyclic graph* (DAG),[1] $G = (V, E)$ over $n = |V|$ nodes, one per variable[2], and $|E| \ll |V|^2$ directed edges. The full joint distribution factors as a product of *conditional probability distributions* (CPDs) of each variable, given its parents in the graph. In a DAG, the set of parents of node X_i consists of all the nodes with edges that point to X_i and is written as $Pa(X_i)$. In this paper, we represent CPDs as tables (see figure 1), in which each row represents a possible assignment of the parents and the corresponding probability of the child, here, a Bernoulli random variable $\in \{0, 1\}$, when it takes the value 1.

$$\mathcal{P}(x_1, \ldots, x_n) \quad = \quad \prod_{X_i \in V} \mathcal{P}(X_i = x_i | Pa(X_i) = x_{Pa(i)}).$$

The set of edges E represents all the conditional independence relations between the variables. Specifically, an edge between two nodes X_i and X_j denotes statistical conditional dependence, no matter on which other variables we condition. Mathematically, this means that for any set of variables $S \subseteq V \setminus \{X_i, X_j\}$, it holds that $\mathcal{P}(X_i, X_j \mid S) \neq \mathcal{P}(X_i \mid S)\mathcal{P}(X_j \mid S)$. In the BN, the symmetrical nature of statistical dependence means that the graphs $X_i \to X_j$ and $X_i \leftarrow X_j$ encode the same conditional independence relations. We call two such graphs I-equivalent[3] and a set of such graphs a Markov equivalence class. In fact, any graphs that contain the same skeletons and v-structures are Markov equivalent. Here, the skeleton refers to the undirected set of edges, in which $X_i \to X_j$ and $X_i \leftarrow X_j$ both map to $X_i \leftrightarrow X_j$, and a v-structure[4] refers to a node with a set of at least two parents, in which no pair of parents share an edge.

[1] A DAG consists of a set of nodes (V) and a set of directed edges (E) between these nodes, such that there are no directed cycles between any two nodes.

[2] In our setting, each node represents a Bernoulli random variable taking values in $\{0, 1\}$. By convention, we refer to variables with upper case letters (e.g. X_i) and the values they take with lower case letter (e.g. x_i).

[3] I stands for independence here.

[4] In BN terminology, parent with no shared edge are considered "unwed parents." For this reason, the v-structure is often called an immorality. In other texts, it is referred to as an unshielded collider.

2.2. **Monotonic Progression Networks.** We define a class of Bayesian networks over Bernoulli random variables called *monotonic progression networks* (MPNs), a term coined in [7]. MPNs formally represent informal and intuitive notions about the progression of persistent events that accumulate monotonically, based on the presence of other persistent events[5]. The conditions for an event to happen are represented in the CPDs of the BN using probabilistic versions of canonical boolean operators, namely conjunction ($\wedge$), inclusive disjunction ($\vee$), and exclusive disjunction ($\oplus$), as well as any combination of propositional logic operators. Figure 1 shows an example of the CPDs associated with various operators.

While this framework allows for any formula to define the conditions of the parent events conducive for the child event to occur, we chose a simpler design to avoid the complexity of the number of possible logical formulas over a set of parents. Namely, we define three types of MPNs, a conjunctive MPN (CMPN), a disjunctive MPN (DMPN[6]), and an exclusive disjunction MPN (XMPN). The operator associated with each network type defines the logical relation among the parents that should hold for the child event to take place. Arbitrarily complex formulas can still be represented as new variables, whose parent set consists of the variables in the formula and whose value is determined by the formula itself. This design choice assumes that most of the relations in a particular application fall under one category, while all others are special cases that can be accounted for individually. Mathematically, the CPDs for each of the MPNs are defined below:

CMPN:

$$\begin{aligned} Pr(X=1|\sum Pa(X) < |Pa(X)|) &\leq \epsilon, \\ Pr(X=1|\sum Pa(X) = |Pa(X)|) &> \epsilon. \end{aligned}$$

DMPN:

$$\begin{aligned} Pr(X=1|\sum Pa(X) = 0) &\leq \epsilon, \\ Pr(X=1|\sum Pa(X) > 0) &> \epsilon. \end{aligned}$$

XMPN:

$$\begin{aligned} Pr(X=1|\sum Pa(X) \neq 1) &\leq \epsilon, \\ \Pr(X=1|\sum Pa(X) = 1) &> \epsilon. \end{aligned}$$

The inequalities above define the *monotonicity constraints* specific to each type of MPN, given a fixed "noise" parameter ϵ. When a particular event occurs despite the monotonicity constraint, we say that the sample is negative with respect to that event. If the event does

[5]In this text, we use the terms variable and event interchangeably.

[6]Sometimes referred to as a semi-monotonic progression network (SMPN).

not occur or occurs in compliance with the monotonicity constraint, then it is a positive sample of that event. Note that in the case in which $\epsilon = 0$, the monotonicity constraints are deterministic and all samples are positive. By convention, we sometimes refer to the rows of a CPD as positive and negative rows and use θ_i^+ to refer to the conditional probability of some positive row i and θ_i^- to refer to the conditional probability of some negative row i.

Finally, we note that probabilistic logical relations encoded in Bayesian networks are not entirely new and have been studied in the artificial intelligence community as *noisy-AND*, *noisy-OR*, and *noisy-XOR* networks [16].

2.3. **Structure learning.** Many algorithms exist to carry out structure learning of general Bayesian networks. They usually fall into two families of algorithms [13], although several hybrid approaches have been recently proposed [4]. The first, *constraint based learning*, explicitly tests for pairwise independence of variables conditioned on the power set of the rest of the variables in the network. The second, *score based learning*, constructs a network to maximize the likelihood of the observed data, with some regularization constraints to avoid over-fitting. Because the data are assumed to be independent and identically distributed (i.i.d.), the likelihood of the data is the product of the likelihood of each datum, which in turn is defined by the factorized joint probability function described in section 2.1. For numerical reasons, log likelihood (LL) is usually used instead of likelihood, and thus the likelihood product becomes the log likelihood sum.

In this paper, we build on the latter approach, specifically relying on the Bayesian Information Criterion (BIC) as the regularized likelihood score. The score is defined below:

$$score_{\text{BIC}}(D, G) = LL(D|G) - \frac{\log M}{2}\dim(G).$$

Here, G denotes the graph (including both the edges and CPDs), D denotes the data, M denotes the number of samples, and $\dim(G)$ denotes the number of parameters in the CPDs of G. The number of parameters in each CPD grows exponentially with the number of parents of that node. For our networks over events, dim(G) for a single node X is $2^{|Pa(X)|}$. Thus, the regularization term $-\dim(G)$ favors nodes with fewer parents or equivalently, graphs with fewer edges. The coefficient $\log M/2$ essentially weighs the regularization term, such that the higher the weight, the more sparsity will be favored over "explaining" the data through maximum likelihood. Note that the likelihood is implicitly weighted by the number of data points, since each point contributes to the score.

With sample size enlarging, both the weight of the regularization term and the "weight" of the likelihood increase. However, the weight of the likelihood increases faster than that of the regularization term[7]. Thus, with more data, likelihood will contribute more to the score. Intuitively, with more data, we trust our observations more and have less need for regularization, although this term needs never completely vanishes.

[7]Mathematically, we say that the likelihood weight increases linearly, while the weight of the regularization term logarithmically.

Statistically speaking, BIC is a consistent score [13]. In terms of structure learning, this property implies that for sufficiently large sample sizes, the network with the maximum BIC score is I-equivalent to the true structure, G^*. From the discussion in 2.1, it is clear that G will have the same skeleton and v-structures as G^*, though nothing is guaranteed regarding the orientation of the rest of the edges. For most graphs, therefore, BIC cannot distinguish among G^* plus all other possible graphs and thus is not sufficient for exact structure learning. In the case of BNs with structured CPDs, such as MPNs, it is possible to improve on the performance of BIC. For example, Farahani et al. modified the BIC score, as described in section 3.2 to drastically improve performance in learning the orientations of all edges.

2.4. **Observational vs Biological Noise.** The notion of probabilistic logical relations among variables to represent disease progression has been developed in two families of models. These two approaches diverge in the treatment of noise, or equivalently, in how the model produces negative, or non-monotonic, samples. The first approach, represented initially by Beerenwinkel et al. [10] and more recently by Ramazzoti et al. [17], encodes a notion of experimental, or observational, noise, in which negative samples result from incorrect labeling of the events. In this model, each generated sample is initially positive in all variables and then may have several event values inverted, with a certain probability. The second approach, represented initially by Farahani et al. [7] and now by the work presented here, encodes biological or causal noise, in which negative samples result from the activation of events by some non-canonical causes, in the absence of canonical ones. In models like these, the level of noise corresponds to the probability that an event occurs despite the absence of its parents.

Observational noise and biological noise have different statistical properties that affect how the model is learned. Namely, observational noise is often assumed to be unbiased and have a Gaussian distribution and thus by the strong law of large numbers, converges to zero for a sufficiently large number of observations. In contrast, biological noise is asymmetric and persists even with large sample sizes. One of the key consequences of these differences is the following: While the asymptotic marginal probabilities of the variables are the same for all levels of noise in the observational noise model, for biological noise, however, the marginal probabilities are very sensitive to the level of noise, irrespective of how large the sample size is. See section 4.5 for details on how this affects learning algorithm presented in this paper.

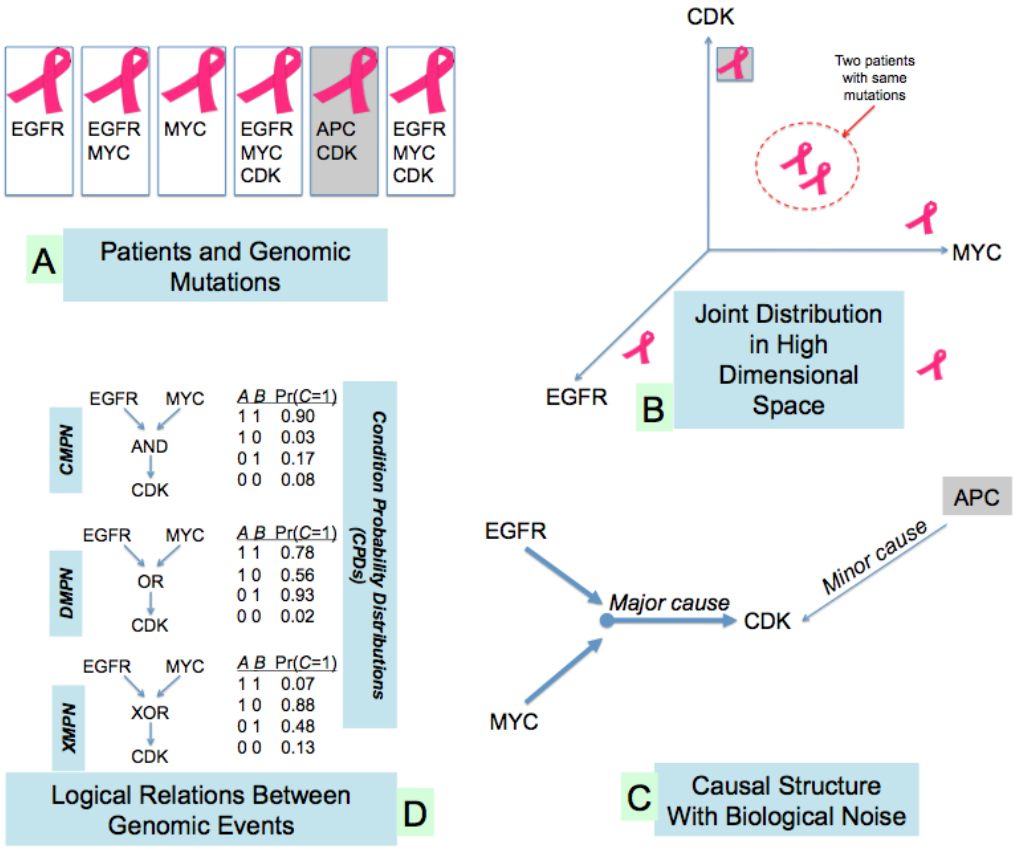

FIGURE 1. The POLARIS algorithm accepts raw cross sectional genomic data and computes a causal progression model with logical relations among the variables. Initially (*top left*), each patient's tumor is sampled during surgery and sequenced afterwards. From the sequencing, we find that each tumor has genomic aberrations in certain genes and not others. Most genes will be common among the tumors, although some may be outliers (highlighted in gray). This data is then projected into a high dimensional space (*top right*) and the genes' co-occurrence frequencies are encoded as a joint distribution over the gene variables. POLARIS mines this data for causal relations (*bottom right*) and encodes the major causal progressions among the genes in a graphical model. The minor causes account for the outliers in the data and often reflect a varying spectrum in cancer types among the patients. These minor causes are averaged and collapsed into a causal or biological noise parameter in the model. Finally, many genomic events, for instance *CDK* mutation, seem to precipitate from the occurrence two or more events, for instance *EGFR* and *MYC* mutations. We provide a language for expressing this dependence (*bottom left*). Using the examples in the figure, we can allow *CDK* to occur only when both *EGFR* and *MYC* occur (CMPN), when either one occurs (DMPN), or when only one but not both occur (XMPN). The examples of conditional probability distributions (CPDs) reflect these logical relations.

3. State of the art review OR Insufficiency of current methods

In our background review, we only consider algorithms that learn progression networks with biological noise. Efficient and effective algorithms to learn models of observational noise have been developed and are described in the literature [14, 17]. Here, we consider global optimization of the BIC score, a representative algorithm for learning general BNs, and DiProg, the only algorithm specifically developed for learning CMPNs and DMPNs.

3.1. **BIC is not sufficient for exact structure learning.** Structure learning for Bayesian networks has improved tremendously in the last decade. In particular, the problem of maximum likelihood learning of Bayesian networks with only discrete and observable variables can usually be solved to optimality using integer programming and LP relaxation. Moreover, regularized ML scores such as BIC have nice mathematical properties that guarantee asymptotic convergence to an I-equivalent structure. However, for most graphs, the optimal BIC score does not belong to one particular structure. In fact, it belongs to the group of structures defined by a Markov equivalence class[8]. Therefore, structure learning through BIC alone cannot distinguish between many structures, only one of which is the structure of the true generating graph. Therefore, BIC is insufficient for exact structure learning. However, for BNs with structured CPDs, such as the progression networks described in 2.2, it is possible to design a score for more accurate structure learning.

3.2. **DiProg algorithm outperforms BIC.** Farahani et al. [7] proposed an algorithm, DiProg, for learning MPNs that outperforms BIC consistently. DiProg learns the structure by optimizing a modified BIC score through reduction to an integer program and LP relaxation. The modification is in the ML parameter estimation of the conditional probabilities. Specifically, if the estimated parameter for $P(X|\sum Pa(X) \neq |Pa(X)|)$ is greater than ϵ, then set it to ϵ. This modification penalizes graph structures that result in non-monotonic conditional probability parameters. Although the authors do not provide a mathematical proof of convergence, it is empirically seen that most of the edges in the original network are learned in the correct orientation, given enough samples.

3.3. **DiProg is not sufficient for real data.** The modification to the BIC score improves performance but relies on a priori knowledge of ϵ, which is rarely available. In fact, the performance of DiProg depends strongly on knowing the correct level of noise (see figure 2). This limitation makes the algorithm unreliable for applications on real data, in which ϵ cannot be known. In this paper, we present an alternative algorithm, Polaris, that learns MPNs and DMPNs without knowledge of ϵ. We also show that Polaris performs significantly better than optimizing BIC and in most cases, better than DiProg with a random ϵ.

4. Developing Our Causal Score

We present a score, namely, the one used in Polaris, that is statistically consistent, like BIC, and correctly orients edges based on the monotonicity of the progression relation, like

[8]Skeleton and immoralities.

DiProg, but without knowing the parameter ϵ a priori. The basic idea behind the score is a heuristic for the likelihood of each sample such that the likelihood reflects both the probability of the sample being generated from its CPD and the probability that the CPD obeys the monotonicity constraints of the true model. Of course, we cannot compute the latter without knowledge of ϵ and thus rely on a nonparametric notion of monotonicity to estimate the underlying CPD. Below, we start with an explanation the development of Polaris and conclude with its philosophical foundations to its asymptotic convergence properties.

4.1. **Foundation in Suppes causality.** We modeled our score after the asymmetrical portion, α, of the causal score, presented earlier in [14]. The authors based this part of the score on Suppes's theory of causality for distinguishing *prima facie* causes from non-causal correlations. Suppes stipulates two conditions for event C to cause event E. First, C must *raise the probability* of E. In the authors' statistical model, this means that $\mathcal{P}(E \mid C) > \mathcal{P}(E \mid \bar{C})$. Second, C must precede E in time. Unfortunately, the authors' model, just like ours, has no notion of time and could not directly infer *temporal priority.* However, under the condition that C is the unique cause of E, it is necessary that C must appear every time E appears but not vice versa. Therefore, the number of occurrences of C must be larger than that of E. From this, it is easy to see that $\mathcal{P}(C) > \mathcal{P}(E)$. In fact, this property of temporal priority also holds for conjunctions over several parents, as E will only appear when all its parents are present.

They define their α score for a causal relation $C \to E$ as $\frac{\mathcal{P}(E|C)-\mathcal{P}(E|\bar{C})}{\mathcal{P}(E|C)+\mathcal{P}(E|\bar{C})}$. They prove that this definition meets both the probability raising and temporal priority conditions explained above. In their paper [14], the authors only consider tree structured graphs, in which every node has at most 1 parent and at most 1 negative row in its CPD. Applied to an MPN, the true α value for each CPD must be strictly positive for each edge – a consequence of the constraint that $\mathcal{P}(E \mid C) > \mathcal{P}(E \mid \bar{C})$ for all MPNs. Thus, when we consider several graphs to fit to observed data, an estimated α with a negative value (below a threshold) means that the corresponding CPD breaks the monotonicity constraint. On the other hand, an estimated α with a positive value (above a threshold) puts more faith in the legitimacy of that CPD. Otherwise, the interpretation of CPD is ambiguous. Justified by these intuitive observations, we claim that α serves as a faithful proxy for monotonicity in tree structured MPNs.

4.2. **Weighted Likelihood Without A Priori Knowledge of Model Parameters.** In this work, we consider more general DAG structured models, in which CPDs can have more than one negative row. To handle this, we assign an α score to each row of the CPD, as defined below. We adopt the notation α_{xi} to denote the α value corresponding to row i of the CPD of variable X. By our convention, θ^{-}_{xi} denotes the probability of negative row i and θ^{+}_{x} the probability of the one[9] positive row of the CPD of X.

[9]This assumption is only true for CMPNs. We extend this notation to DMPNs and XMPNs later.

$$\alpha_{xi} = \begin{cases} 1, & \text{for a positive row;} \\ \frac{\hat{\theta}_x^+ - \hat{\theta}_{xi}^-}{\hat{\theta}_x^+ + \hat{\theta}_{xi}^-}, & \text{for a negative row.} \end{cases}$$

Thus, as argued earlier, α is now a heuristic for the monotonicity of each row of a CPD rather than the CPD as a whole. It follows that each negative sample has a corresponding α between -1 and 1. Thus, we weigh each negative sample by its α value to reflect our belief that its CPD row conforms to the monotonicity constraints. This strategy leads to CPDs with high monotonicity to be favored through their samples, whereas CPDs with poor monotonicity are penalized through their samples. Moreover, by handicapping the samples instead of the CPDs directly, we allow rows whose conditional probabilities were estimated with more samples to have a larger effect on the score. The resulting α-weighted likelihood score ($score_{\alpha\text{WL}}$) for variable X given sample d is defined below, where $\hat{\theta}_x^+$ and $\hat{\theta}_{xi}^-$ are empirical estimates of their respective parameters. Note that because of the indicator function in the exponent of the α term in the score, only the α term of the row that corresponds to the sample is used to weigh the likelihood. Specifically, if the sample is positive, the likelihood is not altered, whereas if the sample is negative, the likelihood is penalized in proportion to the α score for that sample's corresponding row.

$$score_{\alpha\text{WL}}(X:d) = Pr\left(X = d_x | Pa(X) = d_{Pa(X)}\right) \cdot \prod_{i \in |CPD_x|} \alpha_{xi}^{\mathbb{1}(d_{Pa(X)} = CPD_x(i))}.$$

Of course, the score we use for structure learning includes the BIC regularization term, so the full combined score for a single variable X given a datum d is below. The last line defines the composed score for the all the variables, V, over all the data, D.

$$\begin{aligned} & score_{\alpha\text{WL,BIC}}(X:d) \\ &= \log\left[Pr(X = d_x | Pa(X) = d_{Pa(X)}) \cdot \prod_{i=1}^{|CPD_x|} \alpha_{xi}^{\mathbb{1}(d_{Pa(X)} = CPD_x(i))}\right] \\ &\quad - \frac{\log_M}{2} dim(X|Pa(X)), \\ & score_{\alpha\text{WL,BIC}}(X:d) \\ &= \log\left[Pr(X = d_x | Pa(X) = d_{Pa(X)}) + \sum_{i=1}^{|CPD_x|} \mathbb{1}(d_{Pa(X)} = CPD_x(i)) \log \alpha_{xi}\right] \\ &\quad - \frac{\log_M}{2} dim(X|Pa(X)), \\ & score_{\alpha\text{WL,BIC}}(X:d) \end{aligned}$$

$$= \quad LL(d_x, d_{Pa(X)}|G) + \alpha(X|d) - \frac{\log M}{2} dim(X|Pa(X)), \quad \text{and, finally}$$

$$score_{\alpha\text{WL,BIC}}(G:D)$$

$$= \quad LL(D|G) + \sum_{d \in D} \sum_{X \in V} \alpha(X|d) - \frac{\log M}{2} dim(G).$$

For brevity, we use the shorthand

$$\alpha(X|d) = \sum_{i \in |CPD_x|} \mathbb{1}(d_{Pa(x)} = CPD_x(i)) \log \alpha_{xi}.$$

In other words, it is the α of the row of the CPD of X that corresponds to $d_{Pa(X)}$.

4.3. **Multiplicative factor improves performance and makes certain asymptotic guarantees.** Asymptotically, the BIC is known to reconstruct the correct skeleton and orient edges in immoralities correctly. Since we desire our score to enhance this result further and orient the remaining edges correctly without disturbing the correct skeletal structure, we introduce a new weight to the whole monotonicity term of the score. This weight is structured to approach zero in the limit, as the sample size approaches infinity. Thus, for small sample sizes, the monotonicity component will play a larger role in the overall score. Then, as the BIC component converges to a more stable structure, the monotonicity component chooses the exact structure among several equally likely ones. For these asymptotic results, we chose the simplest weight that is inversely proportional to the sample size: $1/M$. The final score we developed for structure learning of MPNs is below.

$$score_{Polaris}(G:D)$$

$$= \quad LL(D|G) + \frac{1}{M} \sum_{d \in D} \sum_{X \in V} \alpha(X|d) - \frac{\log M}{2} dim(G).$$

We prove mathematically that this score asymptotically learns the correct exact structure of an MPN under certain conditions – especially, conditions enforcing the absence of transitive edges and a sufficiently low ϵ parameter. In practice, however, we found that our algorithm converges on the correct structure for graphs with transitive edges and non-negligible ϵ values (see figure 2).

Definition 1 (Faithful temporal priority)**.** *In a monotonic progression network G, if there exists a path from X_j to X_i, then the temporal priority between X_i and X_j is* faithful *if $\mathcal{P}(X_j) > \mathcal{P}(X_i)$.*

Theorem 1 (Convergence conditions for Polaris)**.** *For a sufficiently large sample size, M, under the assumptions of no transitive edges and faithful temporal priority relations*

(see Definition 1) between nodes and their parents at least for nodes that have exactly 1 parent, optimizing POLARIS *converges to the exact structure.* □

See supp. mat. for a complete proof.

4.4. **Extension to DMPNs and XMPNs.** The score stated in the previous section works for all three classes of MPNs, with minor modifications to the definition of α, depending on the monotonicity constraints. The main difference between CMPNs and the other two types lies in the fact that each CPD corresponding to a CMPN has exactly one positive row. In contrast, the CPDs in DMPNs have exactly one negative row and the CPDs in XMPNs may have multiple positive and negative rows (see figure 1). Specifically, the only negative row for DMPNs is the case in which all parent nodes equal zero. For XMPNs, any row with exactly one parent event equal to one is a positive row and all the rest are negative rows. In order to extend the definition of α to DMPNs and XMPNs, we treat all events that correspond to the positive rows of a CPD as one event. The probability of this large event is called θ^+, just as in the CMPN case, and it is defined below for both DMPNs and XMPNS.

$$\begin{aligned}\theta^+_{DMPN}(X) &= \mathcal{P}(\sum Pa(X) > 0),\\ \theta^+_{XMPN}(X) &= \mathcal{P}(\sum Pa(X) = 1).\end{aligned}$$

With these alternative constructions of θ, α is well defined for all three types of MPNs.

4.5. **Temporal Priority in the Presence of Biological Noise.** The α score for learning models in [14] and [17] enforces both probability raising and, for conjunctive or singleton parent sets, temporal priority. The model of noise considered there has the property that, for sufficient large sample sizes, by the large of large numbers, the probability of a negative sample approaches zero. However, in our model of noise, θ^-_i's are fixed parameters and do not approach zero. Thus, temporal priority cannot always be correctly imputed for all causal relations. That is, $C \to E$ does not necessary mean that $\mathcal{P}(C) > \mathcal{P}(E)$. Instead, temporal priority is decided by ϵ, θ^+ and the marginal probabilities, as specified in the equation below. Specifically, high ϵ and correspondingly high θ^-, low θ^+ and close marginal probabilities all make it easier to reverse the observed temporal priority.

$$\mathcal{P}(X) = \mathcal{P}(Pa(X) = 1) \cdot \theta^+ + \sum_i (1 - \mathcal{P}(Pa(X) = CPD_X(i))) \cdot \theta^-_i.$$

Note that in this context, θ^+ is uniquely defined, as we assume either a conjunctive or singleton parent set, and the sum is only over the negative rows of the CPD. Asymptotically, this score works just as well for DMPNs and XMPNs as it does for CMPNs for graphs without transitive connections. This is because, in the proof of Theorem 1, temporal priority must only hold for nodes with exactly one parent, and in that case, the three monotonicity constraints are indistinguishable.

5. MPN structure learning with Polaris

In this section, we describe and analyze the algorithm that uses the Polaris score to learn MPN structure.

5.1. α **Filtering.** Before optimizing the score, there are certain parent sets that one may wish to eliminate as hypotheses. This pre-optimization filtering is done for two reasons. First, it prevents the optimization algorithm from selecting a spurious parent set. Second, it speeds up computation significantly by not computing the full score for that hypothetical parent set. We use the α score to filter hypotheses, rejecting those solutions that create a negative α for at least one row of the CPD. This α-filter is used for all types of MPNs and greatly improves efficiency without eliminating too many true hypotheses. In fact, we proved mathematically that asymptotically, the α filter will be free of any mistakes.

Lemma 1 (Convergence of α-filter)**.** *For a sufficiently large sample size, M, the α-filter produces no false negatives for CMPNs, DMPNs, and XMPNs.* □

See supp. mat. for a proof.

5.2. **Optimizing the score with GOBNILP.** After pruning the hypothesis space with the α filter, we use GOBNILP [6, 1, 12], a free, publicly available BN structure learning package, to find the the network with the highest Polaris score. Given an upper bound on the maximum number of parents (by default, 3), GOBNILP expects as input the scores for each node given each possible combination of parents. For each node, our code produces this information with a depth first search through the power set of the rest of the nodes in the graph. Any hypothetical parent set that is filtered is simply not included as a possible solution for that node in the input to GOBNILP.

6. Results

6.1. **Performance on Synthetic Data.** We conducted several experiments to test the performance of Polaris on data generated from synthetic networks, all on ten variables. The network topologies were generated randomly, and the CPDs were generated according to the monotonic constraints imposed by the type of MPN and the value of ϵ. These networks were sampled with different sample sizes. In all experiments, the performance metrics were measured over fifty synthetic topologies sampled ten times, for each value of ϵ and sample size.

We compared the performance of Polaris against two standards, the optimization of the BIC score and the clairvoyant[10] DiProg algorithm, across a variety of biologically and clinically realistic ϵ values and sample sizes. To evaluate the performance of each algorithm, we measured both the recall, the fraction of true edges recovered, and the precision, the fraction of recovered edges that are true. We placed detailed figures for recall and precision at realistic sample sizes as well as asymptotic sample sizes for CMPNs, DMPNs, and XMPNs in the supplementary material. In figure 2, we summarize these results concisely

[10]By clairvoyant, we mean that the algorithm has a priori knowledge of ϵ.

for all three types of MPNs by using AUPR, or the area under the precision-recall curve, as our performance metric. We expected Polaris to perform significantly better than BIC, which is nonspecific for monotonic relations and slightly worse than the clairvoyant DiProg algorithms, as Polaris does not have access to the correct value of ϵ. The results showed this exact trend for recall, precision, and AUPR. The gap between the clairvoyant DiProg and Polaris remained consistent across all parameter values and relatively low, as opposed to the gap between Polaris and BIC optimization

Finally, we considered the performance of Polaris against a non-clairvoyant DiProg by passing DiProg one of fifty randomly sampled values of ϵ. Because of the cost of running DiProg fifty times, we limited our model type to CMPN, ϵ to 0.15, and sample size to 200. The box plot in figure 2 shows the variance of performance for Polaris, the average performance of the non-clairvoyant DiProg, the performance of the non-clairvoyant DiProg with the most incorrect value of ϵ, and finally, the performance of the clairvoyant DiProg. Again using AUPR as the performance metric, we found that the average performance of the non-clairvoyant DiProg had a significantly lower mean and considerably larger variance than those of Polaris. Moreover, the mean of the worst case performance of DiProg was almost twice as low as that of Polaris, and the variance was slightly larger. From these analyses, we conclude that when ϵ is not known, we expect more accurate and more consistent results from Polaris than from DiProg.

In the supplementary material, we also demonstrate the efficacy and accuracy of the α-filter for CMPNs, DMPNs, and XMPNs. On average, the filter eliminates approximately half of all possible hypotheses and makes considerably less than one mistake per network. In fact, for sufficiently large sample sizes, the false negative rate drops to almost zero.

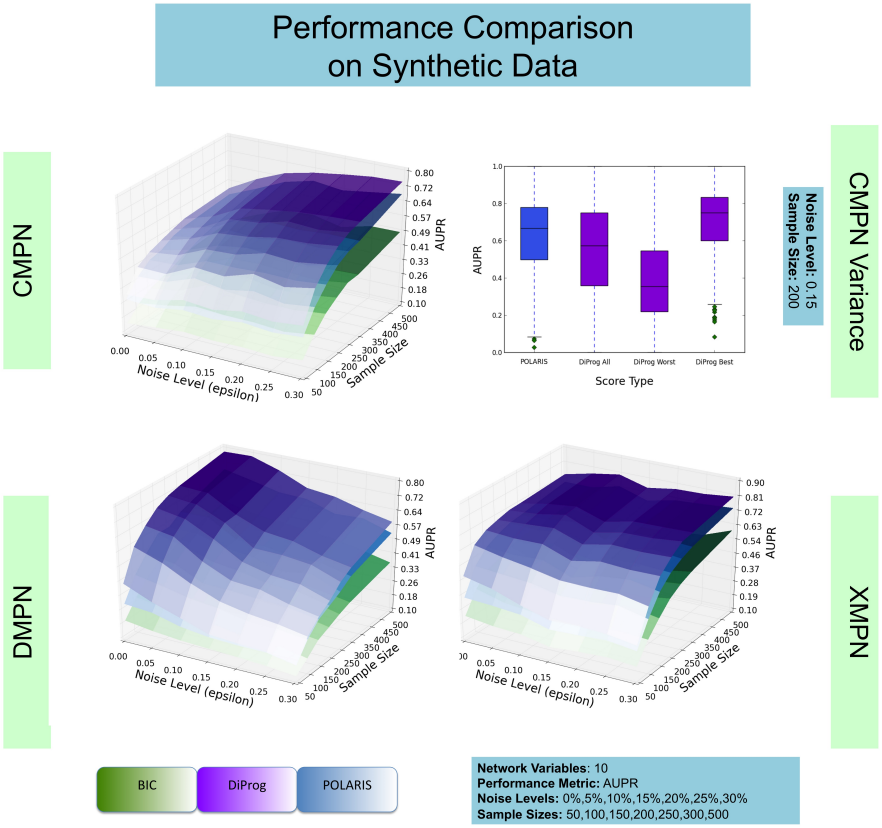

FIGURE 2. We tested the performance of POLARIS against the optimization of a standard symmetric score, BIC, and a clairvoyant algorithm for learning MPNs, DiProg. We tested each algorithm across several different levels of noise (0% to 30%) and across several realistic number of training samples (50 to 500). In each case, the network contained ten variables, common for progression models, although each algorithm can handle a great deal more. The three surface plots show the performance of each algorithm for different MPN types, CMPN on the *top left*, DMPN on the *bottom left* and XMPN on the *bottom right*. The box plots on the top right demonstrate the dependence of DiProg performance on *a priori* knowledge of ϵ. We learned a network with ten variables, 15% noise and 200 samples with POLARIS, DiProg with the correct ϵ, and DiProg with a random ϵ. The second column shows the average performance across the random ϵ values, the third column shows the worst performance with a random ϵ value, and finally, the fourth column shows the performance with knowledge of the correct ϵ value. For all four plots, we measured the rate of both true positives (recall) and true negatives (precision) by computing the area under the precision-recall curve, or the AUPR.

6.2. **Biological Example.** We demonstrate the use of Polaris on prostate cancer (PCA) data. We conducted a literature search to find the genomic events most prominent in PCA and some theories about the ordering of these events. We limited our search to copy number variations (CNVs), mutations and fusion events, as these are believed to be persistent. From the experimental observations of the papers we found [9, 11, 22, 3, 20, 2, 18], we posit a progression model with 3 distinct sub-progressions. To test this theory, we learned a CMPN based on the copy number alteration (CNA), mutation, and fusion event data on the genes discussed above. We used the TCGA [15] prostate adenocarcinoma dataset of 246 sequenced tumors, available through MSKCC's cBioPortal [5, 8] interface.

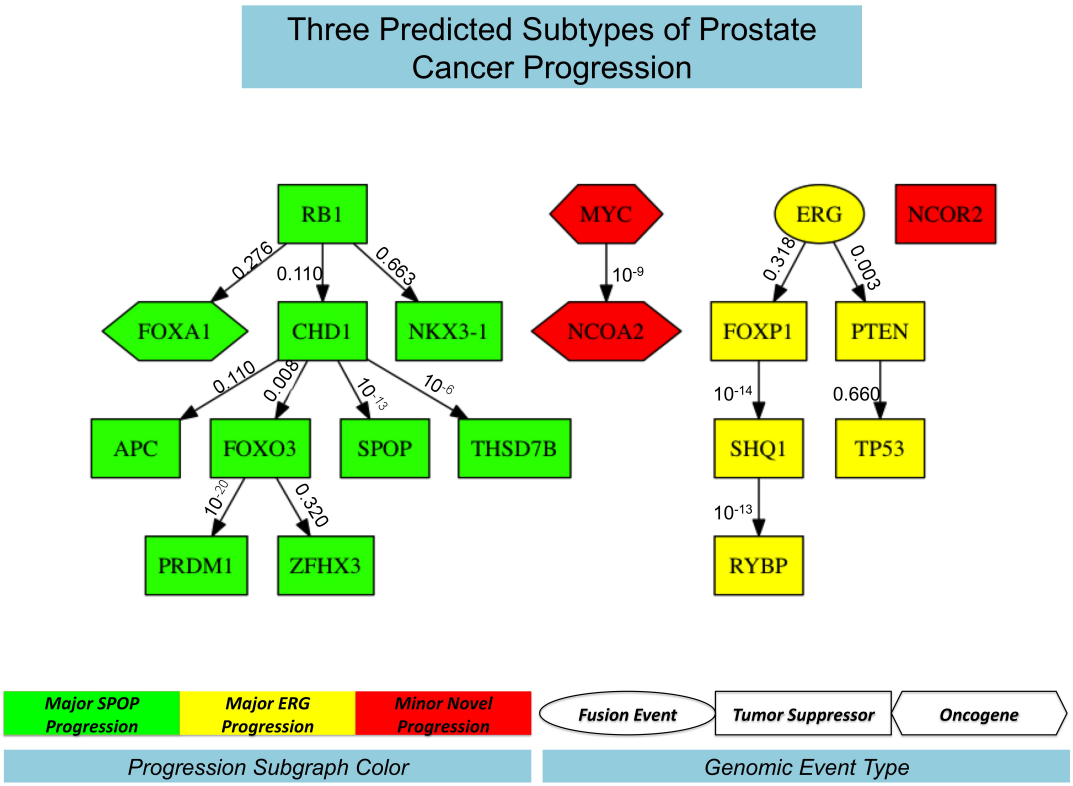

Figure 3. Polaris was used to learn a CMPN model for prostate cancer. We selected the most commonly implicated oncogenes, tumor suppressor genes, and gene fusion events from the literature and used copy number variation and point mutation data from the TCGA database. Each edge is labeled with the fold change in the network score when the edge is left out. Based on the topology and our literature survey, we define three distinct progressions within the graph and each is labeled red, green or yellow.

We found that our learned model, shown in figure 3, validates and unifies the observations of the papers above in one tri-progression model. First, we found two major progressions, one centered around *TMPRSS2-ERG* fusion (below referred to as just "*ERG*") and another around *CHD1* and *SPOP*. This confirms Barbieri et al.'s [3] theory of two distinct progressions defined by *SPOP* and *ERG*. Moreover, our model captures the associated genes Barbieri et al. predicted in each progression. Namely, *CHD1*, *FOXO3*, and *PRDM1* are involved in the *SPOP* progression and *PTEN* and *TP53* in the *ERG* progression. Next, we postulate that *MYC*, *NCOA2* and *NCOR2* are all involved in a third progression, even though *NCOR2* appears isolated from the other two in the graph. We justify this decision by noting the observations of Grasso et al. [11], Taylor [20], Weischenfeldt et al. [22] and Gao et al. [9]. Grasso et al. predict that there is a third progression that includes neither *CHD1* nor *ERG*. Taylor et al. predict that there is a subtype with poor prognosis that involves the amplification of *MYC* and *NCOA2*. Weischenfeldt et al. predict that early onset PCA involves the Androgen receptor (*AR*) pathway and *NCOR2* mutation but does not include *ERG*, *CHD1*, or *PTEN*. Gao et al. show an experimental connection between MYC and AR expression, strengthening the *MYC/NCOA2* involvement in the third pathway. Lastly, the figure shows several key driver genes (*NKX3-1, APC, ZFH3, THSD7B, FOXP1, SHQL, RB, RYBP*) in the progression of PCA that have not been assigned to either the *SPOP* or *ERG* progressions. The model proposes an assignment of these genes to their respective progressions that can be experimentally tested. As a sanity check, we note that *FOXP1*, *SHQ1*, and *RYBP*, all genes in the 3p14 region, are closely related in the progression.

7. Discussion and Future Work

Polaris is a machine learning (ML) algorithm for discovering causal structure from data, founded on score-based graphical models, in which the score builds on classical probabilistic theory of causality developed by Suppes. Graphical models, in particular Bayesian networks, are by now extensively studied and well understood. There is an active community of researchers dedicated to developing powerful tools for efficient structure learning, parameter estimation, and conditional inference. Many such tools are publicly available as open source platforms and are continually evolving with data science applications: both in businesses and sciences. Polaris derives its power and flexibility from this eco-system of tools. Despite the abundance of these ML tools so far, practically all existing learning algorithms for graphical models have been ill-suited to the task of monotonic progression reconstruction. Polaris is able to uniquely tailor these algorithms to suit this particular task. Although, we are not the first ones to attempt to solve this problem (see Fahrani [7]), we are the first to devise a fully score based, *non-clairvoyant* algorithm (i.e., no prior knowledge of the parameters of the causal noise). In particular, we address causal or biological noise in a realistic manner, thus paving the way to real practical applications.

Polaris accomplishes its intended tasks effectively and efficiently. To quantify its efficacy, we provide a theoretical analysis in the supplementary materials, containing a proof of its asymptotic convergence under some mild conditions. Moreover, we empirically tested

the algorithm extensively on a variety of noise levels and sample sizes. We found that it outperforms the standard score for structure learning and closely trails behind the clairvoyant one. We do not believe, however, that Polaris, by virtue of its machine learning abilities, can solely and completely solve all the underlying problems in cancer systems biology. It has several shortcomings: it is not yet the definitive algorithm to cover all notions of causality, some of which could be important to decipher a progressive disease like cancer; it depends on astute experimentalists and incisive experiments to provide those observations that underpin how the disease progresses; and finally, it relies on molecular biologists to interpret its output before it can be related to the mechanistic models of genes, expressions and signaling. What we have achieved instead are the following: an operationalized version of a rigorously developed theory of causality, now well integrated with the machine learning technology, and a useful tool for biologists interested in the progression of genomic events in cancer.

In our future work, we will explore several of these shortcomings of the Polaris framework: we will develop more robust statistical estimators, and infer synthetically lethal interactions from data. With the latter, we will prioritize development of therapy-design tools based on progression models to guide cancer drug selection as well as discovery of concrete hypotheses for novel drug targets.

References

[1] Tobias Achterberg. Scip: solving constraint integer programs. *Mathematical Programming Computation*, 1(1):1–41, 2009.

[2] Sylvan C Baca, Davide Prandi, Michael S Lawrence, Juan Miguel Mosquera, Alessandro Romanel, Yotam Drier, Kyung Park, Naoki Kitabayashi, Theresa Y MacDonald, Mahmoud Ghandi, et al. Punctuated evolution of prostate cancer genomes. *Cell*, 153(3):666–677, 2013.

[3] Christopher E Barbieri, Sylvan C Baca, Michael S Lawrence, Francesca Demichelis, Mirjam Blattner, Jean-Philippe Theurillat, Thomas A White, Petar Stojanov, Eliezer Van Allen, Nicolas Stransky, et al. Exome sequencing identifies recurrent spop, foxa1 and med12 mutations in prostate cancer. *Nature genetics*, 44(6):685–689, 2012.

[4] Eliot Brenner and David Sontag. Sparsityboost: A new scoring function for learning bayesian network structure. *arXiv preprint arXiv:1309.6820*, 2013.

[5] Ethan Cerami, Jianjiong Gao, Ugur Dogrusoz, Benjamin E Gross, Selcuk Onur Sumer, Bülent Arman Aksoy, Anders Jacobsen, Caitlin J Byrne, Michael L Heuer, Erik Larsson, et al. The cbio cancer genomics portal: an open platform for exploring multidimensional cancer genomics data. *Cancer discovery*, 2(5):401–404, 2012.

[6] James Cussens and Mark Bartlett. Advances in bayesian network learning using integer programming. *arXiv preprint arXiv:1309.6825*, 2013.

[7] Hossein Shahrabi Farahani and Jens Lagergren. Learning oncogenetic networks by reducing to mixed integer linear programming. *PloS one*, 8(6):e65773, 2013.

[8] Jianjiong Gao, Bulent Arman Aksoy, Ugur Dogrusoz, Gideon Dresdner, Benjamin Gross, S Onur Sumer, Yichao Sun, Anders Jacobsen, Rileen Sinha, Erik Larsson, et al. Integrative analysis of complex cancer genomics and clinical profiles using the cbioportal. *Science signaling*, 6(269):pl1, 2013.

[9] Lina Gao, Jacob Schwartzman, Angela Gibbs, Robert Lisac, Richard Kleinschmidt, Beth Wilmot, Daniel Bottomly, Ilsa Coleman, Peter Nelson, Shannon McWeeney, et al. Androgen receptor promotes ligand-independent prostate cancer progression through c-myc upregulation. *PloS one*, 8(5):e63563, 2013.

[10] Moritz Gerstung, Michael Baudis, Holger Moch, and Niko Beerenwinkel. Quantifying cancer progression with conjunctive bayesian networks. *Bioinformatics*, 25(21):2809–2815, 2009.

[11] Catherine S Grasso, Yi-Mi Wu, Dan R Robinson, Xuhong Cao, Saravana M Dhanasekaran, Amjad P Khan, Michael J Quist, Xiaojun Jing, Robert J Lonigro, J Chad Brenner, et al. The mutational landscape of lethal castration-resistant prostate cancer. *Nature*, 487(7406):239–243, 2012.

[12] Tommi Jaakkola, David Sontag, Amir Globerson, and Marina Meila. Learning bayesian network structure using lp relaxations. In *International Conference on Artificial Intelligence and Statistics*, pages 358–365, 2010.

[13] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[14] Olde Loohuis Loes, Caravagna Giulio, Graudenzi Alex, Ramazzotti Daniele, Mauri Giancarlo, Antoniotti Marco, and Mishra Bud. Inferring causal models of cancer progression with a shrinkage estimator and probability raising. *arXiv preprint arXiv:1311.6293*, 2013. submitted for publication.

[15] Roger McLendon, Allan Friedman, Darrell Bigner, Erwin G Van Meir, Daniel J Brat, Gena M Mastrogianakis, Jeffrey J Olson, Tom Mikkelsen, Norman Lehman, Ken Aldape, et al. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, 455(7216):1061–1068, 2008.

[16] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[17] Daniele Ramazzotti, Giulio Caravagna, Loes Olde Loohuis, Alex Graudenzi, Ilya Korsunsky, Giancarlo Mauri, Marco Antoniotti, and Bud Mishra. Efficient inference of cancer progression models. *bioRxiv*, 2014.

[18] Mark A Rubin, Christopher A Maher, and Arul M Chinnaiyan. Common gene rearrangements in prostate cancer. *Journal of Clinical Oncology*, 29(27):3659–3668, 2011.

[19] Patrick Suppes. A probabilistic theory of causation, 1970.

[20] Barry S Taylor, Nikolaus Schultz, Haley Hieronymus, Anuradha Gopalan, Yonghong Xiao, Brett S Carver, Vivek K Arora, Poorvi Kaushik, Ethan Cerami, Boris Reva, et al. Integrative genomic profiling of human prostate cancer. *Cancer cell*, 18(1):11–22, 2010.

[21] Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

[22] Joachim Weischenfeldt, Ronald Simon, Lars Feuerbach, Karin Schlangen, Dieter Weichenhan, Sarah Minner, Daniela Wuttig, Hans-Jörg Warnatz, Henning Stehr, Tobias Rausch, et al. Integrative genomic analyses reveal an androgen-driven somatic alteration landscape in early-onset prostate cancer. *Cancer Cell*, 23(2):159–170, 2013.