Learning Ancestral Polytrees

Guangdi Li[1] LIGUANGDI.RESEARCH@GMAIL.COM
Anne-Mieke Vandamme[1,2] ANNEMIE.VANDAMME@UZLEUVEN.BE
Jan Ramon[3] JAN.RAMON@CS.KULEUVEN.BE
[1] Rega Institute for Medical Research, Department of Microbiology and Immunology, KU Leuven, Leuven, Belgium
[2] Centro de Malária e Outras Doenças Tropicais and Unidade de Microbiologia, Instituto de Higiene e Medicina Tropical, Universidade Nova de Lisboa, Lisboa, Portugal
[3] Department of Computer Science, KU Leuven, Leuven, Belgium

Abstract

Causal polytrees are singly connected causal models and they are frequently applied in practice. However, in various applications, many variables remain unobserved and causal polytrees cannot be applied without explicitly including unobserved variables. Our study thus proposes the ancestral polytree model, a novel combination of ancestral graphs and singly connected graphs. Ancestral graphs can model causal and non-causal dependencies, while singly connected models allow for efficient learning and inference. We discuss the basic properties of ancestral polytrees and propose an efficient structure learning algorithm. Experiments on synthetic datasets and biological datasets show that our algorithm is efficient and the applications of ancestral polytrees are promising.

1. Introduction

Causal graphical models have been proposed to explicitly convey causal relations between causes and their effects in reasoning tasks (Pearl, 2000). As a special class, polytrees are singly connected graphical models where each pair of variables is connected through at most one path (Huete & de Campos, 1993). Since Rebane and Pearl introduced polytree-like Bayesian networks (Rebane & Pearl, 1987), called dependency polytrees, further research has shown that, (1) belief propagation can be performed computationally efficiently in polytrees (Pearl, 1988). (2) Learning the maximum likelihood dependency polytrees was proven to be NP-hard (Dasgupta, 1999). (3) Polynomial algorithms were proposed to learn causal polytrees via conditional independence (CI) tests (Huete & de Campos, 1993; Ouerd et al., 2004). (4) Based on independency properties of isomorphic polytrees (Campos, 1998), a sound and complete criterion was proposed to read independence relations from minimal directed independence maps (Pena, 2007).

Insofar, many variables remain unobserved in many applications, which have driven us to design robust causal models bearing unobserved (synonymous with latent and hidden) variables. However, learning large Bayesian networks is slow (Ouerd et al., 2004) and causal polytrees cannot express causal flows without explicitly including unobserved variables, causing the increased complexity of causal reasoning, structure learning and inference. The drawbacks above motivate us to introduce ancestral polytrees (APs) with a fast structure learning algorithm and we show this new model can be used to learn many biological systems.

Polytree models have been applied in real world applications. For example, dependency polytrees were efficiently implemented to enhance caching strategies in distributed databases (Messaouda et al., 2003). Based on dependency polytrees, an inference framework was successfully designed to optimize hardware components according to the performance and price of both traditional and nanotechnology architectures (Zaveri & Hammerstrom, 2010). Moreover, protein signaling pathways might be modeled by causal polytrees. For instance, Figure 1 illustrates an NF-kB protein signaling pathway, which activates mammalian immune system cells to produce antibodies against inflammation (Lodish et al., 2004). In this example, causal flows are indicated by red arrows and the activation or inhibition of involved proteins represent cause or effect events.

Protein signaling pathways can be aptly modeled in cases where protein signalling data have been collected, for instance, using multiparameter flow cytometry (Sachs et al., 2005). So far, many proteins in protein signaling pathways remain unobserved, which have driven us to design robust

The Workshop of Learning Tractable Probabilistic Model (LTPM) at the 31st International Conference on Machine Learning (ICML), Beijing, China, 2014. Copyright 2014 by the authors.

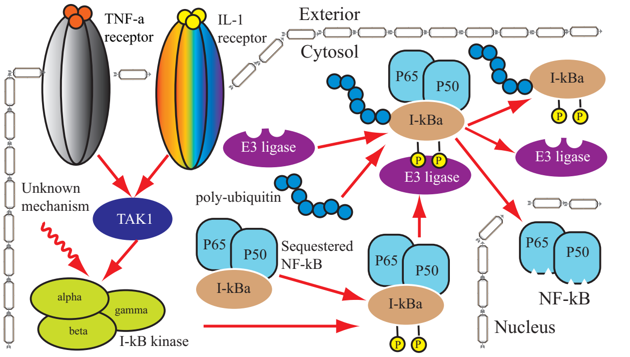


Figure 1. An NF-kB protein signaling pathway (adapted from (Lodish et al., 2004)). IL-1 and TNF-α are transmitted to IL-1 receptors and TNF-α receptors due to extracellular signals. These receptors then activate TAK1, which immediately activates the I-kBα kinase. The I-kBα kinase phosphorylates two serine residues in the I-kBα, allowing for the further binding of the E3 ligase to trigger the degradation of I-kBα and NF-kB. Thereafter, NF-kB is transported into nucleus and activates gene transcription. Twisted red arrow indicates that there is still an unknown mechanism which cooperates with TAK1 to activate the I-kBα kinase.

causal models bearing latent variables. However, causal polytrees cannot express causal flows without explicitly invoking latent variables, causing the increased complexity of causal reasonings. Fortunately, ancestral graphs (AGs) have been proposed to model latent variables without invoking any additional variables (Ali et al., 2009). This has motivated us to introduce ancestral polytrees (APs) as the extensions of causal polytrees.

In ancestral graphs, every missing edge indicates an independence relation (Richardson & Spirtes, 2002). Besides the lack of any directed cycles in both DAGs and AGs, AGs contain directed edges, bidirected edges and undirected edges, in contrast to DAGs, which only permit directed edges. AGs are refined surrogates for DAGs even in the presence of unobserved variables and selection effects (Zhang, 2008). Since causal polytrees are the offsprings of DAGs and singly connected networks, APs are the progenies of AGs and singly connected networks. This merit allows for APs to inherit the merits of both AGs and singly connected networks. On the one hand, APs can express causal diagrams without invoking additional variables in the presence of unobserved variables. On the other hand, due to their simplified structures, APs might guarantee a fast structure learning and inference compared to DAGs and general AGs. However, this inheritance in turn demands a strong assumption — that is, the underlying reasoning diagrams of APs must be singly connected.

This paper begins with basic definitions. We then characterize the properties of APs regarding Markov equivalence, essential graphs and factorization. We thereafter introduce a structure learning algorithm. In the experiments, we compare the performance of our algorithm with other state-of-the-art methods on synthetic datasets. We also apply our model to investigate the protein signalling pathways and HIV-1 mutation pathways using three biological datasets.

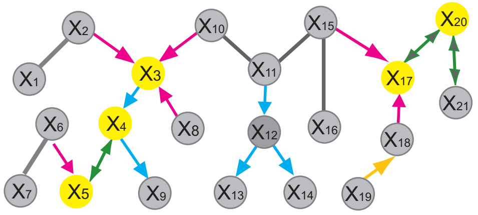


Figure 2. A polytree network with 21 vertices.

2. Definitions and properties

Most notations in this section have been adapted from (Ali et al., 2009). For a graph $G = (V, E)$ we denote with $V(G)$ the set of vertices of G and with $E(G) \subseteq V \times V$ the set of edges of G, where $E \subseteq \{\alpha \; op \; y \mid \alpha, \beta \in V \wedge op \in \{\leftarrow, \rightarrow, \leftrightarrow, -\}\}$. The symbols $\alpha \leftarrow \beta$, $\alpha - \beta$ and $\alpha \leftrightarrow \beta$ denote the directed, undirected and bidirected edges between vertices α and β, respectively. $G^U = (V, E^U)$ represents the undirected version of G called *skeleton*. The endpoint $>$ of an edge is called an *arrowhead*, or the endpoint $-$ is a *tail*. The symbol $*$ is used if the endpoint of an edge is either an arrowhead or a tail. For instance, $\alpha - *\beta$ means either $\alpha \rightarrow \beta$ or $\alpha - \beta$, and $\alpha * \rightarrow \beta$ means either $\alpha \rightarrow \beta$ or $\alpha \leftrightarrow \beta$. The *parent* set of a vertex α is $Pa(\alpha) \equiv \{\beta \mid \beta \rightarrow \alpha\}$; the *neighbor* set is $Ne(\alpha) \equiv \{\beta \mid \beta - \alpha\}$; the *spouse* set is $Sp(\alpha) \equiv \{\beta \mid \beta \leftrightarrow \alpha\}$; the *descendant* set is $De(\alpha) \equiv \{\beta \mid \alpha \rightarrow \cdots \rightarrow \beta$ or $\beta = \alpha\}$; the *ancestor* set is $An(\alpha) \equiv \{\beta \mid \beta \rightarrow \cdots \rightarrow \alpha$ or $\beta = \alpha\}$; the *anterior* set is $Ant(\alpha) \equiv \{\beta \mid \beta - * \cdots - *\alpha$ or $\beta = \alpha\}$; the *archaic* set is $Ar(\alpha) \equiv \{\beta \mid \beta * \rightarrow \cdots * \rightarrow \alpha$ or $\beta = \alpha\}$. A *path* $\pi_{\alpha,\beta}$ refers to a sequence of edges from α to β without duplicate edges. The $\pi_{\alpha,\beta}$ is a *directed*, *undirected* or *bidirected* path if it only contains directed, undirected, or bidirected edges, respectively. A path $\alpha * - * \beta * - * \gamma$ is called a *triple* if α and γ are disjointed. A vertex β is called a *collider* on a path if and only if the path contains a triple $\alpha * \rightarrow \beta \leftarrow *\gamma$, so called *ν-structure*. The $\pi_{\alpha,\beta}$ is an *inducing path* if its internal vertices are all colliders and ancestors of either α or β, or both (see examples and properties of inducing paths on the page 2815 of (Ali et al., 2009)). An undirected subgraph comprises vertices linked by undirected paths alone, whereas a bidirected subgraph comprises vertices linked by bidirected paths alone.

Given the polytree graph in Figure 2, we provide several examples of above definitions: $Pa(X_5) = \{X_6\}$; $Ne(X_2) = \{X_1\}$; $Sp(X_4) = \{X_5\}$; $De(X_3) = \{X_3, X_4, X_9\}$; $An(X_4) = \{X_2, X_3, X_4, X_8, X_{10}\}$; $Ant(X_4) = \{X_1, X_2, X_3, X_4, X_8, X_{10}, X_{11}, X_{15}\}$;

$Ar(X_4) = \{X_2, X_3, X_4, X_5, X_6, X_8, X_{10}\}$; $X_2 \to X_3 \leftarrow X_{10}$ and $X_3 \to X_4 \leftrightarrow X_5$ are ν-structures; X_3, X_4, X_{17}, X_{20} are colliders; $\pi_{X_{10},X_{16}}$ is an undirected path; π_{X_{10},X_9} is a directed path; $\pi_{X_{17},X_{21}}$ is a bidirected path; an undirected subgraph contains X_1, X_2 and a bidirected subgraph contains X_{17}, X_{20}, X_{21}.

Let X, Y, Z be variables or sets of variables, $\langle X, Y|Z\rangle$ denotes that X and Y are conditionally independent given Z; otherwise, $\langle X, Y \nmid Z\rangle$. $\langle X, Y\rangle$ refers to the fact that X and Y are marginally independent; otherwise, $\langle X, Y \nmid \emptyset\rangle$. We use *CI* tests to determine the conditional independence. In ancestral graphs, independency relations can be identified using m-separation (Richardson & Spirtes, 2002).

Definition 1. ***m-separation**. Two vertices μ and ν are m-separated given Z in G — denoted as $\langle\mu, \nu|Z\rangle_m$ — where $Z \subseteq V(G)\backslash\{\mu, \nu\}$ if and only if every path between μ and ν contains either one triple from (1) $\alpha \to \beta \to \gamma$, $\alpha \leftrightarrow \beta \to \gamma$, $\alpha \leftarrow \beta \to \gamma$, $\alpha - \beta \to \gamma$, $\alpha - \beta - \gamma$, and $\beta \in Z$, or one triple from (2) $\alpha \to \beta \leftarrow \gamma$, $\alpha \leftrightarrow \beta \leftarrow \gamma$, $\alpha \leftrightarrow \beta \leftrightarrow \gamma$, and $De(\beta) \cap Z = \emptyset$.*

Two vertex sets X, Y are m-separated given Z if all the paths from X to Y are m-separated by Z. Figure 1(b) indicates the examples that $\langle X_2, X_4|X_3\rangle$, $\langle X_2, X_4 \nmid \emptyset\rangle$, $\langle X_2, X_8\rangle$, $\langle X_2, X_8 \nmid X_3\rangle$, $\langle X_3, X_5\rangle$, $\langle X_3, X_5 \nmid X_4\rangle$, $\langle X_2, X_9|X_3, X_4\rangle$ and $\langle X_3, X_6 \nmid X_4, X_5\rangle$.

Definition 2. ***Ancestral graph (AG)**(Ali et al., 2009). A graph G is an ancestral graph if and only if three conditions hold: (i) there are no directed cycles; (ii) if there is an undirected edge $\alpha - \beta$, then α and β have neither spouses nor parents; (iii) wherever there is a bidirected edge $\alpha \leftrightarrow \beta$, no directed path passes from α to β, or from β to α.*

The attributes and examples of AG, as well as the difference between AGs and Bayesian networks, have been clarified in (Richardson & Spirtes, 2002; Ali et al., 2009)

Definition 3. ***Maximal ancestral graph (MAG)** (Ali et al., 2009). An ancestral graph is maximal if and only if there exists Z such that $\langle\alpha, \beta|Z\rangle$ for any un-adjacent pair $\alpha, \beta \in V(G)$, where $Z \subseteq V(G)\backslash\{\alpha, \beta\}$.*

In ancestral graphs, vertices represent observed variables and edges represent causal relations. Many examples of AGs and MAGs were provided in (Zhang, 2008; Ali et al., 2009). The polytree network in Figure 1(b) is an AG. Regarding the interpretation of directed, undirected and bidirected edges: (1) $\alpha \to \beta$ indicates that the appearance of α cultivates β which might be due to a direct cause (Pearl, 1988); (2) $\alpha \leftrightarrow \beta$ indicates that an unobserved variable L exists in the path $\alpha \leftarrow L \to \beta$, whereas neither α causes β nor β causes α (Zhang, 2008); (3) $\alpha - \beta$ indicates that α is associated with β with no certainty whether α causes β or vice-versa, due to selection bias (Zhang, 2008).

3. Ancestral polytree

In this section, we present our ancestral polytree models, which combine the concept of ancestral graphs with the idea of polytree structures.

Definition 4. ***Ancestral polytree (AP)**. A graphical model $G(V, E)$ with $E \subseteq \{\alpha\ op\ \beta \mid \alpha, \beta \in V \wedge op \in \{\leftarrow, \to, \leftrightarrow, -\}\}$ is an ancestral polytree if and only if two conditions hold: (i) it is singly connected; (ii) if it has an undirected edge $\alpha - \beta$, neither α nor β has any spouse or parent.*

Figure 1(b) demonstrates an example of AP. Moreover, it has been proven that an AG is maximal if and only if there is no inducing path between any non-adjacent vertices (Corollary 4.4, (Richardson & Spirtes, 2002)). Since APs are singly connected, the conditions (i) and (ii) in the definition of AGs are guaranteed so that any AP is also an AG. Because there is no inducing path in singly connected AP so that any AP is an MAG.

As the subgraphs of causal polytrees, *causal basins* start with ν-structures, and continue in the direction of directed paths to traverse the children's descendants and the direct parents of these descendants (Pearl, 1988). For instance, the vertices $\{X_2, X_3, X_4, X_8, X_9, X_{10}\}$ in Figure 1(b) form a causal basin (also see examples on page 393 of (Pearl, 1988)). We herein introduce ancestral basins in ancestral polytrees.

Definition 5. ***Ancestral basin**. A subgraph of an ancestral polytree G is an ancestral basin G^B if it starts with a ν-structure containing a starting collider, and continues in the direction of directed or bidirected paths to pass every linked vertex, whose archaics include at least one collider, or being a parent of a collider.*

Definition 6. ***Simple ancestral polytree (SAP)**. Ancestral polytree G is a SAP if and only if its edges are either in ancestral basins or in undirected paths.*

Proposition 1. *Both β, γ are colliders if $\beta \leftrightarrow \gamma \in E(G^S)$.*

Proof. $Ar(\beta)$ contains at least one collider since G^S is a SAP. If $Ar(\beta) \subset \{\beta\}$, then β itself is a collider; if $Ar(\beta) \supset \{\beta\}$, at least one vertex $\alpha \in Ar(\beta)$ satisfies $\alpha * \to \beta \leftrightarrow \gamma$. It ensures that β is a collider, so is γ. □

A *starting collider*, also termed as a multi-parent node (Pearl, 1988) or articulation point (Ouerd et al., 2004), represents the vertex β on a path containing $-\alpha \to \beta \leftarrow \gamma-$, where an ancestral basin begins. For instance, the subgraph containing $\{X_2, X_3, X_4, X_5, X_6, X_8, X_9, X_{10}\}$ in Figure 1(b) is an ancestral basin whose starting collider is X_3. Yet, neither the subgraph containing $\{X_{11}, X_{12}, X_{13}, X_{14}\}$ nor the one containing $\{X_{15}, X_{17}, X_{18}, X_{19}, X_{20}, X_{21}\}$ is an ancestral basin, because there is no starting collider in both subgraphs, and neither the parents of X_{11}, X_{19} nor the anteriors of X_{11}, X_{19} contain any collider. The AP in Figure 1(b) becomes an SAP after replacing $X_{12} \rightarrow X_{13}$, $X_{19} \rightarrow X_{18}$ with $X_{12} \leftarrow X_{13}$, $X_{19} - X_{18}$, respectively.

Let $H(G)$ represent the independency in a causal diagram:

$$H(G) \equiv \{\langle X, Y|Z\rangle \mid \textit{Disjointed subsets } X, Y, Z \subseteq V(G)\}$$

Definition 7. ***Markov equivalence*** *(Ali et al., 2009). Two ancestral graphs G_1 and G_2 are Markov equivalent, $G_1 \sim G_2$, if $H(G_1) = H(G_2)$.*

Proposition 2. *$G_1 \sim G_2$ if and only if APs G_1 and G_2 have the same skeletons and ν-structures.*

Due to limited space, a concise proof includes two parts. (Necessary) Note that both APs G_1 and G_2 are also MAGs. If G_1, G_2 are MAGs and $G_1 \sim G_2$, then G_1, G_2 have the same adjacencies and ν-structures (Proposition 3.6 in (Ali et al., 2009)). (Sufficient) It was proven that if MAGs G_1, G_2 share the same skeleton and colliders with order, then $G_1 \sim G_2$ (Theorem 3.7, (Ali et al., 2009)). If APs G_1 and G_2 have the same ν-structures, they have the same colliders with order because there is no discriminating path (definition 3.8 and 3.11, (Ali et al., 2009)) in any AP.

Above proposition leads to a sufficient and necessary condition to identify Markov equivalent APs. However, instead of investigating all equivalent structures, it is vital to identify one essential graph which is sufficient to represent all Markov equivalent structures.

Definition 8. ***Essential graph***. *Let $[G]$ denote Markov equivalence class of ancestral polytree G, whose conditional independencies are identical. M_G is the essential graph of G if and only if M_G satisfies two conditions, (i) M_G shares the same skeleton with all APs in $[G]$; (ii) any directed or bidirected edge exits in M_G if and only if it is shared by all APs in $[G]$.*

Note that essential graphs are equivalent to partial ancestral graphs (Zhang, 2008) if AGs are restricted to be polytrees. Let G^S be a SAP, due to the condition (ii) above, $G^S \in [G]$ guarantees that G^S includes all directed and bidirected edges in M_G. This fact leads to the next proposition.

Proposition 3. *M_G is a simple ancestral polytree.*

The factorization of acyclic directed mixed graphs with directed and bidirected edges was studied in (Richardson, 2009). The factorization of Gaussian distribution in MAG can be decomposed as $f(X_V) = f(X_{un_G})f(X_{V\backslash un_G} \mid X_{un_G})$, where un_G and $V\backslash un_G$ contain vertices of undirected and directed subgraphs in MAG, respectively (Richardson & Spirtes, 2002). To assess structure learning and inference, the next proposition clarifies the factorization of joint probability of ancestral polytrees.

Proposition 4. *Given an ancestral polytree $G = (V, E)$, the joint probability of random variables can be decomposed into products of conditional probability distributions as:*

$$P_G(X_V) = \frac{1}{Z} \prod_{X_i - X_j \in E(S^U)} \psi(X_i, X_j) \times \textstyle\prod_{b \in S^B} P(X_b|Ant(X_b))$$
$$\times \prod_{X \in S^D} P(X \mid Ant(X), Ar(X))$$

Where S^B is the set of subgraphs containing bidirected edges, S^U is the set of subgraphs containing undirected edges, S^D is the set of subgraphs containing directed edges, ψ is a factor potential of a clique formed with edge $\alpha - \beta$ as a non-negative function, the normalization coefficient is defined as $Z = \int_{X_{V(S^U)}} \prod_{\alpha - \beta \in S^U} \psi(\alpha, \beta) dX$.

Proof. $P_G(X) = P(X_{S^U})P(X_{S^B}, X_{S^D}|X_{S^U}) = P(X_{S^U})P(X_{S^B}|Ne(X_{S^B}), X_{S^U})P(X_{S^D}|Ne(X_{S^D}), X_{S^U})$. Firstly, given any undirected subgraph s in S^U and any bidirected subgraph b in S^B, the factorization can be expressed as $\prod_{c \in C(s)} \psi_c(X)/Z^*$, where $C(s)$ is the set of cliques in s and Z^* is the normalization coefficient. Note that undirected subgraphs in S^U are disjointed and cliques in singly connected S^U contain two neighbouring nodes α, β on an undirected edge $\alpha - \beta$. Therefore, the factorization of S^U is:

$$P(X_{S^U}) = \prod_{s \in S^U} \frac{1}{Z^*} \prod_{c \in C(s)} \psi_c(X) = \frac{1}{Z} \prod_{c \in C(S^U)} \psi_c(X)$$
$$= \frac{1}{Z} \prod_{X_i - X_j \in E(S^U)} \psi(X_i, X_j)$$

Secondly, Theorem 4 in (Richardson, 2009) reveals the factorization of bidirected subgraphs as:

$$P(X_{S^B}|Ne(X_{S^B}), X_{S^U}) = \prod_{b \in S^B} P(X_b \mid Pa(X_b), V(S^U))$$
$$= \prod_{b \in S^B} P(X_b \mid Ant(X_b))$$

Thirdly, the factorization of directed subgraphs is:

$$P(X_{S^D}|Ne(X_{S^D}), X_{S^U}) = \prod_{X \in S^D} P(X|Pa(X), V(S^B), V(S^U))$$
$$= \prod_{X \in S^D} P(X \mid Ant(X), Ar(X))$$

The proof is complete by multiplying three parts. □

Particularly, if the entire ancestral polytree G is an ancestral basin, we have $S^U = \emptyset$, $S^D = V - V(S^B)$, $Ant(X) =$

$Ar(X)$. Mimicking each vertex in S^D as an entire bidirected subgraph, the factorization of ancestral basin G^B is:

$$P_{G^B}(X_V) = \prod_{X \in V - V(S^B)} P(X|Ant(X)) \times \prod_{b \in S^B} P(X_b|Ant(X_b))$$
$$= \prod_{b \in S^B \cup (V - V(S^B))} P(X_b \mid Ant(X_b))$$

Examples in Figure 2 include: $V(S^U) = \{\{X_1, X_2\}, \{X_6, X_7\}, \{X_{10}, X_{11}, X_{15}, X_{16}\}\}, V(S^B) = \{\{X_4, X_5\}, \{X_{17}, X_{20}, X_{21}\}\}, V(S^D) = \{\{X_9\}, \{X_{12}, X_{13}, X_{14}\}, \{X_3, X_8\}, \{X_{18}, X_{19}\}\}$. The factorization of AP with configuration X=x in Figure 2 is: $P(X = x) = [1/Z\ \psi(x_1, x_2)\psi(x_6, x_7)\psi(x_{10}, x_{11})\psi(x_{11}, x_{15})\psi(x_{15}, x_{16})] \times [p(x_{19})p(x_{18}|x_{19})p(x_{13}|x_{12})p(x_{14}|x_{12})p(x_3|x_1, x_2, x_8, x_9, x_{10}, x_{11}, x_{15}, x_{16})p(x_{12}|x_{10}, x_{11}, x_{15}, x_{16})] \times [p(x_4, x_5|x_3, x_6, x_7)p(x_{17}, x_{20}, x_{21}|x_{10}, x_{11}, x_{15}, x_{16}, x_{18})]$

4. Learning ancestral polytree

Given a training dataset, the methodology of training causal polytrees usually involves two stages. Firstly, undirected skeletons are trained either by maximum weighted spanning trees (Pearl, 1988) or by CI tests (Huete & de Campos, 1993). Secondly, the directionality of edges in undirected skeletons are recovered by orienting principles (Pearl, 1988; Ouerd et al., 2004) including two rules. Rule 1: for all $\alpha - \beta - \gamma$ and $\langle\alpha, \gamma\rangle$, orient $\alpha - \beta - \gamma$ into $\alpha \rightarrow \beta \leftarrow \gamma$. Rule 2: for remaining $\alpha \rightarrow \beta - \gamma$, orient $\alpha \rightarrow \beta - \gamma$ into $\alpha \rightarrow \beta \rightarrow \gamma$. Based on the m-separation, orienting principles for learning ancestral polytrees rely on the orienting principles of ancestral polytree (OPAP), which also includes two rules:

Rule 1 : for all $\alpha * - * \beta * - * \gamma$ and $\langle\alpha, \gamma\rangle$, orient $\alpha * - * \beta * - * \gamma$ into $\alpha * \rightarrow \beta \leftarrow * \gamma$.

Rule 2 : for remaining $\alpha * \rightarrow \beta - \gamma$, orient $\alpha * \rightarrow \beta - \gamma$ into $\alpha * \rightarrow \beta \rightarrow \gamma$.

Based on the OPAP, we have designed a structure learning algorithm using three tips: (1) search colliders through visiting inner vertices from the one with the most undirected edges to the one with the least, because every collider is an inner vertex. (2) For each selected inner vertex, examine CI test $\langle\alpha, \gamma\rangle$ on the triplet $\alpha - \beta - \gamma$ first, and examine CI test on $\alpha \leftarrow \beta \rightarrow \gamma$ last (because it is comparatively rare for both α and γ to be colliders). (3) Distinguish bidirected from undirected edges by withdrawing detected bidirected edges, and recover them back into the oriented structure.

```
Algorithm: Learning Ancestral Polytree (LAP)
Input: A training dataset and a polytree skeleton $G^U$.
Output: A partially oriented ancestral polytree $G$.
Abbreviations: $UV$, the set of unvisited inner vertices; $CS$, the set of colliders; $BA$, the set of bidirected edges; $V_{In}$, the set of inner nodes; $CI$, the set of CI test.
Initiate $CI = BA = \emptyset$, $UV = V_{In}, G = G^U$.
While $UV \neq \emptyset$
    $\beta = \arg\max_{v \in UV} |Ne_G(v)|$; $UV = UV \backslash \{\beta\}$;
    For all $\alpha - \beta - \gamma \in E(G)$ and $(\alpha, \gamma) \notin CI$
        Do $CI = CI \cup \{(\alpha, \gamma)\}$; if $\langle\alpha, \gamma\rangle$, orient $\alpha - \beta - \gamma$ into $\alpha \rightarrow \beta \leftarrow \gamma$ and $CS = CS \cup \{\beta\}$;
    End for
    For all $\alpha - \beta \leftarrow \gamma \in E(G)$ and $(\alpha, \gamma) \notin CI$
        Do $CI = CI \cup \{(\alpha, \gamma)\}$; if $\langle\alpha, \gamma\rangle$, orient $\alpha - \beta \leftarrow \gamma$ into $\alpha \rightarrow \beta \leftarrow \gamma$ and $CS = CS \cup \{\beta\}$;
    End for
    For all $\alpha - \beta \rightarrow \gamma \in E(G)$ and $(\alpha, \gamma) \notin CI$
        Do $CI = CI \cup \{(\alpha, \gamma)\}$; if $\langle\alpha, \gamma\rangle$, orient $\alpha - \beta \rightarrow \gamma$ into $\alpha \rightarrow \beta \leftrightarrow \gamma$ and $CS = CS \cup \{\beta\}$,$E(G) = E(G) \backslash \{\beta \leftrightarrow \gamma\}, BA = BA \cup \{(\beta, \gamma)\}$;
    End for
    For all $\alpha \leftarrow \beta \rightarrow \gamma \in E(G)$ and $(\alpha, \gamma) \notin CI$
        Do $CI = CI \cup \{(\alpha, \gamma)\}$; if $\langle\alpha, \gamma\rangle$, orient $\alpha \leftarrow \beta \rightarrow \gamma$ into $\alpha \leftrightarrow \beta \leftrightarrow \gamma$ and $CS = CS \cup \{\beta\}$, $E(G) = E(G) \backslash \{\alpha \leftrightarrow \beta, \beta \leftrightarrow \gamma\}, BA = BA \cup \{(\alpha, \beta), (\beta, \gamma)\}$;
    End for
    For all $\alpha \rightarrow \beta - \gamma \in E(G)$
        Do orient $\alpha \rightarrow \beta \rightarrow \gamma$;
    End for
End while
For all $\alpha \in CS$, create an empty queue $Q$, $push(Q, \alpha)$;
  While $Q \neq \emptyset$
    $\beta = pop(Q), UV = UV \backslash \{\beta\}$;
    For all $\gamma \in UV, \beta - *\gamma \in E(G)$
      Do orient $\beta - \gamma$ into $\beta \leftarrow \gamma, push(Q, \gamma)$;
    End for
  End while
End for
For all $(\alpha, \beta) \in BA$
    Do $E(G) = E(G) \cup \{\alpha \leftrightarrow \beta\}$.
End for
```

Return G

LAP has two major parts. The first part in the "while" loop and the second part in the "for" loop orient the unvisited edges based on the rule 1 and rule 2 of OPAP, respectively. Using CI tests, each of the five "for" loops in the first part orients $\alpha - \beta - \gamma, \alpha - \beta \leftarrow \gamma, \alpha - \beta \rightarrow \gamma, \alpha \leftarrow \beta \rightarrow \gamma$ and $\alpha \rightarrow \beta - \gamma$, respectively. The visited edges and nodes are recorded to avoid repeating tests. The second part uses the depth-first traversal to visit all colliders in ancestral basins and to orient the undirected edges subsequently.

Ideally, consider there is an oracle to provide CI information from a faithful ancestral polytree G, denoted as $G_{\mathcal{T}}$. An algorithm is *sound* if it outputs a predicted G that $G \in [G_{\mathcal{T}}]$, and algorithm is *complete* if it predicts a maximally informative G for $[G_{\mathcal{T}}]$ (Zhang, 2008). The soundness and completeness of 11 orientation rules have been proven to train ancestral graphs (Zhang, 2008). These 11 orientation rules can be simplified into OPAP if AGs are singly connected, which ensures the soundness and completeness of LAP. Remind that two OPAP rules were also included in the Fast Causal Inference algorithm (*F-CI*) (Spirtes et al., 1995) and the augmented *FCI* (Zhang, 2008), both of which can model equivalent structures as LAP if underlying structures were singly connected.

Many studies have endeavored to learn polytree skeletons and well-known algorithms for maximum-likelihood learning of tree distributions have achieved the complexity of $\mathcal{O}(n^2 \log(n))$ (Leiserson et al., 2001). Herein we analyze the computational complexity of LAP to show that LAP can achieve a fast structure learning using refined orienting procedures.

Proposition 5. *Suppose skeleton G^U is an undirected tree which has one root with K adjacent vertices, and has inner vertices all with K+1 adjacent vertices. Let N be the number of vertices in G^U, the number of required CI tests $R(G^U)$ satisfies:*

$$R(G^U) \leq (K+1)(N-1)/2 - K$$

Proof. Let H be the depth of tree G^U, we have $N = \sum_{i=1}^{H} K^{i-1} = (K^H - 1)/(K - 1)$. Therefore, $\sum_{i=2}^{H-1} K^{i-1} = (K^{H-1} - K)/(K-1) = (N-1)/K - 1$. The number of required CI tests is maximal if we test $\langle \alpha, \gamma \rangle$ in all triplets formed as $\alpha - \beta - \gamma$. Consider every inner vertex has $K(K+1)/2$ pairs of adjacent vertices except the root, the summation of CI tests over all inner vertices satisfies:
$R(G^U) \leq K(K-1)/2 + K(K+1)\sum_{i=2}^{H-1} K^{i-1}/2 = (K+1)(N-1)/2 - K$. □

Proposition 5 analyzes the complexity of LAP for a special case of skeleton G^U. For the general cases, we then consider the average CI tests regarding the entire space of the set of marginally independent tests $\mathcal{T}$, denoted as Ω. Given an arbitrary G^U, let $f(n)$ count the number of $\mathcal{T}$ that requires n CI tests to train G^U. In fact, the set of $f(n)$ has $k(k-1)/2 - \lceil k/2 \rceil + 1$ elements where $n = \lceil k/2 \rceil, \lceil k/2 \rceil + 1, \cdots, k(k-1)/2$ and $\lceil k/2 \rceil$ denotes the upper bound of integer upon $k/2$ (e.g. $\lceil 3/2 \rceil = 2$). The average CI tests of inner node v is:

$$E[R(G_v^U)] = \frac{1}{|\Omega|} \sum_{n=\lceil k/2 \rceil}^{k(k-1)/2} [n \times f(n)]$$

To date, we have not found any simple formula to decode $f(n)$ whose experiment data is:

f(2)={1},f(3)={2,6},f(4)={16,8,8,14,18},f(5)={128,192, 192,224,104,72,62,50}, f(6)={4096,4096,4096,3584,3584, 3968,3520,2560,1776,744,392,222,130}, f(7)={131072, 262144, 327680,311296,262144,233472,169984,123904, 88064,66560, 50048,33856,19808,10480,3944,1672,702, 322}, f(8)={16777216,25165824, 29360128,29360128, 27787264,25165824, 22413312,18874368,14680064, 10485760,7061504,4673536, 2883584,1775616,1086464, 681728,415488,233792,116128,51248,17384,6152,2046, 770}.

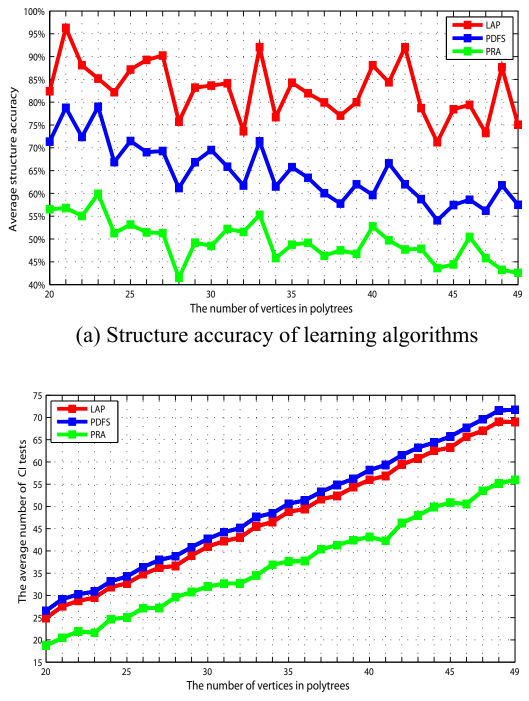
Even so, we have observed that regression methods can estimate $E[R(G_v^U)]$ sufficiently. Particularly, we have found similar results using both least square and robust linear regressions, which use iteratively re-weighted least squares. The output of least square regression is: $E[R(G_v^U)] = -1.3439 + 1.3425K$, where the root mean square error is $\varepsilon = 0.2652$ and the maximum residue is $\varepsilon_{max} = 0.3126$, observed at $K = 5$. Based on the above estimation, we have:

$$E[R(G^U)] = \sum_{V_i \in V_{In}} [\frac{1}{|\Omega|} \sum_{T \in \Omega} R(G_{V_i}^U, \mathcal{T})] = \sum_{V_i \in V_{In}} E[R(G_{V_i}^U)]$$

If $K = \max_{v \in V} |Ne_{G^U}(v)| \leq 8$, $E[R(G^U)] \leq |V_{In}| \times (1.3425K - 1.3439 + \varepsilon_{max})$. In other words, the average CI tests for each inner node are bound by 1.3425K-1.0313 if the maximal graphical degree meets the condition of K≤8.

5. Experiments

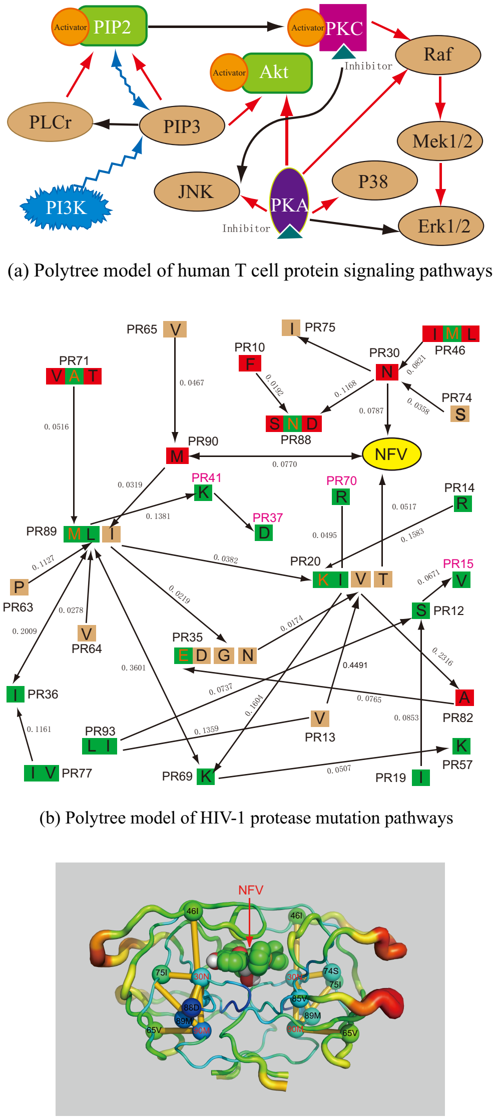
Four experiments were carried out in this study: (1) we compared LAP with both the causal Polytree Recovery Algorithm (PRA) (Pearl, 1988) and the Polytree-Depth-First-Search (PDFS) algorithm (Ouerd et al., 2004) using a synthetic dataset, (2) we applied LAP to model protein signaling pathways using a human immune cell dataset (Sachs et al., 2005), (3) we explored the HIV-1 resistance



(a) Structure accuracy of learning algorithms

(b) Number of CT tests required by learning algorithms

Figure 3. Average structure accuracy (a) and umber of CI tests (b) required by LAP, PDFS and PRA on the a synthetic dataset.



(a) Polytree model of human T cell protein signaling pathways

(b) Polytree model of HIV-1 protease mutation pathways

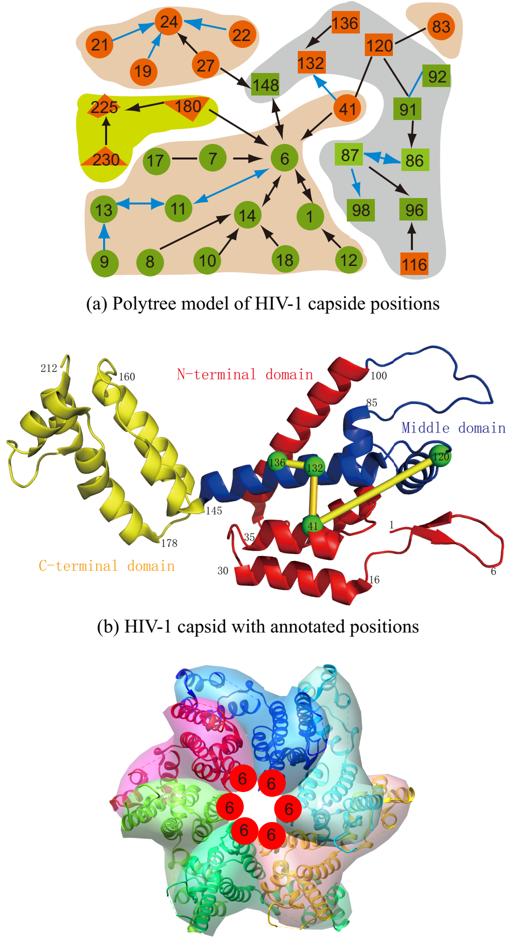
(c) HIV-1 protease with annotated mutations

Figure 4. (a) Human T cell protein signaling pathway network modeled by LAP. Pink circles and blue triangles represent activators and inhibitors respectively. Other nodes represent proteins in the signaling pathways. Red arrows are faithfully predicted edges, black arrows are undetected edges but reported by biological studies and blue arrows are recovered assuming that the protein PI3K is observed (Sachs et al., 2005). (b) Ancestral polytree network of HIV-1 protease mutations selected by the protease inhibitor NFV. NFV is colored yellow and colored squares distinguish protease drug resistance mutations from wild type residues. Mutations from the same residue position are clustered and mutual information is notified on edges. (c) HIV-1 protease structure. Residue positions are annotated accordingly. The HIV-1 protease PDB is 2QAK. Visualization software: PyMOL v1.5).

mutation pathways by LAP using the HIV-1 protease inhibitor nelfinavir (NFV) dataset (Deforche et al., 2006), and (4) we used LAP to model the interaction networks in the HIV-1 capsid protein. In our experiments, the algorithm performance was evaluated using structure accuracy defined as: $Acc = (|\{(X,Y)|(X,Y) \in E, (X,Y) \in E^*, (Y,X) \notin E^*\}|)/(|V|-1)$, where $G = (V,E)$ and $G^* = (V,E^*)$ are the known and the predicted structures, respectively. We used non-parametric bootstrapping with 100 replicates to cultivate polytree skeletons from the undirected maximal spanning tree algorithm based on mutual information (Chow & Liu, 1968). We used Fisher's exact test for CI tests (confidence level: 95%) and Laplace smoothing for probability calculations.

In the first experiment, we compared LAP with two causal polytree algorithms PDFS and PRA using a synthetic dataset. The synthetic data contained 6000 random polytrees with variable numbers (ranging from 20 to 49) and CI sets randomly generated. For each variable number, 200 unique skeletons were randomly sampled for our analysis. Figure 3(a) illustrates the comparison results, showing that LAP (average structure accuracy: 82.72%) performs better than the PDFS (64.6%) and PRA (49.54%) algorithms. We also compared the number of CI tests required by the polytree algorithms, illustrated in the Figure 3(b). The average CI tests for PDFS, LAP and PRA were found to be 49.51, 47.53 and 37.02, respectively. This suggests that the weak structure accuracies of PRA are compromised by the least CI tests, and LAP performs better than PDFS.

In the second experiment, LAP was applied to flow cytom-



(a) Polytree model of HIV-1 capside positions

(b) HIV-1 capsid with annotated positions

(c) Residue position 6 in HIV-1 capsid hexamer

Figure 5. (a) Ancestral polytree modeling of residue interaction networks in HIV-1 capsid. Green and pink indicate residue positions in the loop and helix structures of capsid, respectively. Circle, square and triangle represent positions in the C-terminal (1-84), middle (85-145) and N-terminal (146-231) domains. Red edges indicate positions whose C_α atoms are closer than 10Å of the Euclidean distance in the capsid structure. (b) HIV-1 capsid structure (PDB:3P05, visualized by PyMOL v1.5). The residue positions 41, 120, 132, 136 across different functional domains are annotated, as well as major clusters in two loop regions (positions: 1-16 and 85-100). Yellow links indicate the associations between residue positions 136, 132, 41 and 120, predicted by our ancestral polytree model. (c) Position 6 in the structure of HIV-1 capsid hexamer (PDB:3H4E, visualized by Chimera v1.7).

etry datasets of human T cell protein signaling pathways (Sachs et al., 2005). The data was collected through intracellular multicolor flow cytometry, which measures the protein expression levels of 11 proteins with single-cell data points (Sachs et al., 2005). We first removed the outliers whose values were 3 times larger than the mean values and discretized the continuous data using an information preserving algorithm (Hartemink, 2001). By doing so, 500 datasets were created containing data for 400 cells each. Using 100 bootstrap samples on each dataset lead to 500 trained APs. Figure 4(A) shows a consensus AP which recovers 10 out of 14 expected signaling pathways, while BN analysis recovered 12 (Sachs et al., 2005). Note that the bidirected edge PIP2↔PIP3 was recovered assuming that the protein PI3K was observed.

In the third experiment, we modeled the interaction network of HIV-1 protease mutations selected by the protease inhibitor nelfinavir (NFV). We trained polytree networks using the HIV-1 NFV dataset, which includes 1307 HIV-1 protease amino acid sequences sampled from 967 drug-naive and 340 NFV-treated patients (Deforche et al., 2006). The trained polytree network is illustrated in Figure 4(b). We mapped the mutation positions to the crystalized structure of HIV-1 protease (Figure 4(c)). We found that for protease mutations that had less than 5 edges to the NFV in the consensus AP, the Euclidean distances of C_α atoms between these mutations had less than 10 angstroms in the 3D structure. Moreover, our AP shared 38 out of 58 edges compared to the trained BNs, described previously in (Deforche et al., 2006). AP may help to study the associations between HIV-1 drug resistance mutations, whereas causal effects in drug resistance pathways and the impact of unobserved variables require further investigations.

In our last experiment, we modeled the interaction networks of natural residues in HIV-1 capsid - a hexamer protein in the length of 231 residue positions. HIV-1 capsid is a key protein to construct the structural surface of HIV-1 mature virions (Li et al., 2013). We followed the procedure described in (Li et al., 2013) to collect HIV-1 capsid sequences sampled from 787 treatment naive patients in the HIV Los Alamos database. We removed both duplicate sequences and sequences with stop codons. We aligned nucleotide sequences using Seaview v4.4.0. Figure 5(a) shows our consensus ancestral polytree for HIV-1 capsid. It suggests that positions are clustered in AP regarding to the loop and helix regions in functional domains of capsid (Figure 5(b)), except the positions 83 and 116. Being crucial for protein multimerization, position 6 connected with many positions in AP (Figure 5(c)). Potentially, positions from the same functional domains tempt to cluster in our causal network may explain the role of structural constraints. As shown in our recent study (Li et al., 2013), this information can be useful for designing novel inhibitors targeting HIV-1 capsid.

6. Conclusions and future work

This study introduces ancestral polytrees and their simple structures which guarantee fast learning algorithms. Our future study will focus on maximum likelihood and inference problems. We will also apply polytree models to large biological networks such as genomic interaction networks.

Acknowledgments

This work was supported by the Fonds voor Wetenschappelijk Onderzoek Flanders (FWO) [G069214N]; the European Community's Seventh Framework Programme (FP7/2007-2013) under the project Collaborative HIV and Anti-HIV Drug Resistance Network (CHAIN) [223131]. Jan Ramon is supported by ERC StG 240186 "MiGraNT: Mining Graphs and Networks, a Theory-based approach".

References

Ali, R Ayesha, Richardson, Thomas S, and Spirtes, Peter. Markov equivalence for ancestral graphs. *The Annals of Statistics*, 37(5B):2808–2837, 2009.

Campos, Luis M. de. Independency relationships and learning algorithms for singly connected networks. *Journal of Experimental & Theoretical Artificial Intelligence*, 10(4):511–549, 1998.

Chow, C and Liu, C. Approximating discrete probability distributions with dependence trees. *Information Theory, IEEE Transactions on*, 14(3):462–467, 1968.

Dasgupta, Sanjoy. Learning polytrees. In *15th Conf. on UAI*, pp. 134–141, 1999.

Deforche, Koen, Silander, T, Camacho, R, Grossman, Z, Soares, MA, Van Laethem, Kristel, Kantor, R, Moreau, Yves, Vandamme, A-M, et al. Analysis of hiv-1 pol sequences using bayesian networks: implications for drug resistance. *Bioinformatics*, 22(24):2975–2979, 2006.

Hartemink, Alexander John. *Principled computational methods for the validation of and discovery of genetic regulatory networks*. PhD thesis, MIT, 2001.

Huete, Juan F and de Campos, Luis M. Learning causal polytrees. In *Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pp. 180–185. Springer, 1993.

Leiserson, Charles E, Rivest, Ronald L, Stein, Clifford, and Cormen, Thomas H. *Introduction to algorithms*. The MIT press, 2001.

Li, Guangdi, Verheyen, Jens, Rhee, Soo-Yon, Voet, Arnout, Vandamme, Anne-Mieke, and Theys, Kristof. Functional conservation of hiv-1 gag: implications for rational drug design. *Retrovirology*, 10(1):126, 2013.

Lodish, Harvey, Berk, Arnold, Matsudaira, Paul, Kaiser, Chris A., Krieger, Monty, Scott, Matthew P., Zipursky, Lawrence, and Darnell, James. *Molecular cell biology*. WH Freeman, 5th edition, 2004.

Messaouda, Ouerd, Oommen, John B, and Matwin, Stan. Enhancing caching in distributed databases using intelligent polytree representations. In *Advances in Artificial Intelligence*, pp. 498–504. Springer, 2003.

Ouerd, M, Oommen, BJ, and Matwin, S. A formal approach to using data distributions for building causal polytree structures. *Information Sciences*, 168(1):111–132, 2004.

Pearl, Judea. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausble Inference*. Morgan Kaufmann Pub, 1988.

Pearl, Judea. *Causality: models, reasoning and inference*. Cambridge Univ Press, 2000.

Pena, Jose M. Reading dependencies from polytree-like bayesian networks. In *23th Conf. on UAI*, pp. 303–309, 2007.

Rebane, George and Pearl, Judea. The recovery of causal poly-trees from statistical data. In *3rd Conf. on UAI*, pp. 222–228, 1987.

Richardson, Thomas and Spirtes, Peter. Ancestral graph markov models. *The Annals of Statistics*, 30(4):962–1030, 2002.

Richardson, Thomas S. A factorization criterion for acyclic directed mixed graphs. In *25th Conf. on UAI*, pp. 462–470, 2009.

Sachs, Karen, Perez, Omar, Pe'er, Dana, Lauffenburger, Douglas A, and Nolan, Garry P. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523, 2005.

Spirtes, Peter, Meek, Christopher, and Richardson, Thomas. Causal inference in the presence of latent variables and selection bias. In *11th Conf. on UAI*, pp. 499–506, 1995.

Zaveri, Mazad S and Hammerstrom, Dan. Cmol/cmos implementations of bayesian polytree inference: Digital and mixed-signal architectures and performance/price. *Nanotechnology, IEEE Transactions on*, 9(2):194–211, 2010.

Zhang, Jiji. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, 172(16):1873–1896, 2008.