Journal of Machine Learning Research 16 (2015) 3219-3267 Submitted 2/14; Revised 4/15; Published 12/15

Ultra-Scalable and Efficient Methods for Hybrid Observational and Experimental Local Causal Pathway Discovery

Alexander Statnikov ALEXANDER.STATNIKOV@MED.NYU.EDU
Sisi Ma SISI.MA@NYUMC.ORG
Mikael Henaff MBH305@NYU.EDU
Nikita Lytkin NIKITA.LYTKIN@GMAIL.COM
Efstratios Efstathiadis STRATOS@NYU.ORG
Eric R. Peskin ERIC.PESKIN@NYUMC.ORG
Center for Health Informatics and Bioinformatics
New York University School of Medicine
New York, NY 10016, USA
Constantin F. Aliferis CALIFERI@UMN.EDU
Institute for Health Informatics
University of Minnesota
Minneapolis, MN 55455, USA

Editor: Peter Sprites

Abstract

Discovery of causal relations from data is a fundamental objective of several scientific disciplines. Most causal discovery algorithms that use observational data can infer causality only up to a statistical equivalency class, thus leaving many causal relations undetermined. In general, complete identification of causal relations requires experimentation to augment discoveries from observational data. This has led to the recent development of several methods for active learning of causal networks that utilize both observational and experimental data in order to discover causal networks. In this work, we focus on the problem of discovering local causal pathways that contain only direct causes and direct effects of the target variable of interest and propose new discovery methods that aim to minimize the number of required experiments, relax common sufficient discovery assumptions in order to increase discovery accuracy, and scale to high-dimensional data with thousands of variables. We conduct a comprehensive evaluation of new and existing methods with data of dimensionality up to 1,000,000 variables. We use both artificially simulated networks and *in-silico* gene transcriptional networks that model the characteristics of real gene expression data.

Keywords: causality, large-scale experimental design, local causal pathway discovery, observational data, experimental data, randomized experiments

©2015 Alexander Statnikov, Sisi Ma, Mikael Henaff, Nikita Lytkin, Efstratios Efstathiadis, Eric R. Peskin, and Constantin F. Aliferis.

1. Introduction

Discovery of causal relations from data is a fundamental objective of several scientific disciplines including computer science, statistics, and applied mathematics (Pearl, 2009; Spirtes et al., 2000; Neapolitan, 2003; Pearl, 1997). Obtaining data from randomized controlled experiments, while being essential for the discovery of causality, is very expensive and is often infeasible or unethical. On the other hand, observational data that is collected without experimental interference of the values of variables is highly abundant and can often be collected cheaply. Over the last 20 years, many sound algorithms have been proposed that can use observational data to infer causal relations (Pearl, 2009; Spirtes et al., 2000; Glymour and Cooper, 1999) and several empirical studies have verified their applicability and scalability to high-dimensional data (Aliferis et al., 2010a,b). However, observational data is, in general, insufficient to completely unravel all causal relations among measured variables, because many causal relations cannot be statistically distinguished with observational data alone (e.g., multiple graphs in the Markov equivalence class). Therefore, it is essential to refine discoveries from observational data with limited and targeted experimental data (Spirtes et al., 2000). This has led to the recent development of several methods for *active learning of causal networks* that utilize observational and experimental data in order to discover causal networks (Tong and Koller, 2001; Murphy, 2001; He and Geng, 2008; Meganck et al., 2006; Hyttinen et al., 2010; Eberhardt et al., 2010; Hyttinen et al., 2012; Pe'er et al., 2001; Sachs et al., 2005).

The present work is concerned with the problem of discovery of local causal pathways that only contain direct causes and direct effects of the target variable of interest, rather than learning the structure of the entire causal network that represents all causal relations among all measured variables. Knowledge of direct causes and effects is crucial for understanding the mechanisms of causality, and knowledge of direct causes particularly facilitates the design of effective interventions. Existing methods for discovery of local causal pathways fully rely on observational data and can discover causality up to a Markov equivalence class, leaving many causal relations undetermined (Spirtes et al., 2000; Aliferis et al., 2010a). Thus, experimental/manipulated data is needed to complement the discovery from observational data. For experimental/manipulated data, we consider here only data from fully randomized experiments (also known as *surgical* or *edge-breaking*). In the present study, all decisions about edge orientation are based on experimental data exclusively. It is noteworthy that the problem of local causal pathway discovery from observational and limited experimental data has not been addressed in the literature previously.

While developing new methods for local causal pathway discovery from observational and experimental data, we set four objectives. First, to *minimize the number of experiments* needed to refine discoveries from observational data. Second, to *relax sufficient assumptions of existing discovery methods* in order to take into account multiplicity of local causal pathways consistent with the data (Statnikov et al., 2013; Statnikov and Aliferis, 2010). The latter has potential to reduce the number of false negative and false positives predictions and improve overall discovery accuracy. Third, to *scale to very high-dimensional data* with many thousands of variables. Finally fourth, to *achieve sufficiently good structure discovery performance*.

As a result of this work, we introduce new ultra-scalable and experimentally efficient

local causal pathway discovery methods and conduct a comprehensive evaluation of new and existing techniques with high-dimensional data with up to 1,000,000 variables. We use both artificially simulated networks and *in-silico* gene transcriptional networks that model the characteristics of real gene expression data. In the latter networks, we focus on discovery of local causal transcriptional pathways of genes. Learning transcriptional pathways is one of the key problems in biomedicine and is a major component of the efforts to develop new diagnostics, vaccines and therapies that will diagnose, prevent and treat deadly human diseases.

The remainder of the paper is organized as follows. Section 2 provides general theory and background. Section 3 provides an overview and discussion of prior methods for active learning of causal networks and how these methods were applied in our study. Section 4 introduces new methods for local causal pathway discovery from observational and experimental data. Section 5 describes empirical assessment of methods in artificially simulated networks and realistic *in-silico* gene networks of high dimensionality. The paper concludes with Section 6, which summarizes the main findings and outlines directions for future work.

2. Background and Theory

In this section, general theory and background on causal modeling is provided.

2.1 Notation and Key Definitions

In this paper upper-case letters in italics denote random variables (e.g., A, B, C) and lower-case letters in italics denote their values (e.g., a, b, c). Upper-case bold letters in italics denote random variable sets (e.g., $\boldsymbol{X}$, $\boldsymbol{Y}$,$\boldsymbol{Z}$) and lower-case bold letters in italics denote their values (e.g., $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$). The terms variables and vertices are used interchangeably. If a graph contains an edge $X \rightarrow Y$, then X is a parent of Y and Y is a child of X. An undirected edge X — Y denotes an adjacency relation between X and Y (i.e., presence of an edge directly connecting X and Y). A path p is a set of consecutive edges (independent of the direction) without visiting a vertex more than once. A directed path p from X to Y is a set of consecutive edges with same direction ("$\rightarrow$") connecting X with Y, i.e. $X \rightarrow ... \rightarrow Y$. X is an ancestor of Y (and Y is a descendant of X) if there exists a directed path p from X to Y. A directed cycle is a nonempty directed path that starts and ends on the same vertex X. We consider in this work two types of graphs: (i) directed graphs where vertices are connected only with edges "$\rightarrow$" and (ii) directed acyclic graphs (DAGs) without directed cycles and where vertices are connected only with edges "$\rightarrow$".

When the two sets of variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ are conditionally independent given a set of variables $\boldsymbol{Z}$ in the joint probability distribution $\mathbb{P}$, we denote this as $\boldsymbol{X} \perp \boldsymbol{Y}|\boldsymbol{Z}$. For notational convenience, conditional dependence is defined as absence of conditional independence and denoted as $\boldsymbol{X} \not\perp \boldsymbol{Y}|\boldsymbol{Z}$. Two sets of variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ are considered independent and denoted as $\boldsymbol{X} \perp \boldsymbol{Y}$, when X and Y are conditionally independent given an empty set of variables. Similarly, the dependence of X and Y is defined and denoted as $\boldsymbol{X} \not\perp \boldsymbol{Y}$.

We further refer the readers to (Pearl, 2009; Spirtes et al., 2000; Neapolitan, 2003; Glymour and Cooper, 1999) to review the standard definitions of conditional independence, collider, blocked path, d-separation, and causal sufficiency that are used in this work. Be-

low we review only several essential definitions:

Definition of local Markov condition: The joint probability distribution $\mathbb{P}$ over variables $\boldsymbol{V}$ satisfies the local Markov condition for a directed acyclic graph (DAG) $\mathbb{G} =< \boldsymbol{V}, \mathbb{E} >$ if and only if for each W in $\boldsymbol{V}$, W is conditionally independent of all variables in V excluding descendants of W given parents of W (Richardson and Spirtes, 1999).

Definition of global Markov condition: The joint probability distribution $\mathbb{P}$ over variables $\boldsymbol{V}$ satisfies the global Markov condition for a directed graph $\mathbb{G} =< \boldsymbol{V}, \mathbb{E} >$ if and only if for any three disjoint subsets of variables $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$ from $\boldsymbol{V}$, if $\boldsymbol{X}$ is d-separated from $\boldsymbol{Y}$ given $\boldsymbol{Z}$ in $\mathbb{G}$ then $\boldsymbol{X}$ is independent of $\boldsymbol{Y}$ given $\boldsymbol{Z}$ in $\mathbb{P}$ (Richardson and Spirtes, 1999).

If the underlying graph $\mathbb{G}$ is a DAG, then the global Markov condition is equivalent to the local Markov condition (Richardson and Spirtes, 1999).

Definition of Bayesian network: $\mathbb{N} =< \mathbb{G}, \mathbb{P} >$ is a Bayesian network if the joint probability distribution $\mathbb{P}$ satisfies the local Markov condition for the DAG $\mathbb{G}$.

Next we provide an operational definition of causation and of a causal Bayesian network and local causal pathway. Notice that the following definition of causation matches the notion of randomized controlled experiment, which is the de facto standard for assessing macroscopic causation in the sciences (Pearl, 2009; Spirtes et al., 2000; Neapolitan, 2003; Glymour and Cooper, 1999).

Definition of causation, direct/indirect causation: Assume that a hypothetical experimenter can force a variable X to take specific values (i.e., to manipulate it). We say that X is a cause of Y (and Y is an effect of X) if the probability distribution of Y changes for some manipulation of X. X is the direct cause of Y with respect to $\boldsymbol{V}$, if: (i) X is a cause of Y, (ii) some manipulation of X would result in changes in the probability distribution of Y, no matter whether any variable in $\boldsymbol{V} \setminus \{X, Y\}$ were manipulated. If X is a direct cause of Y relative to $\boldsymbol{V}$, we say that there is a causal chain from X to Y. X is an indirect cause of Y with respect to $\boldsymbol{V}$ if there is a causal chain from X to Y of length greater than 2 (Pearl, 2009; Spirtes et al., 2000; Neapolitan, 2003; Glymour and Cooper, 1999).

We define causal Markov condition and causal Bayesian network by using the original definitions with the additional semantics that if there is an edge $A \rightarrow B$ in $\mathbb{G}$ then A directly causes B (for all A and $B \in \mathbb{V}$) (Spirtes et al., 2000).

Definition of local causal pathway: A local causal pathway of a target variable T is the set of its parents (direct causes) and children (direct effects) of T in the data-generative directed graph $\mathbb{G} =< \boldsymbol{V}, \mathbb{E} >$.

Definition of passenger: A passenger is a correlate of a target variable T and is neither a cause nor an effect of T.

Definition of local causal sufficiency: The variable set $\boldsymbol{V'}$ satisfies the local causal sufficiency condition if and only if it contains every common cause of all variables adjacent with a target variable T in the data-generative directed graph $\mathbb{G} =< \boldsymbol{V}, \mathbb{E} >$.

Next we provide several definitions of the faithfulness condition. This condition is essential for causal discovery from data.

Definition of graph faithfulness: If all and only the conditional independence relations that are true in $\mathbb{P}$ defined over variables $\boldsymbol{V}$ are entailed by the global Markov condition applied to a directed graph $\mathbb{G} =< \boldsymbol{V}, \mathbb{E} >$, then $\mathbb{P}$ and $\mathbb{G}$ are graph faithful to one another.

A relaxed version of graph faithfulness is given in the following definition:

Definition of adjacency faithfulness: Given a directed graph $\mathbb{G} =< \boldsymbol{V}, \mathbb{E} >$ and a joint

probability distribution P defined over variables $\boldsymbol{V}$, $\mathbb{P}$ and $\mathbb{G}$ are adjacency faithful to one another if every adjacency relation between X and Y in $\mathbb{G}$ implies that X and Y are conditionally dependent given every subset of $\boldsymbol{V} \setminus \{X, Y\}$ in $\mathbb{P}$ (Ramsey et al., 2006).

The adjacency faithfulness condition can be relaxed to focus on the specific target variable of interest:

Definition of local adjacency faithfulness: Given a directed graph $\mathbb{G} =< \boldsymbol{V}, \mathbb{E} >$ and a joint probability distribution $\mathbb{P}$ defined over variables $\boldsymbol{V}$, $\mathbb{P}$ and $\mathbb{G}$ are locally adjacency faithful with respect to T if every adjacency relation between T and X in $\mathbb{G}$ implies that T and X are conditionally dependent given any subset of $\boldsymbol{V} \setminus \{T, X\}$ in $\mathbb{P}$ (Statnikov et al., 2013).

It is known that some violations of the adjacency faithfulness condition can be attributed to violations of the intersection property of probability distributions (Pearl, 1997; Statnikov et al., 2013). This leads to distributions with variables that contain equivalent information (Statnikov et al., 2013; Lemeire, 2007). Such violations of the adjacency faithfulness condition constitute the focus of the paper because they are abundant in real biological networks, such as transcriptional gene regulatory networks (Statnikov et al., 2013; Statnikov and Aliferis, 2010; Dougherty and Brun, 2006), which are commonly investigated in computational causal discovery. For completeness, we also note that other violations of faithfulness exist in real biological networks and other real-life distributions, e.g. Simpsons paradox (Spirtes et al., 2000). While the latter violations may be equally important and not infrequent, they require a principally different treatment and often discovery techniques to address them are yet to be discovered; therefore we focus here only on violations due to information equivalencies.

Definition of target information equivalency: Two subsets of variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ from $\boldsymbol{V}$ are target information equivalent with respect to a variable T iff the following conditions hold $T \not\perp \boldsymbol{X}$, $T \not\perp \boldsymbol{Y}$, $T \perp \boldsymbol{X}|\boldsymbol{Y}$, and $T \perp \boldsymbol{X}|\boldsymbol{Y}$ (Lemeire, 2007).

For example, consider a joint probability distribution $\mathbb{P}$ described by a causal Bayesian network with graph $A \rightarrow B \rightarrow T$ where A, B, and T are binary random variables that take values $\{0, 1\}$. Given the local Markov condition, the joint probability distribution can be defined as follows: $P(A = 0) = 0.3, P(B = 0|A = 1) = 1.0, P(B = 1|A = 0) = 1.0, P(T = 0|B = 1) = 0.2, P(T = 0|B = 0) = 0.4$. It follows that A and B contain equivalent information about T and adjacency faithfulness is violated because $T \perp \boldsymbol{B}|\boldsymbol{A}$.

While the above example showed information equivalencies resulting from deterministic relations, information equivalencies follow from a broader class of distributions with both deterministic and non-deterministic information equivalencies (e.g., see Figure 1 in Statnikov et al. (2013)).

Finally, we provide a definition of a near-faithfulness condition, which is going to be one of the sufficient assumptions for the novel causal discovery algorithms described in this work.

Definition of target information equivalency (TIE) near-faithfulness: A joint probability distribution $\mathbb{P}$ and a directed graph $\mathbb{G} =< \boldsymbol{V}, \mathbb{E} >$ are target information equivalency (TIE) near-faithful to one another if all violations of faithfulness can be attributed only to presence of target information equivalency relations in $\mathbb{P}$.

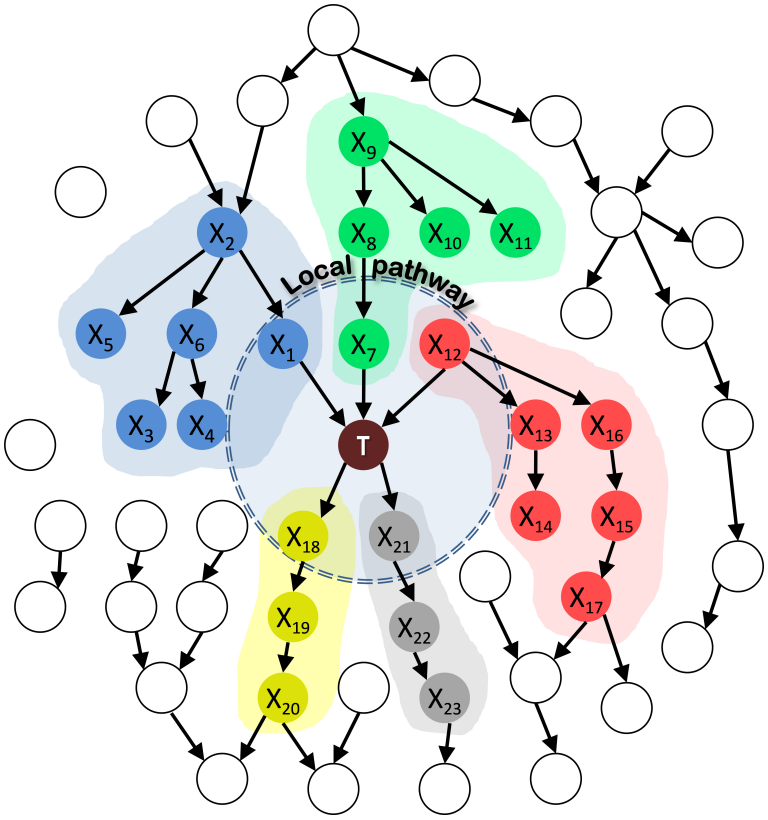

Figure 1: Graphical representation of an example TIE near-faithful causal network around a target variable T. The target variable T is shown in the middle of the network. Variables that are shown with the same color contain equivalent information about T. Variables in the local causal pathway of T are X_1, X_7, X_{12}, X_{18}, and X_{21}. Local causal discovery techniques that assume faithfulness (e.g., GLL-PC) will output one variable of each colored group. TIE* will output all subsets of the union of colored variables such that each subset has one variable from each colored group. No existing method will precisely determine the correct set $\{X_1, X_7, X_{12}, X_{18}, X_{21}\}$.

2.2 Local Causal Pathway Discovery from Observational Data in Faithful and Target Information Equivalency (TIE) Near-faithful Distributions

Prior research has provided sound conditional independence-based algorithms (e.g., GLL-PC from the Generalized Local Learning (GLL) family) for discovery of local causal pathway members from observational data under the assumptions of graph faithfulness (or local adjacency faithfulness with causal Markov condition), local causal sufficiency, and correctness of statistical decisions about dependence and independence (Aliferis et al., 2010a,b). To be precise, these methods only output the set of direct causes and effects of the target variable, but *do not distinguish which members of the output set are direct causes and which ones are direct effects*. The latter task requires randomized experiments or determination of edge

orientation through edge-orienting algorithms, temporal order, domain knowledge, or other post-processing criteria.

When the distribution is TIE near-faithful but not faithful, GLL-PC and other local causal discovery methods that assume faithfulness may lead to both false positives and false negatives in their output. Furthermore, false positives may neither be causes nor effects of the target variable. Consider an example causal network in Figure 1, which represents a TIE near-faithful distribution. The local causal pathway of the target variable T consists of five variables: $\{X_1, X_7, X_{12}, X_{18}, X_{21}\}$. Variables that are shown with circles of the same color contain equivalent information about T. For example, since X_1 and X_6 contain equivalent information about T, the following relation holds: $T \perp X_1 | X_6$. Thus, for example, GLL-PC may erroneously eliminate X_1 from the output (false negative) and conclude that X_6 is a member of the local causal pathway of T (false positive). In this distribution, there are 1,620 sets of five variables (= 6 'blue' × 5 'green' × 6 'red' × 3 'grey' × 3 'yellow' variables) that contain equivalent information about T. Notice that while only one of these 1,620 five-variable sets constitutes a local causal pathway of T, each of these five-variable sets can be arbitrarily output by GLL-PC or another local causal discovery algorithm that requires the same assumptions for soundness as GLL-PC, e.g. algorithms from (Peña et al., 2007). We say in such cases that there is a *multiplicity of local causal pathways consistent with the data*.

To address causal discovery in TIE near-faithful distributions, we have recently introduced two sound and complete algorithms TIE* and iTIE* (Statnikov et al., 2013) (described in Appendix F). These algorithms utilize conditional independence tests and allow discovery of all possible local causal pathways consistent with the data. In the example in Figure 1, these algorithms would identify all equivalency relations and output all 1,620 five-variable sets that span over variables $X_1, ..., X_{23}$. To further identify direct causes and direct effects of T in the variables output by the algorithms (the union of all equivalent sets of variables), one would need to resort to randomized experiments both because of target information equivalency and statistical indistinguishability of direct causes and effects in the context of local learning.

Now consider the global network learning methods such as SGS, PC (Spirtes et al., 2000), IC (Pearl, 2009), MMHC (Tsamardinos et al., 2006a), and LGL (Aliferis et al., 2010b) or region-based learning methods such as PCD-by-PCD (Yin et al., 2008), that under graph faithfulness, causal sufficiency, and correctness of statistical decisions can identify not only adjacency relations but also some edge orientations. The graphs output by these methods will be in general incomplete with regards to orientation because multiple graphs belong to the same Markov equivalence class of graphs and thus cannot be distinguished with observational data alone (Spirtes et al., 2000).

3. Prior Methods and Variants

Because learning a global causal network (that spans all measured variables) is substantially harder than learning a local causal pathway for a target variable, global methods fail to scale as the local ones. In order to experimentally test prior methods in high dimensional settings, we also introduce local versions of those that do not affect their soundness or quality.

Overall, we have considered 58 existing methods/variants spanning three main algorithmic families: conditional independence constraint-based structure learning (He and Geng, 2008; Meganck et al., 2006), linear cyclic models (Hyttinen et al., 2010; Eberhardt et al., 2010; Hyttinen et al., 2012) and Bayesian search-and-score (Pe'er et al., 2001; Sachs et al., 2005). These methods were chosen because they (i) reflect the current state-of-the-art in causal discovery, (ii) make use of observational and experimental data to produce directed causal networks, and (iii) are likely to scale to data of high dimensionality, unlike early methods for active learning of causal networks such as (Tong and Koller, 2001; Murphy, 2001). We describe the core ideas of each algorithmic family along with various variants below.

3.1 Conditional Independence Constraint-based Structure Learning

This family includes the ALCBN (Meganck et al., 2006) method and the method due to He and Geng (He and Geng, 2008). The main idea of these approaches is to learn an undirected[1] or partially directed graph from observational data (which represents the Markov equivalence class of graphs consistent with observational data), and then perform experiments to orient undirected edges. Both methods use the PC algorithm (Spirtes et al., 2000) to obtain an undirected or partially directed graph from observational data. The methods then use some decision criterion to select a variable for experimentation/manipulation, with the goal of maximizing the number of edges that are oriented after the experiment. The ALCBN algorithm uses either the mini-max, maxi-min or Laplace decision criteria (Meganck et al., 2006), whereas the method of He and Geng uses either the maxi-min or maximum entropy criteria (He and Geng, 2008). Once the variable is selected and manipulated, they perform a statistical independence test between the manipulated variable and each of its unoriented adjacencies in the graph, using experimental data. Adjacent variables that are associated with the manipulated variable are deduced to be direct effects, and all other adjacencies are direct causes (Spirtes et al., 2000). The ALCBN method repeats this process until all edges in the graph are oriented. The method of He and Geng first partitions the graph into chain components which are only connected by directed edges and orients each of these components separately. In addition to original methods, we also explored variants of these methods that restrict experimentation to the local causal pathway around a variable of interest/target or the chain component containing the variable of interest/target. A detailed list of all employed 24 methods/variants from this family (denoted as ALCBN and HE-GENG, accordingly) is given in Table A1 in Appendix A.

3.2 Linear Cyclic Models

This family includes three methods based on linear cyclic models with latent variables (Hyttinen et al., 2010; Eberhardt et al., 2010; Hyttinen et al., 2012). The main idea of these

1. The original methods considered using PC to learn a partially directed graph from observational data and then using experiments to further orient edges. Since orientation of PC is by design affected by errors in the adjacency structure, we also included in this work variants of these methods that work from the undirected graph obtained by PC from observational data.

approaches is to assume that all relations between variables are linear and can therefore be represented by an effects matrix. Discovering the causal structure then amounts to finding the coefficients of the effects matrix, which can be obtained by manipulating variables and deriving linear constraints on the effects. Specifically, these constraints are combined into a linear system and solved to obtain the coefficients of the effects matrix. Optionally, assuming faithfulness enables the use of the PC algorithm (Spirtes et al., 2000) on the manipulated and possibly observational data to learn adjacencies between variables. Non-adjacent variables imply additional constraints on the effects matrix, which are added to the linear system. The adjacencies also define an optimal order of variables to manipulate, which can minimize the number of required experiments. The derivation of constraints on the effects matrix and solution of the resulting linear system can be performed using any of the methods proposed by the authors, which we denote as LLC1 (Eberhardt et al., 2010), LLC2 (Hyttinen et al., 2010) and LLC3 (Hyttinen et al., 2012) ("LLC" stands for "linear, latents, cyclic"). The resulting effects matrix requires further filtering to obtain edges in the output graph. We used several approaches recommended by the authors: (i) removing all edges whose coefficients are less than a small fixed threshold, (ii) estimating the null distribution of the coefficients by rerunning the algorithm many times on permuted data and keeping only edges whose coefficients are statistically significant, and (iii) rerunning the algorithm a number of times on data sampled with replacement and keeping only edges whose mean coefficients are higher than their standard deviation. While the LLC method uses data for all variables in the network in order to estimate the effects matrix and produce the resulting causal graph, we limited the experiments only to variables with univariate association with the target (these methods have names beginning with "LLC"). In addition, we also experimented by limiting input data only for variables with univariate association with the target (these methods have names beginning with "UNIV-LLC"). A detailed list of all employed 32 methods/variants from this family is given in Table A2 in Appendix A.

3.3 Bayesian Search-and-score

This family includes the Biolearn method (Pe'er et al., 2001; Sachs et al., 2005). The main idea of this method is to define a space of candidate models, along with a scoring function that measures how well each model fits that data. Specifically, the score evaluates the posterior probability of a graph given the data. If given only observational data, graphs with the same undirected graph structure and unshielded colliders will have the same score (Neapolitan, 2003), and thus one can learn at best an equivalence class of graphs. Given experimental data, a score for each directed graph can be constructed by using the fact that the score decomposes into the local contributions of each variable. For each variable, only samples from experimental datasets where the variable was not manipulated were used, and the contributions of each variable were combined into a global score. This method can yield different scores for different orientations of the same graph structure, and thus can be used to evaluate how well directed graphs fit the combination of observed and manipulated data. Computing scores for all possible directed graphs is exponential in the number of variables, and thus it is usually not feasible to find the graph with the absolute highest score. Therefore, heuristics such as Greedy Hill-Climbing are used to limit the search space to a feasible

number. This method starts with an initial graph structure (such as the empty graph) and computes the score for closely related graphs obtained by adding, removing or reversing different edges. It selects the graph with the highest score, and repeats the procedure until it has found a local maximum. The entire process is repeated many times (e.g., 500), and the final model consists of all the edges present in a significant portion (85%) of the graphs. We used two variants of this method: one with the Normal Gamma scoring function (denoted as BIOLEARN.NG) and another one with the BDE scoring function (denoted as BIOLEARN.BDE).

4. New Methods

Below we provide new algorithms for local causal discovery. These algorithms rely on observational data for identifying members of the local causal pathway of a target variable; *however all orientation decisions are based on experimental data exclusively.* While prior research has provided theoretically sound approaches for orienting edges from observational data (e.g., V-structure based orientation in PC algorithm (Spirtes et al., 2000)), the empirical accuracy of these methods is affected by errors in constructing undirected skeleton and violations of faithfulness. We provide in Appendix D and Table D1 an empirical comparison of orientation approaches that concludes that significantly higher quality of orientation can be achieved from experimental data.

4.1 Algorithm ODLP*

In order to facilitate comprehension of the general methodology, we first address the problem of local causal pathway discovery in faithful distributions. The algorithm ODLP* is shown in Figure 2.

Theoretical analysis of the algorithm correctness: ODLP* is sound and complete under the sufficient assumptions of (i) local adjacency faithfulness; (ii) causal Markov condition; (iii) local causal sufficiency; (iv) acyclicity of the data-generative graph; and (v) correctness of statistical decisions. The proof of correctness relies on a previously established theoretical result showing that GLL-PC algorithms can identify members of the local causal pathway (direct causes and direct effects of the target variable) from observational data under the above stated assumptions (Aliferis et al., 2010a,b). In principle ODLP* can call another sound and complete algorithm for identification of local causal pathway members in step 1. Notice however, that algorithms for identification of local causal pathway members (such as GLL-PC) do not differentiate between direct causes and direct effects in the local causal pathway, and in general this task has to be accomplished with additional experimental data, as outlined in steps 2 and 3 of ODLP*.

Trace of the ODLP* algorithm: Consider running the ODLP* algorithm on observational data generated from the causal graph shown in Figure 2. We aim to identify the local causal pathway of the target variable T. In step 1 of ODLP*, GLL-PC will identify that variables X_1, X_2, X_3, X_4, X_5 belong to the local causal pathway of T, however would not define causal role of any of these variables. If it is possible to manipulate T, we would do so (step 2.a) and reveal that X_4 and X_5 change due to manipulation of T, and thus are direct effects of T (step 2.b); the remaining variables $X_1, X_2, and X_3$ thus have to be direct

Algorithm ***ODLP****

- *Input*:
  - Observational data D^O, including a target variable *T*;
  - Experimental protocols/methods to manipulate one variable at a time and generate experimental data D^E that quantifies response of the system to the manipulation.
- *Output*: Local causal pathway of *T*.

1. Apply *GLL-PC* or another sound and complete method to the observational data D^O to identify the set of variables ***V*** that are members of the local causal pathway of *T*.
2. If it is possible to manipulate *T*,
   a. Manipulate *T* and obtain experimental data D^E.
   b. Mark all variables in ***V*** that change in D^E due to manipulation of *T* as "*direct effects*" and mark remaining variables in ***V*** as "*direct causes*".
3. Else
   a. Manipulate a variable *X* in ***V*** to obtain experimental data D^E.
   b. If *T* changes in D^E due to manipulation of *X*, mark *X* as a "*direct cause*"; otherwise mark *X* as a "*direct effect*".
   c. Repeat steps 3.a and 3.b for all variables in ***V***.
4. Return the local causal pathway of *T*.

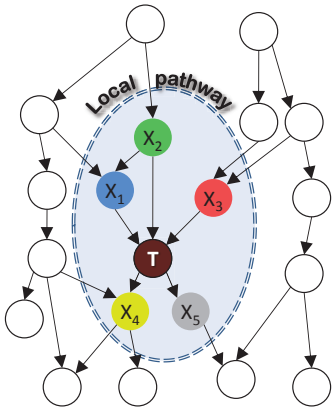

Figure 2: Pseudo-code of the ODLP* algorithm for faithful distributions. Left: Pseudo-code of the algorithm. Right: Graphical representation of an example causal network around a target variable T. Variables are shown with white circles, and edges represent direct causal influences. Variables in the local causal pathway of T are X_1, X_2, X_3, X_4, and X_5.

causes of T (step 2.b). On the other hand, if T cannot be manipulated, we can manipulate X_1 (step 3.a) and observe that T changes due to manipulation of X_1 (step 3.b); therefore X_1 is a direct cause of T (step 3.b). If we consider manipulating X_4 (step 3.a), we would observe that T does not change due to manipulation of X_4 (step 3.b); therefore X_4 is a direct effect of T (step 3.b). When steps 3.a and 3.b are applied to other variables in the local causal pathway, we will also find two additional direct causes of T (X_2 and X_3) and one additional direct effect (X_5) of T.

Analysis of the algorithm's experimental strategy and its efficiency: The experimental strategy of ODLP* is efficient because it relies only on single-variable manipulation experiments that are expected to generate a small number of samples in order to assess univariate association of the manipulated variable with all other variables. Furthermore, the algorithm tries to minimize the number of single-variable manipulation experiments and will conduct only one experiment if T can be manipulated (step 2.a). If it is not possible to manipulate T (e.g., T is a disease in humans), it will conduct the same number of experiments as the number of variables in the output of GLL-PC (set $\boldsymbol{V}$). In the most general case, it is impossible to further reduce this number of experiments because every variable in $\boldsymbol{V}$ can potentially be a direct cause of T and has to be confirmed by an experiment. We

note that situations exist that can lead to additional savings in experiments (e.g., when X, a direct effect of T, is causing Y, another direct effect of T, then manipulation of X would also reveal that Y is an effect of T and save an experiment) and we do check for them in the algorithm implementation, although they are not described in the algorithm pseudo-code in order to help understanding of its basic principles. Finally, it is also worthwhile to point out that the ODLP* algorithm can incorporate background knowledge both during the stage of learning the local causal pathway members (step 1) and when determining the causal role of the involved variables (steps 2 and 3), which can potentially lead to further reducing the number of required manipulation experiments.

We note that ODLP* does not represent a radical departure over previously known algorithms (it is a modest extension of preexisting ideas), however it is essential to conceptually describe the much more complex and generally applicable algorithm ODLP.

4.2 Algorithm ODLP

A more general algorithm ODLP shown in Figure 3 addresses the problem of local causal pathway discovery in TIE near-faithful distributions. This algorithm is specifically designed for situations when the target variable T can be manipulated.

Theoretical analysis of the algorithm correctness: The following theorem states correctness of ODLP; the proof is given in Appendix G. Specifically, the theorem shows that under certain assumptions, ODLP will return all and only members of the true local causal pathway of a target variable T.

Theorem 1 *ODLP is sound under the following sufficient assumptions: (i) TIE near-faithfulness (as a relaxation of local adjacency faithfulness to allow for target information equivalency relations); (ii) causal Markov condition; (iii) local causal sufficiency; (iv) acyclicity of the data-generative graph; and (v) correctness of statistical decisions.*

In non-technical terms, the first two assumptions mean that with the exception of empirical target information equivalency relations, there is a direct correspondence between the data and a directed acyclic data-generative graph in terms of statistical relations (specifically, there is an edge between two variables if and only if they have association in the data conditioned on every subset of other variables). The third assumption means that every common cause of two or more measured variables is also measured in the dataset. If this assumption is violated, direct causation cannot be discovered by using observational data together with experiments limited to single-variable manipulations, as demonstrated in Figure 1 of (Eberhardt et al., 2010). The fourth assumption means that there are no feedback cycles in the graph. The fifth assumption means that determination of variable (in)dependency in the population from the available data sample is correct.

Trace of the ODLP algorithm: Consider running ODLP on data generated from the network in Figure 1. The algorithm aims to identify the local causal pathway of the target variable T. In step 1, TIE* will find 1,620 local causal pathways of T consistent with the data. The union of these data-consistent pathways (set $\boldsymbol{V}$) will be variables $X_1, ..., X_{23}$ (step 2). Then in step 3, ODLP will form five equivalence clusters of variables based on

Algorithm ***ODLP***

- *Input*:
  - Observational data D^O, including a target variable *T*;
  - Experimental protocols/methods to manipulate one variable at a time and generate experimental data D^E that quantifies response of the system to the manipulation.
- *Output*: Local causal pathway of *T*.

1. Apply TIE* or iTIE* to the observational data D^O to identify all local causal pathways of *T* consistent with the data.
2. ***V*** ← Union of all variables that participate in local causal pathways of *T* consistent with the data (*this is a draft of the local causal pathway*).
3. Form equivalence clusters over variables in ***V*** such that each equivalence cluster contains variables that have equivalent information about T (*this can be accomplished directly from the output or the operation of TIE* or iTIE**).

Identify effects of T

4. Manipulate *T* and obtain experimental data D^E.
5. Mark all variables in ***V*** that change in D^E due to manipulation of T as "*effects*".

Identify direct and other causes of T

6. Repeat
   a. If there is an equivalence cluster that contains a single unmarked variable *X* and all marked variables in this cluster (if any) are only passengers and/or effects, then mark *X* as a "*direct cause*" and go to step 6.
   b. Select (according to some heuristic function or at random) an unmarked variable *X* from an equivalence cluster.
   c. Manipulate *X* and obtain experimental data D^E.
   d. If *T* does not change in D^E due to manipulation of *X*, mark *X* as a "*passenger*" and mark all other non-effect variables that change in D^E due to manipulation of *X* as "*passengers*"; otherwise mark *X* as a "*cause*".
7. Until there are no equivalence clusters with unmarked variables.
8. For every cause *X*, mark *X* as a "*direct cause*" if there exist no other cause in the same equivalence cluster that changes due to manipulation of *X*; otherwise mark *X* as an "*other cause*".

Identify direct effects of T

9. Repeat
   a. If there is an equivalence cluster that contains a single effect variable *X* which has neither been marked as "other effect" nor as "direct effect" and other effect variables in this cluster (if any) are only other effects, then mark *X* as a "*direct effect*" and to go step 9.
   b. Select (according to some heuristic function or at random) an effect variable *X* that has neither been marked as "*other effect*" nor as "*direct effect*".
   c. Manipulate *X* and obtain experimental data D^E.
   d. Mark all effect variables that change in D^E due to manipulation of *X* and belong to the same equivalence cluster as "other *effects*".
10. Until all effect variables are either marked as "*other effects*" or *"direct effects"*.
11. Return the local causal pathway of *T*, i.e. only direct causes and direct effects of *T*.

Figure 3: Pseudo-code of the ODLP algorithm for TIE near-faithful distributions. Notice that even though the algorithm outputs the local causal pathway of T, during its execution it also discovers the causal role of other variables that will provide additional clues about underlying mechanisms. Steps 4, 6.c, 9.c provide an interface of the algorithm with the external world through experiments that are conducted by an experimentalist, and are shown with dark grey highlighting.

information that they provide about the T (the clustering will coincide with the color of highlighting of variables in Figure 1). In steps 4 and 5 the algorithm will manipulate T and identify its effects $X_{18}, ..., X_{23}$. Then the algorithm will proceed to identification of direct/other causes ("other causes" are defined as causes that are not identified as direct causes, they could be indirect causes or both direct and indirect causes at the same time) of T in the candidate set of variables $X_1, ..., X_{17}$. There is no equivalence cluster that satisfies criterion of step 6.a, so ODLP will proceed to step 6.b and select a variable for manipulation (for example without loss of generality, X_6) in step 6.c. The algorithm will then identify that X_6 is a passenger and so are X_3 and X_4 (step 6.d). Steps 6.a-6.d will be repeated until the causal role of every non-effect variable is deciphered. Next, the algorithm will conclude that X_1, X_7, and X_12 are direct causes of T and X_2, X_8, and X_9 are other causes (step 8). Then ODLP will proceed to the identification of direct effects and other effects of T in the set of effects $(X_{18}, ..., X_{23})$. Similarly, "other effects" are effects that are not identified as direct effects, they could be indirect effects or both direct and indirect effects at the same time. There is no equivalence cluster that satisfies criterion of step 9.a, so the algorithm will proceed to step 9.b and select a variable for manipulation (for example without loss of generality, X_19) in step 9.c. In step 9.d ODLP will identify that X_20 is other effect of T and repeat iterations until all effects are either marked as "other effects" (X_{19}, X_{20}, X_{22}, and X_{23}) or "direct effects" (X_{18} and X_{21}). Thus the local causal pathway of T (that consists of direct causes X_1, X_7, X_{12} and direct effects X_{18}, X_{21}) has been identified correctly.

Analysis of the algorithm's experimental strategy and its efficiency: The strategy of ODLP relies on single-variable manipulation experiments and usually requires a small number of samples from each experiment to assess univariate associations of the manipulated variable with other variables. In general, the number of experiments necessary for identification of the local causal pathway would be manageable for experimentalists, although it varies and depends on the structure of the local causal pathway. The number of experiments for the best and worst case is 1 and $|\boldsymbol{V}| + 1$, respectively, where the set V is the union of all variables that participate in local causal pathways of T consistent with the data. In any case, the number of experiments would be manageable because $\boldsymbol{V}$ in real distributions, even in high-throughput datasets, is typically between 10 and 200 variables, as we have observed by running TIE* in > 30 datasets (Statnikov et al., 2013; Statnikov and Aliferis, 2010).

An important principle behind minimization of experiments is to first manipulate in step 6.c passengers of T that are causing many other passengers of T. For example, manipulation of X_6 in Figure 1 would lead to changes in X_3, X_4 but not in T. Therefore, X_3, X_4, and X_6 are not causes of T. The algorithm can also infer from manipulation of T that X_3, X_4, and X_6 do not change and thus are not effects of T. Therefore, they are passengers. The algorithm determined the causal role of X_3, X_4, and X_6 by manipulating only one of these variables. However, the graphical structure is not known when the algorithm performs experiments, and thus it has to resort to heuristics to manipulate first variables that are likely to yield savings in experiments. The algorithm uses a partial network-based heuristic that chooses a variable that has the highest topological order relative to T. The topological order can be established from constraints learned from experimental data, as well as from domain knowledge, temporal order information, computational edge orientation algorithms based on observational data, and other sources. In addition to the above

Network Name	Description	Construction Methodology	Num. Variable	Num. Edges	Num. Samples in Obs. Data	Num. Samples in Exp. Data	Reference
REGED	Resimulated transcriptional gene regulatory network from gene expression data of human lung cancer patients. Variables represent expression levels of genes, and target variable represents lung cancer subtype.	Used a publicly available microarray gene expression dataset to learn a network structure of transcriptional interactions. Parameterized the network using non-linear regression.	1,000	1,148	500	100	Guyon et al. (2008)
ECOLI	Resimulated transcriptional gene regulatory network based on the current knowledge of regulation in E.Coli. Variables represent expression levels of genes.	Used large-scale experimental data to infer the network structure, and then used principles of thermodynamics and molecular kinetics to parameterize the network.	1,565	3,648	1000	200	Marbach et al. (2009); Schaffter et al. (2011)
YEAST	Resimulated transcriptional gene regulatory network based on the current knowledge of regulation in S. Cerevisiae. Variables represent expression levels of genes.	Same as above	4,441	12,873	1000	200	Marbach et al. (2009); Schaffter et al. (2011)
P1000	Artificially simulated network, where the target information equivalency phenomenon is present in the local causal pathway of the target variable. As a result, the target variable has multiple 1,620 data-consistent local causal pathways.	Manually generated graph of the network and parameterized using Gaussian distribution.	1,000	51	1000	20	Novel
P1M	Large-scale version of P1000 network with 1,000,000 variables.	Tiled with P1000 as the basic component with inter-tile connections.	1,000,000	81,969	1000	20	Novel

Table 1: Description of networks and data used in empirical experiments.

heuristic, other heuristic functions can be used. We refer interested readers to Appendix H for more detailed examples explaining ODLP's experimental strategy and its efficiency. Similarly, prioritizing manipulation of direct effects in step 9.c allows saving experiments by avoiding manipulation of indirect effects. Finally, it is also worthwhile to point out that the ODLP algorithm can incorporate background knowledge both during the stage of drafting the local causal pathway (step 1) and when determining the causal role of variables (steps 4-10), which can potentially lead to further reducing the number of required experiments.

We also note that in settings when the assumptions of the algorithm are violated and TIE* outputs false positives, one may choose not to perform step 6.a and always manipulate a single unmarked variable in the equivalence cluster to ensure that it is indeed the cause. Otherwise, a false positive variable (e.g., passenger in the equivalence cluster consisting of one variable) will be erroneously classified as "direct cause" in step 6.a. However, when the sufficient assumptions of the algorithm hold, step 6.a does not lead to errors and provides savings in the number of experiments. Similarly, step 9.a can be omitted which leads to improving robustness in handling false positives but decreasing experimental efficiency.

5. Empirical Experiments

This section describes the data used in the empirical experiments, implementation of different causal discovery algorithms, performance metric and statistical comparison methods, and results of the empirical experiments.

5.1 Networks and Data

The networks and data used in empirical experiments are summarized in Table 1. The REGED, ECOLI, and YEAST networks produce resimulated gene expression data that resembles data from real transcriptional gene regulatory networks. Since these networks have been previously published (Guyon et al., 2008; Marbach et al., 2009; Schaffter et al., 2011), we do not describe them in detail here. We will only mention that variables in ECOLI and YEAST networks (genes) typically have very few (0-2) direct causes (direct upstream regulators), and some variables (transcription factor genes) have a large number of direct effects (direct downstream targets) that can even reach low hundreds. This is consistent with the principles of transcriptional regulation. The P1000 network is intended i) to resemble data from real transcriptional gene regulatory networks which are generally very sparse and ii) to demonstrate the effect of multiplicity of causal pathways consistent with the data, a phenomenon which is omni-present in real biological networks (Statnikov et al., 2013; Statnikov and Aliferis, 2010; Dougherty and Brun, 2006). This network was obtained by parameterizing the local causal pathway structure shown in Figure 1 using linear Gaussian distribution and adding unconnected Gaussian variables, so that the total number of variables is 1,000. The parameterization of the network is provided in Table B1 in Appendix B. The P1M network was obtained by "tiling" the P1000 network one thousand times. The structural and probabilistic properties of individual tiles were similar to that of P1000, so that the distribution of P1M network resembles the distribution of the P1000 network. More specifically, one thousand copies (i.e. tiles) of the P1000 network were generated, each copy with the set of vertices $\boldsymbol{V_i}$ and the set of edges $\boldsymbol{E_i}$ that are copies of

$\boldsymbol{V_{P1000}}$ and $\boldsymbol{E_{P1000}}$, the vertex set and the edge set of the original P1000 network. Then, the tiles were interconnected with edges between $\boldsymbol{V_i}$ to $\boldsymbol{V_{i+1}}$. The vertices that received inter-tile edges were re-parameterized to preserve their marginal distribution, following the approach described in (Tsamardinos et al., 2006b). See Figure B1 in Appendix B for visualization of the fragment of the connected components of the P1M network.

We generated 1,000 samples for the observational datasets for all networks, except for REGED because this network has been previously used with 500 samples in the international challenge on Causation and Prediction (Guyon et al., 2008). Prior to running experiments, we generated experimental datasets by manipulating each variable in every network. The sample size for experimental datasets was minimized for each network over $\{20, 100, 200\}$ in order to be realistic and at least have sufficient power to estimate univariate associations of manipulated variables with other variables in the network. As a result, we used 100 samples in REGED, 200 in ECOLI and YEAST, and 20 in P1000 and P1M networks for experimental datasets. All generated experimental datasets were saved in a working database. Causal pathway discovery methods could query this database to obtain an experimental dataset where the variable of interest was manipulated. The decoupling of the two most time consuming components of experiments, simulation of experimental data and running causal discovery algorithms, allowed us to setup a robust algorithm evaluation environment (Figure 4). All data for the simulations is available on the manuscript supplemental website: `http://ccdlab.org/odlp.html`.

Since we are focusing here on discovery of local causal pathways, the next step is to select target variables of interest. The networks REGED, P1000, and P1M have designated target variables. However, there are no designated targets in YEAST and ECOLI networks. Therefore, we selected four variables from each network (the number of selected variables was limited by computational resources of the study) and designated them as targets. These four variables were selected randomly from the subset of transcription factors (that play key regulatory role in these networks) such that they represent local causal pathways of varying sizes for each network. This also allows assessing sensitivity of methods to the size of the local causal pathway. More details are given in Table 2.

5.2 Local Causal Pathway Discovery Methods and Implementations

In addition to ODLP, we evaluated 58 existing methods/variants for active learning of causal networks that are described in Section 3. ODLP and conditional independence constraint-based structure learning methods ALCBN and HE-GENG were implemented in Matlab and used the implementation of Fisher's Z test of conditional independence from the Causal Explorer library (Statnikov et al., 2010). ODLP was run using the iTIE* algorithm to find all data-consistent local causal pathways, parameter max-k (denoting maximum cardinality of the conditional test) set to 3, and 0.05 alpha for assessing dependence/independence. ALCBN and HE-GENG used the implementation of the PC algorithm from the Causal Explorer library (Statnikov et al., 2010) and were run with maximum cardinality of conditional tests set to 2 and 0.05 alpha for assessing dependence/independence. We tried to run the algorithms with larger cardinality of conditional tests, but it was not computationally feasible because PC did not terminate in most cases in less than one month of single-core

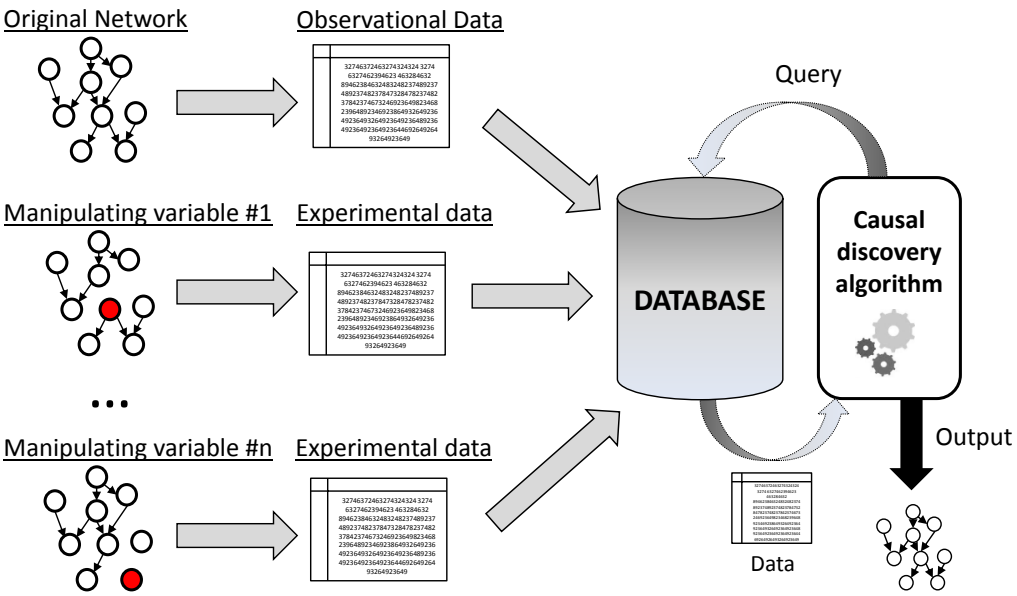

Figure 4: Data generation process/experimental setup. The depicted experimental setup allowed to decouple data generation and running of algorithms, therefore providing a robust algorithm evaluation environment.

time. We used the original authors' R implementations of methods based on linear cyclic models (obtained directly from the authors) and improved their efficiency in Matlab, e.g. to solve very large-dimensional sparse linear systems that cannot be solved easily in R due to current memory restrictions. Finally, we used the original authors' software implementation of the Bayesian search-and-score method. Table E1 in Appendix E provides information and location of publicly available software implementations of the above discovery methods.

5.3 Performance Metrics and Statistical Comparison

Assessment of the performance of algorithms was based on the following metrics: (i) sensitivity, (ii) specificity, and (iii) number of required experiments. Sensitivity and specificity are metrics to assess the accuracy of structure learning, and they were computed for the task of discovery of all direct causes and all direct effects of the target variable T. Sensitivity and specificity range from 0 to 1 (or 0% to 100%), with larger values denoting better performance. We also combined sensitivity and specificity into a single metric, the Euclidean distance from the point with sensitivity and specificity equal to 1: $\frac{\sqrt{(1-sensitivity)^2+(1-specificity)^2}}{\sqrt{2}}$. The latter metric is referred to as "distance" in the manuscript and it ranges from 0 to 1 (or 0% to 100%), with larger values denoting worse performance. In addition to using the raw values for the number of experiments, we also normalized this metric by dividing it by the number of variables in the local causal path-

Network Name	Target Variable T	Num. Variables in the Local Causal Pathway of T	Num. Direct Causes of T	Num. Direct Effects of T
REGED	Adenocarcinoma vs. squamous lung cancer subtype.	15	2	13
ECOLI	Expression levels of gene agaR	8	0	8
	Expression levels of gene allR	10	0	10
	Expression levels of gene zur	6	0	6
	Expression levels of gene lexA	54	0	54
YEAST	Expression levels of gene YBL005W	30	1	29
	Expression levels of gene YFL044C	15	0	15
	Expression levels of gene YLR014C	31	0	31
	Expression levels of gene YKL112W	300	2	298
P1000	Artificial	5	3	2
P1M	Artificial	5	3	2

Table 2: Description of target variables chosen from each network and their local causal pathways. As mentioned in the manuscript, the small number of direct causes of the target variables in ECOLI and YEAST networks is representative of these two networks and principles of transcriptional regulation.

way of T or by the number of variables in the entire network. To test whether the differences in distance metric between the nominally best performing algorithm and other algorithms are non-random for a specific local causal pathway discovery task, we used a statistical permutation-based test adapted from (Menke and Martinez, 2004). We obtained a null distribution for each comparison task and computed the corresponding p-value. When the p-values are not statistically significant at 0.05 alpha level after adjusting for multiple comparisons using the methodology of (Benjamini and Hochberg, 1995; Benjamini and Yekutieli, 2001), the resulting algorithms are deemed to have statistically comparable distance with the algorithm with best distance value. We refer to such values of distance metric as 'optimal' for a specific local causal pathway discovery task relative to the tested methods.

5.4 Computing Resources and Execution of Experiments

To run the experiments, we used three high-performance computing (HPC) clusters available to us at the time of experiments. These HPC clusters included: the Asclepius cluster of the NYU Langone Medical Center, the Bowery cluster of the New York University main campus, and the BuTina cluster of the New York University Abu Dhabi campus in the United Arab Emirates. Asclepius had $\sim 1,000$ Intel x86 processing cores and 4TB of RAM distributed among the cluster's compute nodes. The Bowery cluster had $\sim 2,500$ cores and

9TB of RAM total among all the nodes. The BuTina cluster had $\sim 6,400$ latest Intel x86 processing cores with a total of 26TB of RAM.

In addition to distributing the tasks of running various causal pathway discovery algorithms for various networks/target variables among compute cores of the cluster, we also often divided the individual tasks (of running a single algorithm for a specific network/target) into many sub-tasks. For example, Biolearn requires running Greedy Hill-Climbing procedure 500 times, all of which can be run independently on individual cores. In many cases, the independent nature of the sub-tasks enabled linear speedup. In order to complete execution of experiments with available resources, we imposed three termination criteria: (i) 30 day single-core time limit for tasks that cannot be easily parallelized; (ii) 3,000 day multi-core time limit for tasks that can be further parallelized (spread over 100 cores); and (iii) 48 GB RAM. We used 500-700 cores at a time over 2.5 calendar years. We estimate that the final results reported here required 800 core-years of computation.

5.5 Results

The detailed results of experiments are provided in Table C1 (for REGED, P1000, and P1M networks), Table C2 (for ECOLI network), and Table C3 (for YEAST networks) in Appendix C. These tables provide values of sensitivity, specificity, distance, and number of experiments for each method and local causal pathway discovery task. As mentioned in the previous sub-section, in some cases experiments were terminated due to extensive computational resource requirements or, for Biolearn, failure of the original software implementation of the method. These cases are marked in the tables with special codes T1, T2, T3, T4, and the legend is given in Table C1.

Before reporting detailed analysis of the results, it is worth noting that the ODLP algorithm resulted in better performance than any other algorithm when applied to the P1000 dataset. This is partly due to the fact that TIE* and the ODLP algorithm specifically address local pathway multiplicity, which is present in P1000 dataset. On the other hand, many other algorithms rely on the PC algorithm, which assumes faithfulness.

Analysis based on the counts of successes/failures: In the following analyses, presented in Figures 5-8, we provide for each method the number of counts of successes/failures (according to various metrics) within 11 local causal discovery pathway tasks.

Figure 5 reports for each method the number of local causal pathway discovery tasks where a method either exceeded available computational resources or its original software implementation failed to run. ODLP is the only method that was able to run for all 11 local causal pathway discovery tasks. No other method was able to run for P1M network with 1,000,000 variables. However, within each algorithmic family except for Biolearn, there are methods that were able to run on the remaining 10 local causal pathway discovery tasks (represented by a failure number of 1). From ALCBN and HE-GENG families, these are mostly methods restricted to the local neighborhood of the target variable. From LLC family, these are methods that use only variables with significant univariate association with the target variable. This observation motivates the approach of using local methods for solving local causal pathway discovery problems. Also, for ALCBN and HE-GENG methods that discover the global network, the ones that use undirected PC skeleton (ALCBN.S. or

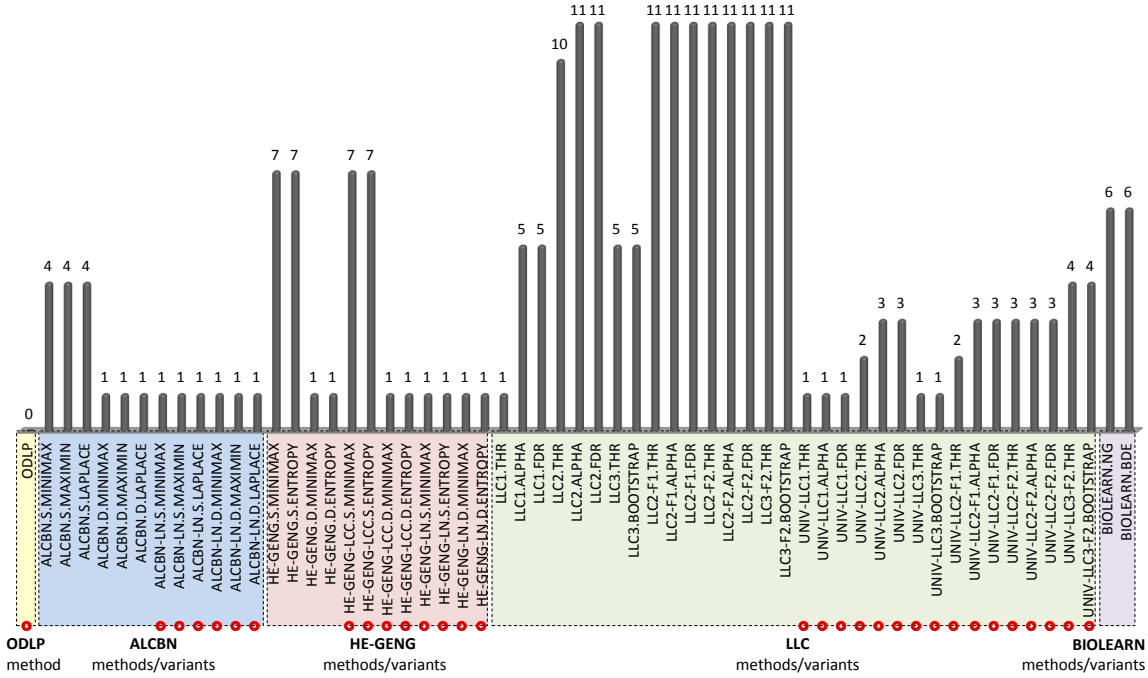

Figure 5: Number of local causal pathways where the algorithm was terminated/failed (out of 11 local causal pathways). Red circles denote methods designed for the discovery of local causal pathways. These include our modifications of the original global methods for local learning.

HE-GENG.S.*) fail more often in comparison to the ones that use partially directed global graph (ALCBN.D.* or HE-GENG.D.*). This is due to the fact that more computation is needed to determine which variable(s) to manipulate in the completely undirected graph, and our experiments have a restriction on computational time. It is also worthwhile to mention that the runtime of ODLP was under 10-15 minutes for all pathways, except for YEAST pathway for gene YKL112W where it took the algorithm one hour to run because the underlying local causal pathway was of large size (300 members). Other methods took orders of magnitude more computing time, e.g. it took ALCBN and HE-GENG of the order of 10 hours to obtain unoriented PC skeleton, and it took LLC of the order of several days to derive constraints on the effects matrix and combine them into a linear system. These run-time estimates are for the major computing components of the core methods, without bootstrapping/permutations. If the latter techniques are used, the run-time typically increases by more than two orders of magnitude due to a large number of independent runs of the core method.

Figure 6 reports for each method the number of local causal pathway discovery tasks where a method achieved optimal value of the distance metric (defined as a distance value that is not significantly different from the best distance achieved by all method examined,

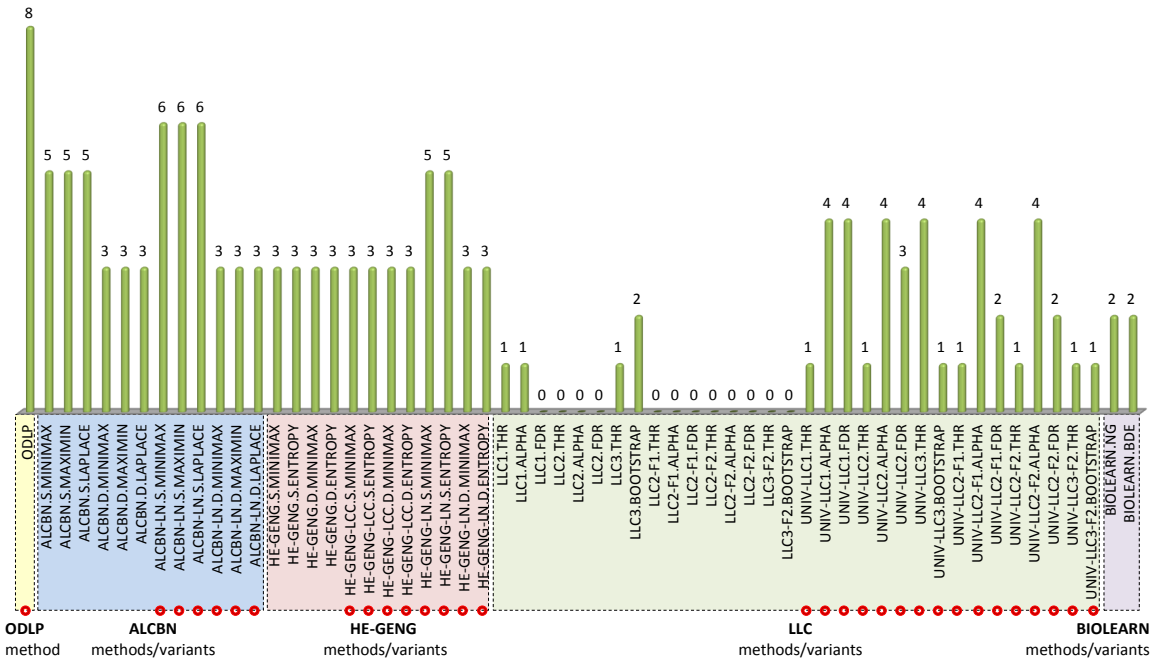

Figure 6: Number of local causal pathways discovered by an algorithm with optimal distance (out of 11 local causal pathways).Red circles denote methods designed for the discovery of local causal pathways. These include our modifications of the original global methods for local learning.

reflecting accuracy of structural discovery of the pathway). ODLP achieved optimal distance in eight out of 11 pathways, local versions of ALCBN based on unoriented PC skeleton achieved optimal distance in six pathways, local versions of HE-GENG based on unoriented PC skeleton and versions of ALCBN based on unoriented skeleton achieved optimal distance in five pathways, and some versions of LLC limited to variables univariately associated with the target achieved optimal distance in four pathways. Other methods achieved optimal distance in three or fewer pathways.

Figure 7 reports for each method the number of local causal pathway discovery tasks where a method achieved optimal values of the distance metric and did not perform more experiments than the number of members in the pathway. Figure 8 provides similar data but for the number of experiments limited by 10, which is commonly used in biological sciences for expensive experiments. In both analyses, ODLP and local versions of ALCBN based on the unoriented PC skeleton succeeded in six out of 11 pathways. Local versions of HE-GENG also based on the unoriented PC skeleton succeed in five (if the number of experiments is limited by the number of members in the pathway) or four (if the number of experiments is limited by 10) pathways. Two versions of LLC limited to variables univariately associated with the target succeeded in four pathways (if the number of experiments is limited by the number of members in the pathway). All other methods/variants succeeded in three or

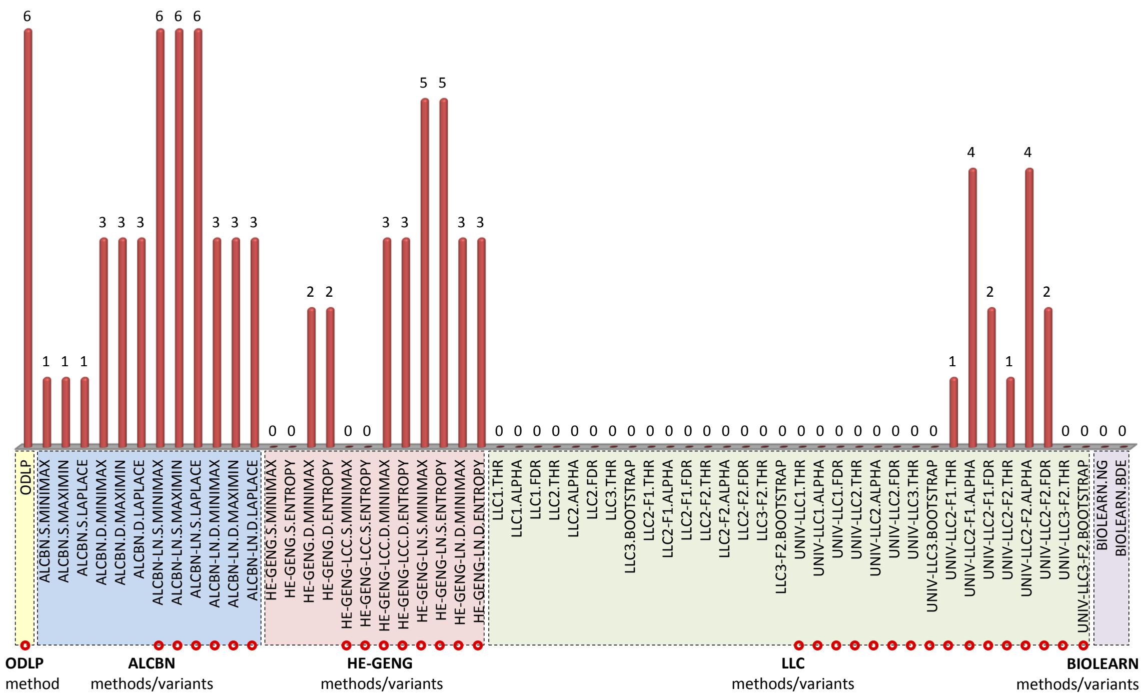

Figure 7: Number of local causal pathways discovered by an algorithm with optimal distance and with the same or fewer experiments than members of the pathway (out of 11 local causal pathways).Red circles denote methods designed for the discovery of local causal pathways. These include our modifications of the original global methods for local learning.

fewer pathways.

Table 3 uses data from Figures 5-8 for 58 methods/variants for active learning of causal networks to assess how the original global network learning methods (N = 28) perform relative to the methods modified specifically for local learning (N = 30). As can be seen, method variants modified for local learning fail in significantly fewer pathways, discover more pathways with optimal distance metric (reflecting structural discovery accuracy), and also achieve optimal distance metric with small number of experiments in more pathways than the original global learning methods/variants.

Analysis based on averages: The following analyses in Figures 9-11 visualize values of various metrics averaged over local causal pathway discovery tasks where all participating methods have completed and returned results (since we consider different number pathways from different networks, we first average results within each network and then over all networks). These analyses provide additional information compared to the counts of successes/failures because they also quantify the magnitude of successes/failures by reporting the average values. However, since only one method (ODLP) has completed on all 11 pathways, we have to a use a subset with 10 pathways (excluding P1M) and focus only

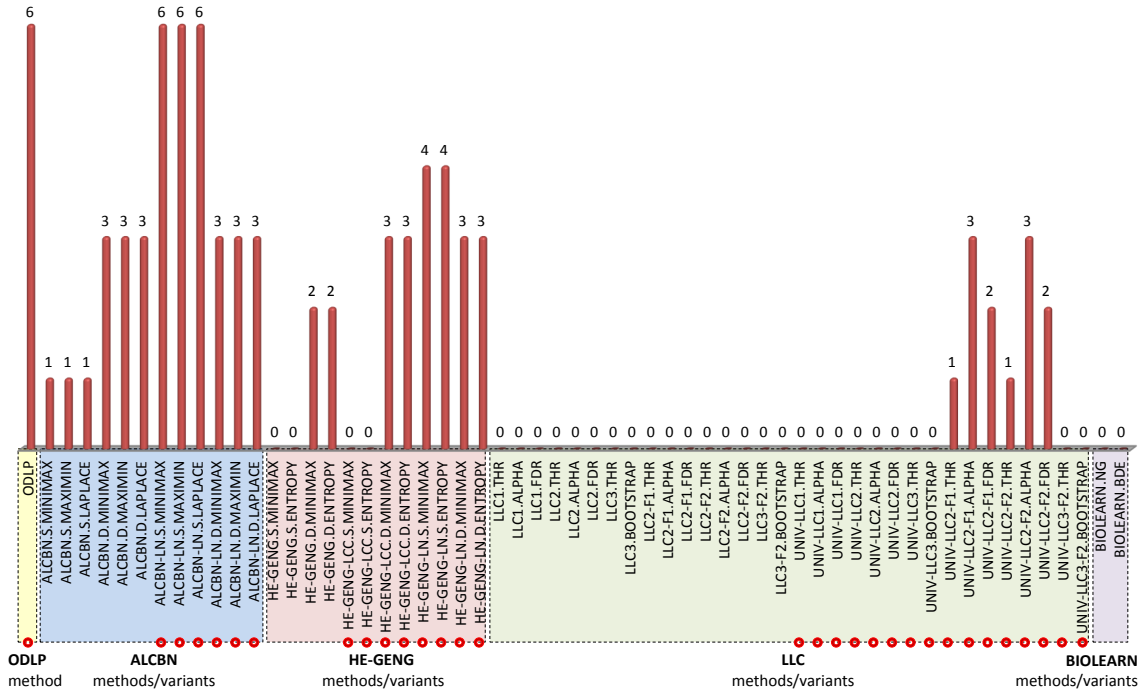

Figure 8: Number of local causal pathways discovered by an algorithm with optimal distance and with 10 or fewer experiments (out of 11 local causal pathways). Red circles denote methods designed for the discovery of local causal pathways. These include our modifications of the original global methods for local learning.

on 24 out of all 59 methods that have completed for all pathways in the considered subset.

Figure 9 shows a bull's eye plot for the distance metric and the number of experiments averaged over 10 local causal pathways. Location of the circles corresponds to values of the distance metric: the closer is circle to the center, the smaller (better) is the distance. The color of the circles corresponds to the number of experiments: the lighter is color, the more experiments are required. As can be seen, ODLP and a variant of LLC, UNIV-LLC3.THR, have the smallest average values of the distance metric, 9.6% and 12%, respectively. ODLP achieves this result with only 5 experiments, while the result of UNIV-LLC3.THR is based on 280 experiments. It is fair to note here that the ODLP method specifically optimizes the number of experiments, while UNIV-LLC3.THR uses experiments for all variables with significant univariate association with the target variable. An alternative and more detailed visualization of the data from Figure 9 is given in Figure 10 that shows a plot of distance versus number of experiments/number of variables in the network averaged over 10 local causal pathways.

Finally, Figure 11 shows a plot of sensitivity versus specificity averaged over 10 local causal pathways. A variant of LLC, UNIV-LLC3.THR, is the only method that has larger sensitivity than ODLP: sensitivity of ODLP and UNIV-LLC3.THR is 86.5% and 88.3%, respectively. However, this small 1.8% increase in sensitivity is accompanied by a significant

	Global Learning	Local Learning	P-value
Number of methods/variants	28	30	N.A.
Number of local causal pathways where the method was terminated/failed	Mean = 6.57 (St. dev. = 3.97)	Mean = 2.13 (St. dev. = 1.68)	6.33×10^{-7}
Number of local causal pathways discovered by a method with optimal distance (structural accuracy)	Mean = 1.61 (St. dev. = 1.73)	Mean = 3.10 (St. dev. = 1.56)	1.06×10^{-3}
Number of local causal pathways discovered by a method with optimal distance and with the same or fewer experiments than members of the pathway	Mean = 0.57 (St. dev. = 1.03)	Mean = 2.10 (St. dev. = 2.12)	1.09×10^{-3}
Number of local causal pathways discovered by a method with optimal distance and with 10 or fewer experiments	Mean = 0.57 (St. dev. = 1.03)	Mean = 1.97 (St. dev. = 1.99)	1.62×10^{-3}

Table 3: Comparison of performance of local and global learning methods/variants. P-values were obtained with a two-sample t-test. Statistical significance was assessed at 5% alpha level.

loss of specificity: specificity of ODLP is 99.97%, while specificity of UNIV-LLC3.THR is 90.4%. Finally, there are no methods that have larger specificity than ODLP.

6. Discussion

Methods for experimentally efficient and accurate discovery of local causal pathways from data can readily provide significant advances in many fields. For example, they can increase efficiency of drug discovery, facilitate development of socio-economic policies with desirable outcomes, or lead to successful marketing campaigns. Prior research has introduced several methods for active learning of the *entire/global* causal networks that utilize both observational and limited experimental data. The current study introduced new methods (termed ODLP) for discovery of local causal pathways around the target variable of interest using observational and experimental data, a topic not previously explored in the literature. Our new methods aim to minimize the number of experiments and also have substantially less restrictive theoretical assumptions for correctness compared to existing alternatives. An extensive empirical comparison of ODLP with 58 state-of-the-art methods/variants in high-dimensional datasets revealed that: (i) ODLP scales to datasets with 1,000,000 variables unlike comparator methods, which often fail to terminate within reasonable time even on datasets with of the order of 1,000 variables; (ii) ODLP achieves best local causal pathway discovery accuracy with minimal number of experiments compared to existing techniques

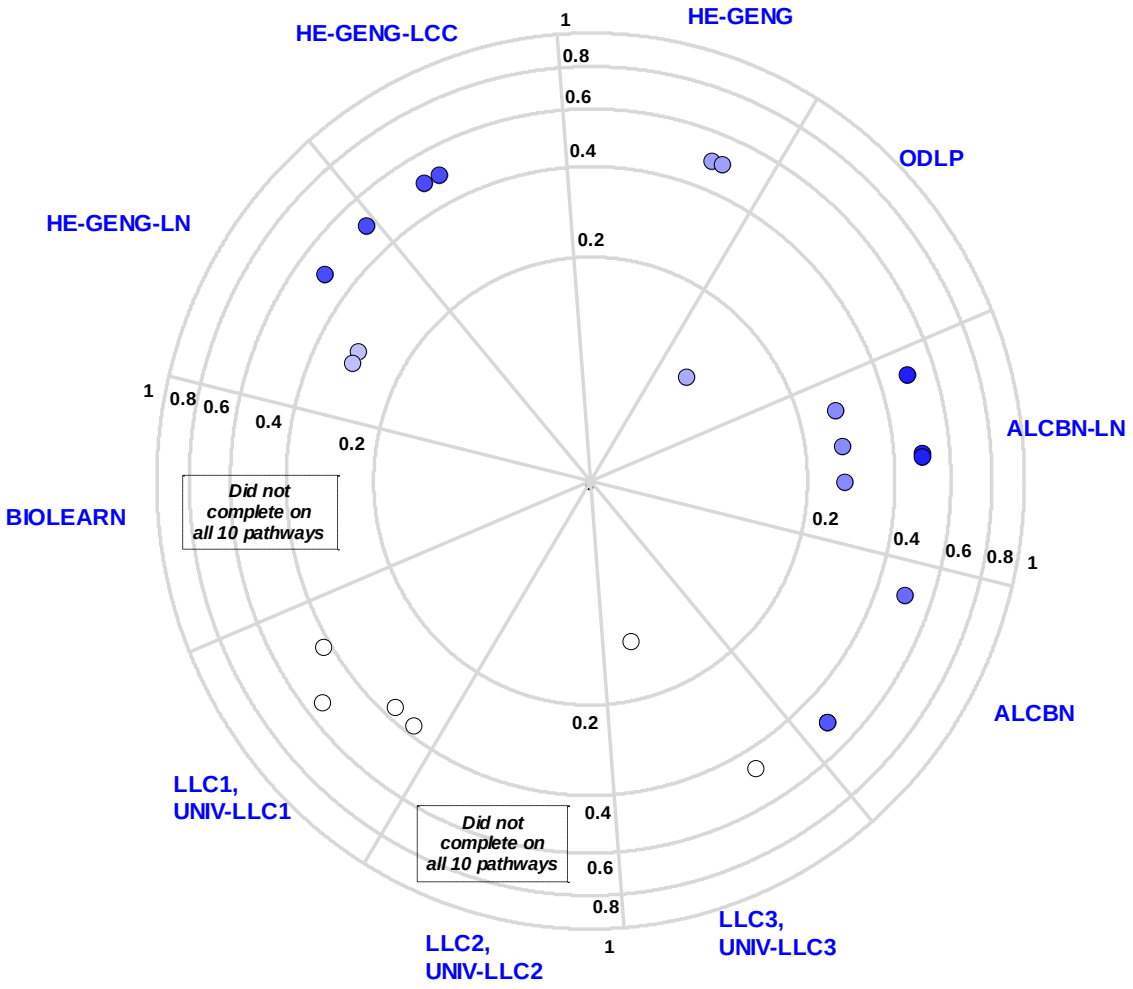

Figure 9: Bulls eye plot for the distance metric and the number of experiments averaged over 10 local causal pathways. Methods are denoted by circles. Location of the circles corresponds to values of the distance metric: the closer is circle to the center, the smaller (better) is the distance. Color of the circles corresponds to the number of experiments: the lighter is color, the more experiments are required.

under the assumption that all variables in the local neighborhood of the target are manipulable; and (iii) ODLP runs orders of magnitude faster than other methods (in most cases within 10-15 minutes for datasets with thousands of variables). A secondary contribution of this study is that we introduced local versions of prior methods for active learning of the entire/global causal networks so that the modified methods scale much better than the original techniques for this task.

There are several major directions for extending this work. First, further development of ODLP for situations when the target variable cannot be manipulated (e.g., it is a disease in humans) and therefore it is challenging to identify effects of the target variable. One

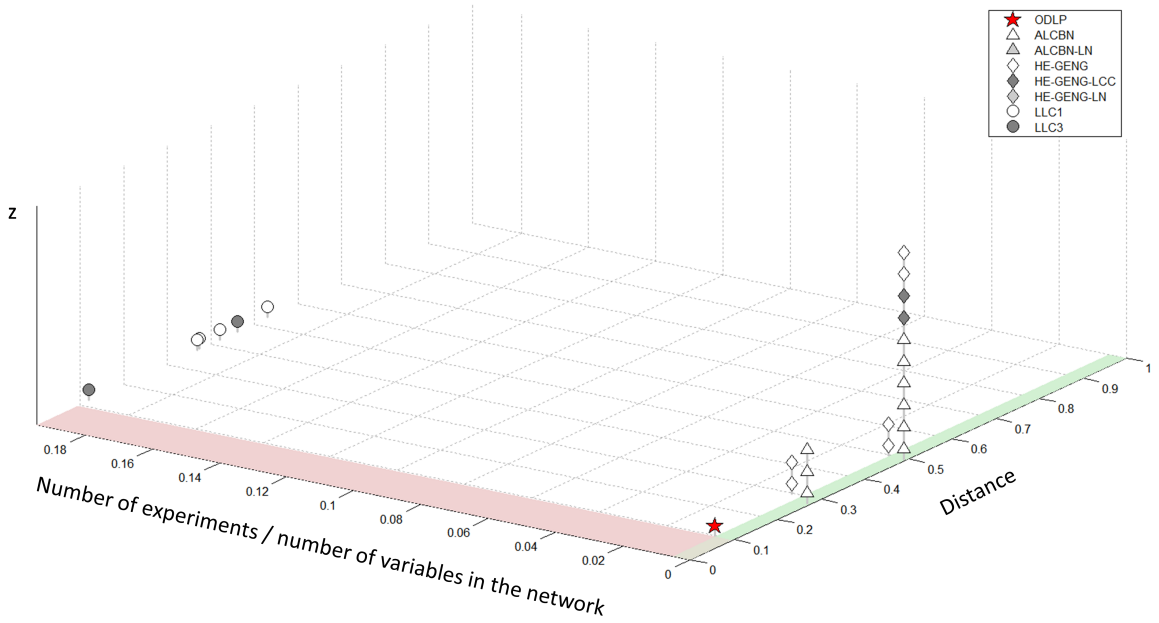

Figure 10: Distance versus number of experiments/number of variables in the network averaged over 10 local causal pathways. The vertical (z) dimension is used to produce a jitter plot so that multiple methods that have the same values of distance and number of experiments/number of variables in the network are not hidden in the graph. Methods located in the pale red area have smaller (better) distance than ODLP, and methods located in the pale green area require smaller number of experiments relative to the number of variables in the network.

possible strategy to solve this problem is to first identify all causes of the target variable and then identify effects through knowledge gained by manipulation of direct causes of the target variable. Second, extension of the ODLP method to work when there are hidden variables and/or feedback cycles. Related to this, the completeness of the algorithm can be improved by incorporating multi-variable manipulation experiments. Third, utilizing existing methods for causal orientation from non-experimental data to avoid unnecessary experiments, to the extent that these methods can produce accurate results in given distributions. These include both classical independence constraint-based (e.g., v-structure) techniques (Spirtes et al., 2000; Yin et al., 2008) or newer methods that can orient pairs of variables (Statnikov et al., 2012; Shimizu et al., 2006; Hoyer et al., 2009; Zhang and Hyvärinen, 2008; Daniusis et al., 2012; Janzing et al., 2012; Mooij et al., 2010). These newer methods could uncover the orientation of edges in non-linear (e.g. additive noise models (Hoyer et al., 2009)) or non-Gaussian (e.g. LinGAM (Shimizu et al., 2006)) cases, which are common in data from the biomedical domain. Fourth, further modifications of the existing state-of-the-art methods for active learning of the entire/global networks to adopt them for local causal pathway discovery task and seek to minimize the number of experiments. For instance, methods other than the PC algorithm could be implemented as

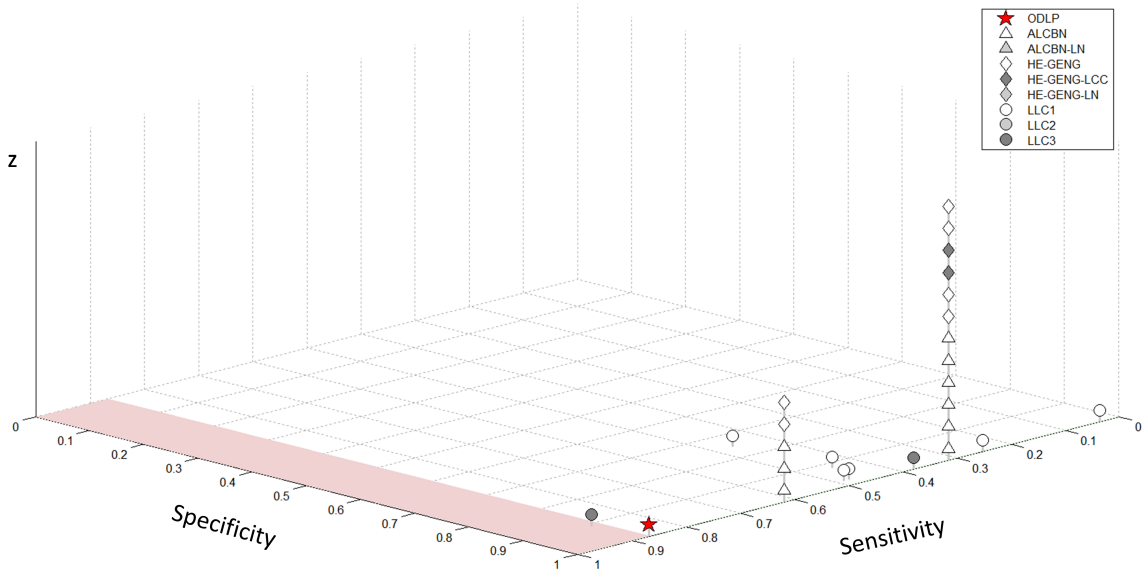

Figure 11: Sensitivity versus specificity averaged over 10 local causal pathways. The vertical (z) dimension is used to produce a jitter plot so that multiple methods that have the same values of sensitivity and specificity in the network are not hidden in the graph. Methods located in the pale red area have larger sensitivity than ODLP. There are no methods that have larger specificity than ODLP.

the starting point for edge orientation. The PC-Stable (Colombo and Maathuis, 2014) and GES algorithms (Chickering, 2003) might lead to increased accuracy, and Richardson's CCD Algorithm (Richardson, 1996) is applicable when acyclicity is not assumed. Finally fifth, extending the empirical comparison study to real (i.e., non-simulated) high-dimensional data. Use of real data is challenging because (i) for most large-scale systems the underlying causal relations are not known, and (ii) obtaining real experimental data is very expensive. Performing such studies in other domains (e.g., economics, marketing, ecology, etc.) is also worthwhile.

Acknowledgments

The evaluation of the methods in this work was supported in part by the grants 1UL1 RR029893 from the National Center for Research Resources and R01 LM011179-01A1 from the National Library of Medicine, National Institutes of Health. The authors thank Frederick Eberhardt and Antti Hyttinen for providing codes of the LLC methods and advice on running these algorithms in the empirical study.

Appendix A. List of Variants of the ALCBN, HE-GENG, and LLC Methods Used in This Work

ALCBN:
- First use the PC algorithm to learn an undirected or partially directed global graph from observational data (i.e., graph over all observed variables).
- Then orient edges by sequentially manipulating variables chosen by some decision criterion.

Method variant name	*Method variant description*
1. ***ALCBN.S.MINIMAX***	Starting from the undirected graph, use mini-max decision criterion to select variables for manipulation.
2. ***ALCBN.S.MAXIMIN***	Starting from the undirected graph, use maxi-min decision criterion to select variables for manipulation.
3. ***ALCBN.S.LAPLACE***	Starting from the undirected graph, use Laplace decision criterion to select variables for manipulation.
4. ***ALCBN.D.MINIMAX***	Starting from the partially directed graph, use mini-max decision criterion to select variables for manipulation.
5. ***ALCBN.D.MAXIMIN***	Starting from the partially directed graph, use maxi-min decision criterion to select variables for manipulation.
6. ***ALCBN.D.LAPLACE***	Starting from the partially directed graph, use Laplace decision criterion to select variables for manipulation.
7. ***ALCBN-LN.S.MINIMAX***	Same as *ALCBN.S.MINIMAX*, but select variables for manipulation only from the local causal pathway of the target.
8. ***ALCBN-LN.S.MAXIMIN***	Same as *ALCBN.S.MAXIMIN*, but select variables for manipulation only from the local causal pathway of the target.
9. ***ALCBN-LN.S.LAPLACE***	Same as *ALCBN.S.LAPLACE*, but select variables for manipulation only from the local causal pathway of the target.
10. ***ALCBN-LN.D.MINIMAX***	Same as *ALCBN.D.MINIMAX*, but select variables for manipulation only from the local causal pathway of the target.
11. ***ALCBN-LN.D.MAXIMIN***	Same as *ALCBN.D.MAXIMIN*, but select variables for manipulation only from the local causal pathway of the target.
12. ***ALCBN-LN.D.LAPLACE***	Same as *ALCBN.D.LAPLACE*, but select variables for manipulation only from the local causal pathway of the target.

Method of He and Geng (HE-GENG):
- First use the PC algorithm to learn an undirected or partially directed global graph from observational data (i.e., graph over all observed variables).
- Then orient edges in each obtained chain component separately by sequentially manipulating variables chosen by some decision criterion.

Method variant name	*Method variant description*
1. ***HE-GENG.S.MINIMAX***	Starting from the undirected graph, use mini-max decision criterion to select variables for manipulation.
2. ***HE-GENG.S.ENTROPY***	Starting from the undirected graph, use maxi-min entropy decision criterion to select variables for manipulation.
3. ***HE-GENG.D.MINIMAX***	Starting from the partially directed graph, use mini-max decision criterion to select variables for manipulation.
4. ***HE-GENG.D.ENTROPY***	Starting from the partially directed graph, use maximum entropy decision criterion to select variables for manipulation.
5. ***HE-GENG-LCC.S.MINIMAX***	Same as *HE-GENG.S.MINIMAX*, but select variables for manipulation only from the local chain component of the target.
6. ***HE-GENG-LCC.S.ENTROPY***	Same as *HE-GENG.S.ENTROPY*, but select variables for manipulation only from the local chain component of the target.
7. ***HE-GENG-LCC.D.MINIMAX***	Same as *HE-GENG.D.MINIMAX*, but select variables for manipulation only from the local chain component of the target.
8. ***HE-GENG-LCC.D.ENTROPY***	Same as *HE-GENG.D.ENTROPY*, but select variables for manipulation only from the local chain component of the target.
9. ***HE-GENG-LN.S.MINIMAX***	Same as *HE-GENG.S.MINIMAX*, but select variables for manipulation only from the local causal pathway of the target.
10. ***HE-GENG-LN.S.ENTROPY***	Same as *HE-GENG.S.ENTROPY*, but select variables for manipulation only from the local causal pathway of the target.
11. ***HE-GENG-LN.D.MINIMAX***	Same as *HE-GENG.D.MINIMAX*, but select variables for manipulation only from the local causal pathway of the target.
12. ***HE-GENG-LN.D.ENTROPY***	Same as *HE-GENG.D.ENTROPY*, but select variables for manipulation only from the local causal pathway of the target.

Table A1: Conditional independence constraint-based structure learning methods/variants used in this work. Modifications of the original methods that focus on discovery of local causality are highlighted.

LLC:

- Assume linear relations between variables. These relations can be represented by an "effects matrix" where each element is the coefficient of the linear relation between variables.
- From each manipulated dataset, derive constraints on the effects matrix which are combined into a linear system (we refer to these constraints as "main constraints").
- Additionally assuming faithfulness allows:
  - utilizing PC algorithm on manipulated data and possibly observational data to learn adjacencies between variables. Non-adjacent variables imply additional constraints on the effects matrix that are added to the linear system (we refer to these constraints as "0-constraints").
  - defining an optimal order of variables for manipulation geared towards identification of the effects matrix.
- Solve the above linear system to identify the effects matrix.
- Elements in the effects matrix correspond to coefficients of the underlying linear relations.
- Filter the effects matrix to obtain edges in the output graph using one of the following methods:
  - THR: Obtain edges by applying a threshold of 0.1 on the coefficients of the identified effects matrix.
  - ALPHA: Using 100 data permutations, estimate the null distribution of the coefficients of the effects matrix. Obtain edges by choosing significant coefficients at 5% alpha level.
  - FDR: Using 100 data permutations, estimate the null distribution of the coefficients of the effects matrix. Obtain edges by choosing significant coefficients at 5% FDR level.
  - BOOTSTRAP: Identify effects matrix in 30 datasets sampled from the original data with replacement. Obtain edges by choosing elements of the effects matrix whose mean coefficient over resampled datasets is higher than the standard deviation.

Method variant name	*Method variant description*
1. ***LLC1.THR***	Manipulate all variables associated with the target to obtain manipulated data. Derive main constraints on the effects matrix and solve the linear system using the method LLC1. Find edges in graph by method THR.
2. ***LLC1.ALPHA***	Same as *LLC1.THR*, but use method ALPHA to find edges in graph.
3. ***LLC1.FDR***	Same as *LLC1.THR*, but use method FDR to find edges in graph.
4. ***LLC2.THR***	Same as *LLC1.THR*, but use LLC2 method to derive main constraints on the effects matrix and solve the linear system.
5. ***LLC2.ALPHA***	Same as *LLC1.THR*, but use LLC2 method to derive main constraints on the effects matrix and solve the linear system and method ALPHA to find edges in graph.
6. ***LLC2.FDR***	Same as *LLC1.THR*, but use LLC2 method to derive main constraints on the effects matrix and solve the linear system and method FDR to find edges in graph.
7. ***LLC3.THR***	Same as *LLC1.THR*, but use LLC3 method to derive main constraints on the effects matrix and solve the linear system.
8. ***LLC3.BOOTSTRAP***	Same as *LLC1.THR*, but use LLC3 method to derive main constraints on the effects matrix and solve the linear system and method BOOTSTRAP to find edges in graph.
9. ***LLC2-F1.THR***	Manipulate a random variable to obtain manipulated data. Apply PC algorithm on manipulated data to obtain 0-constraints on the effects matrix. Derive main constraints on the effects matrix and solve the linear system using the method LLC2. Determine optimal variable for manipulation. Repeat the above steps until the effects matrix has been identified. Find edges in graph by method THR.
10. ***LLC2-F1.ALPHA***	Same as *LLC2-F1.THR*, but use method ALPHA to find edges in graph.
11. ***LLC2-F1.FDR***	Same as *LLC2-F1.THR*, but use method FDR to find edges in graph.
12. ***LLC2-F2.THR***	Same as *LLC2-F1.THR*, but apply PC to both observational and manipulated data to obtain 0-constraints on the effects matrix.
13. ***LLC2-F2.ALPHA***	Same as *LLC2-F1.THR*, but apply PC to both observational and manipulated data to obtain 0-constraints on the effects matrix and method ALPHA to find edges in graph.
14. ***LLC2-F2.FDR***	Same as *LLC2-F1.THR*, but apply PC to both observational and manipulated data to obtain 0-constraints on the effects matrix and method FDR to find edges in graph.
15. ***LLC3-F2.THR***	Same as *LLC2-F1.THR*, but apply PC to both observational and manipulated data to obtain 0-constraints on the effects matrix and method LLC3 to derive main constraints on the effects matrix and solve the linear system.
16. ***LLC3-F2.BOOTSTRAP***	Same as *LLC2-F1.THR*, but apply PC to both observational and manipulated data to obtain 0-constraints on the effects matrix, method LLC3 to derive main constraints on the effects matrix and solve the linear system, and method BOOTSTRAP to find edges in graph.
17. ***UNIV-LLC1.THR***	18. ***UNIV-LLC2.ALPHA***	19. ***UNIV-LLC2-F1.THR***	20. ***UNIV-LLC2-F2.ALPHA***	Same as above methods without prefix "UNIV", except for using only variables that are univariately associated with the target to identify the effects matrix. All other variables are not considered at all by the method.
21. ***UNIV-LLC1.ALPHA***	22. ***UNIV-LLC2.FDR***	23. ***UNIV-LLC2-F1.ALPHA***	24. ***UNIV-LLC2-F2.FDR***	
25. ***UNIV-LLC1.FDR***	26. ***UNIV-LLC3.THR***	27. ***UNIV-LLC2-F1.FDR***	28. ***UNIV-LLC3-F2.THR***	
29. ***UNIV-LLC2.THR***	30. ***UNIV-LLC3.BOOTSTRAP***	31. ***UNIV-LLC2-F2.THR***	32. ***UNIV-LLC3-F2.BOOTSTRAP***	

Table A2: Linear cyclic models-based structure learning methods/variants used in this work. Modifications of the original methods that focus on discovery of local causality are highlighted.

Appendix B. Information about P1000 and P1M Networks

Vertex	Parent	Coefficient	Noise Coefficient
1	2	0.9	0
2	40	0.8	0.2
	41	0.8	0.2
3	6	0.6	0
4	6	0.8	0
5	2	0.8	0
6	2	0.9	0
7	8	0.9	0
8	9	1.1	0
9	39	0.9	0
10	9	0.8	0
11	9	0.7	0
13	12	0.6	0
14	13	0.8	0
15	16	0.7	0
16	12	0.9	0
17	15	0.9	0
18	54	0.7	0.2
19	18	0.9	0
20	19	0.1	0
21	54	0.6	0.1
22	21	0.9	0
23	22	0.2	0
24	23	0.4	0.3
26	25	0.9	0.2
	17	0.6	0.2

Vertex	Parent	Coefficient	Noise Coefficient
27	26	0.4	0.1
28	17	0.5	0.1
29	30	0.1	0.1
30	31	0.7	0.2
31	32	0.9	0.1
32	35	0.6	0.1
33	35	0.9	0.2
34	35	0.5	0.3
35	36	0.1	0.2
	37	0.6	0.2
37	38	0.8	0.2
38	39	0.1	0.1
40	39	0.5	0.2
47	44	0.7	0.3
48	45	0.9	0.3
49	46	0.1	0.1
50	48	0.3	0.2
	49	0.4	0.2
51	20	0.9	0.1
	50	0.4	0.1
52	20	0.6	0.2
	53	0.8	0.2
54 (T)	1	0.3	0.1
	7	0.3	0.1
	12	0.3	0.1

Table B1: Parameterization of the P1000 network. Data for a given vertex/variable V is a linear combination of its parents and Gaussian noise: $V = \sum_p (Coef_{parent_p} + N(0, Coef_{noise_p}))$. The data for vertices without any parents was sampled from Gaussian distribution $N(0, 1)$ and is not shown in the following table.

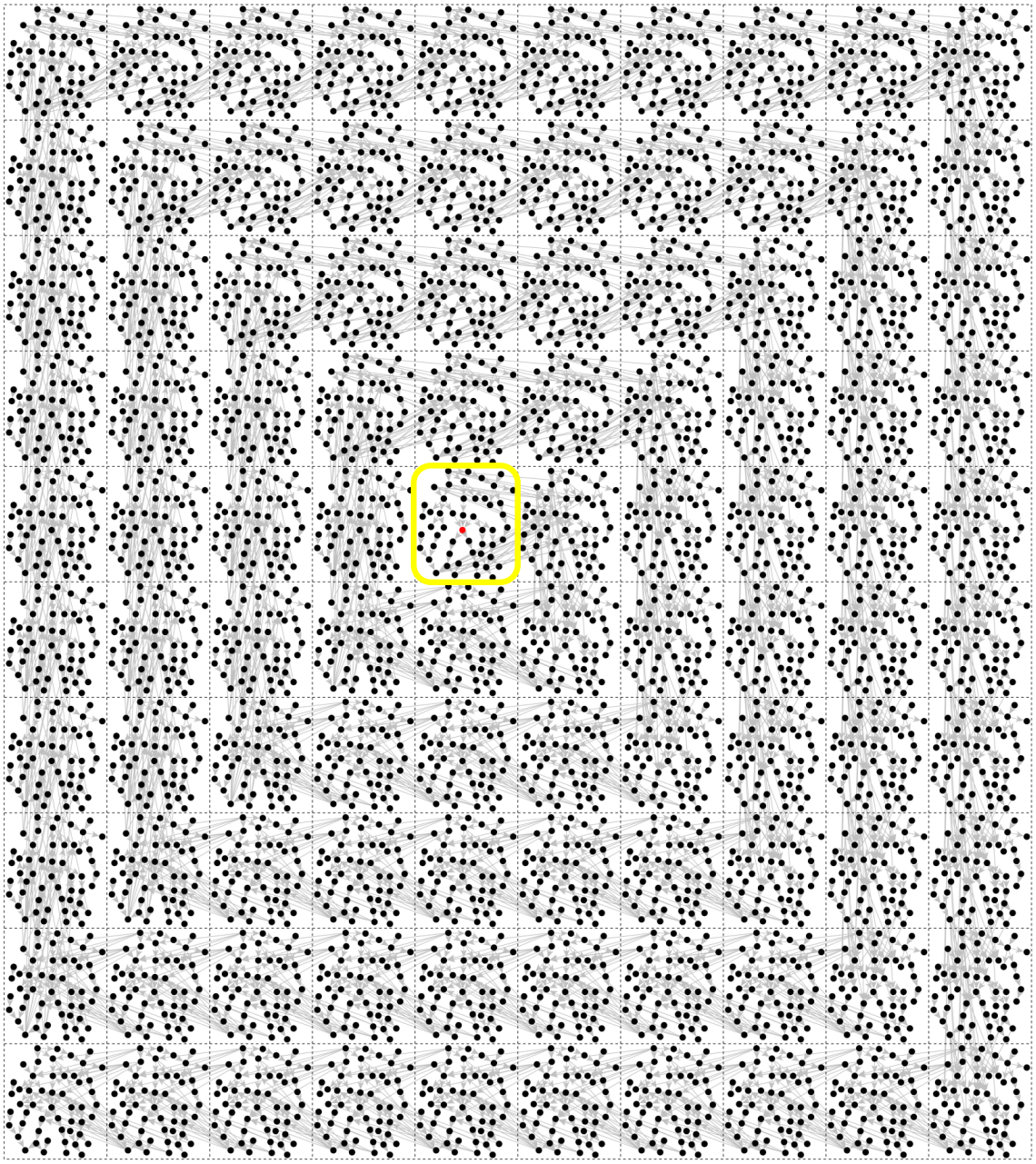

Figure B1: A fragment of the P1M network. The little red dot in the middle (in the tile with yellow outline) represents the target variable, black dots represent other variables. Only the connected components of the first 100 tiles were shown.

Appendix C. Detailed Results of Empirical Experiments

	REGED				P1000				P1M			
Method Name	Sensitivity	Specificity	Distance	N exp	Sensitivity	Specificity	Distance	N exp	Sensitivity	Specificity	Distance	N exp
ODLP	86.7%	100.0%	9.4%	1	100.0%	100.0%	0.0%	18	100.0%	100.0%	0.0%	19
ALCBN.S.MINIMAX	86.7%	100.0%	9.4%	47	0.0%	99.8%	70.7%	577	T1			
ALCBN.S.MAXIMIN	86.7%	100.0%	9.4%	91	0.0%	99.8%	70.7%	368	T1			
ALCBN.S.LAPLACE	86.7%	100.0%	9.4%	62	0.0%	99.8%	70.7%	442	T1			
ALCBN.D.MINIMAX	26.7%	99.6%	51.9%	0	0.0%	99.8%	70.7%	0	T1			
ALCBN.D.MAXIMIN	26.7%	99.6%	51.9%	0	0.0%	99.8%	70.7%	0	T1			
ALCBN.D.LAPLACE	26.7%	99.6%	51.9%	0	0.0%	99.8%	70.7%	0	T1			
ALCBN-LN.S.MINIMAX	86.7%	100.0%	9.4%	1	0.0%	99.8%	70.7%	1	T1			
ALCBN-LN.S.MAXIMIN	86.7%	100.0%	9.4%	1	0.0%	99.8%	70.7%	1	T1			
ALCBN-LN.S.LAPLACE	86.7%	100.0%	9.4%	1	0.0%	99.8%	70.7%	1	T1			
ALCBN-LN.D.MINIMAX	26.7%	99.6%	51.9%	0	0.0%	99.8%	70.7%	0	T1			
ALCBN-LN.D.MAXIMIN	26.7%	99.6%	51.9%	0	0.0%	99.8%	70.7%	0	T1			
ALCBN-LN.D.LAPLACE	26.7%	99.6%	51.9%	0	0.0%	99.8%	70.7%	0	T1			
HE-GENG.S.MINIMAX	86.7%	100.0%	9.4%	337	0.0%	99.8%	70.7%	32	T1			
HE-GENG.S.ENTROPY	86.7%	100.0%	9.4%	337	0.0%	99.8%	70.7%	32	T1			
HE-GENG.D.MINIMAX	26.7%	99.6%	51.9%	0	0.0%	99.8%	70.7%	0	T1			
HE-GENG.D.ENTROPY	26.7%	99.6%	51.9%	0	0.0%	99.8%	70.7%	0	T1			
HE-GENG-LCC.S.MINIMAX	86.7%	100.0%	9.4%	108	0.0%	99.8%	70.7%	63	T1			
HE-GENG-LCC.S.ENTROPY	86.7%	100.0%	9.4%	108	0.0%	99.8%	70.7%	63	T1			
HE-GENG-LCC.D.MINIMAX	26.7%	99.6%	51.9%	0	0.0%	99.8%	70.7%	0	T1			
HE-GENG-LCC.D.ENTROPY	26.7%	99.6%	51.9%	0	0.0%	99.8%	70.7%	0	T1			
HE-GENG-LN.S.MINIMAX	86.7%	100.0%	9.4%	13	0.0%	99.8%	70.7%	5	T1			
HE-GENG-LN.S.ENTROPY	86.7%	100.0%	9.4%	13	0.0%	99.8%	70.7%	5	T1			
HE-GENG-LN.D.MINIMAX	26.7%	99.6%	51.9%	0	0.0%	99.8%	70.7%	0	T1			
HE-GENG-LN.D.ENTROPY	26.7%	99.6%	51.9%	0	0.0%	99.8%	70.7%	0	T1			
LLC1.THR	86.7%	50.5%	36.3%	540	100.0%	51.1%	34.6%	85	T4			
LLC1.ALPHA	6.7%	98.8%	66.0%	540	0.0%	99.9%	70.7%	85	T4			
LLC1.FDR	6.7%	99.7%	66.0%	540	0.0%	100.0%	70.7%	85	T4			
LLC2.THR	T1				0.0%	100.0%	70.7%	85	T4			
LLC2.ALPHA	T2				T2				T4			
LLC2.FDR	T2				T2				T4			
LLC3.THR	0.0%	100.0%	70.7%	540	0.0%	100.0%	70.7%	85	T4			
LLC3.BOOTSTRAP	100.0%	93.8%	4.4%	540	100.0%	69.9%	21.3%	85	T4			
LLC2-F1.THR	T1				T1				T1			
LLC2-F1.ALPHA	T2				T2				T2			
LLC2-F1.FDR	T2				T2				T2			
LLC2-F2.THR	T1				T1				T1			
LLC2-F2.ALPHA	T2				T2				T2			
LLC2-F2.FDR	T2				T2				T2			
LLC3-F2.THR	T4				T4				T4			
LLC3-F2.BOOTSTRAP	T4				T4				T4			
UNIV-LLC1.THR	80.0%	73.6%	23.4%	540	100.0%	95.0%	3.5%	85	T4			
UNIV-LLC1.ALPHA	6.7%	98.8%	66.0%	540	0.0%	99.5%	70.7%	85	T4			
UNIV-LLC1.FDR	6.7%	99.7%	66.0%	540	0.0%	100.0%	70.7%	85	T4			
UNIV-LLC2.THR	0.0%	100.0%	70.7%	540	100.0%	98.5%	1.1%	85	T4			
UNIV-LLC2.ALPHA	T2				60.0%	99.9%	28.3%	85	T4			
UNIV-LLC2.FDR	T2				40.0%	100.0%	42.4%	85	T4			
UNIV-LLC3.THR	80.0%	73.8%	23.3%	540	100.00%	93.48%	4.6%	85	T4			
UNIV-LLC3.BOOTSTRAP	13.3%	100.0%	61.3%	540	0.00%	100.00%	70.7%	85	T4			
UNIV-LLC2-F1.THR	0.0%	100.0%	70.7%	4	0.0%	100.0%	70.7%	5	T1			
UNIV-LLC2-F1.ALPHA	T2				40.0%	99.1%	42.4%	5	T2			
UNIV-LLC2-F1.FDR	T2				0.0%	100.0%	70.7%	5	T2			
UNIV-LLC2-F2.THR	T1				0.0%	99.9%	70.7%	2	T1			
UNIV-LLC2-F2.ALPHA	T2				40.0%	97.2%	42.5%	2	T2			
UNIV-LLC2-F2.FDR	T2				20.0%	99.7%	56.6%	2	T2			
UNIV-LLC3-F2.THR	T4				0.0%	100.0%	70.7%	11	T4			
UNIV-LLC3-F2.BOOTSTRAP	T4				100.0%	96.8%	2.2%	11	T4			
BIOLEARN.NG	T3				0.0%	99.6%	70.7%	85	T3			
BIOLEARN.BDE	T3				20.0%	100.0%	56.6%	85	T3			

Explanation of termination/ failure codes:

T1 = Experiments when the algorithm was terminated after 30 days of single-core time limit for tasks that cannot be easily parallelized;

T2 = Experiments when the algorithm was terminated after 3,000 day multi-core time limit (spread over 100 cores) for tasks that can be easily parallelized;

T3 = Experiments when the authors' implementation of the algorithm failed for unknown reason;

T4 = Experiments when the algorithm required more than 48 GB RAM.

Table C1: Detailed results of experiments for REGED, P1000, and P1M networks.

	ECOLI (agaR)				ECOLI (allR)				ECOLI (zur)				ECOLI (lexA)			
Method Name	Sensitivity	Specificity	Distance	N exp	Sensitivity	Specificity	Distance	N exp	Sensitivity	Specificity	Distance	N exp	Sensitivity	Specificity	Distance	N exp
ODLP	87.5%	99.9%	8.8%	1	100.0%	99.9%	0.1%	1	100.0%	99.9%	0.1%	3	90.7%	99.9%	6.6%	1
ALCBN.S.MINIMAX	87.5%	100.0%	8.8%	264	100.0%	99.9%	0.1%	162	100.0%	99.9%	0.1%	213	90.7%	99.9%	6.6%	4
ALCBN.S.MAXIMIN	87.5%	100.0%	8.8%	269	100.0%	99.9%	0.1%	436	100.0%	99.9%	0.1%	288	90.7%	99.9%	6.6%	4
ALCBN.S.LAPLACE	87.5%	100.0%	8.8%	212	100.0%	99.9%	0.1%	143	100.0%	99.9%	0.1%	292	90.7%	99.9%	6.6%	4
ALCBN.D.MINIMAX	100.0%	100.0%	0.0%	0	100.0%	99.9%	0.1%	0	100.0%	99.9%	0.1%	4	3.7%	98.3%	68.1%	0
ALCBN.D.MAXIMIN	100.0%	100.0%	0.0%	0	100.0%	99.9%	0.1%	0	100.0%	99.9%	0.1%	4	3.7%	98.3%	68.1%	0
ALCBN.D.LAPLACE	100.0%	100.0%	0.0%	0	100.0%	99.9%	0.1%	0	100.0%	99.9%	0.1%	3	3.7%	98.3%	68.1%	0
ALCBN-LN.S.MINIMAX	87.5%	100.0%	8.8%	1	100.0%	99.9%	0.1%	1	100.0%	99.9%	0.1%	1	90.7%	99.9%	6.6%	1
ALCBN-LN.S.MAXIMIN	87.5%	100.0%	8.8%	1	100.0%	99.9%	0.1%	1	100.0%	99.9%	0.1%	1	90.7%	99.9%	6.6%	1
ALCBN-LN.S.LAPLACE	87.5%	100.0%	8.8%	1	100.0%	99.9%	0.1%	1	100.0%	99.9%	0.1%	1	90.7%	99.9%	6.6%	1
ALCBN-LN.D.MINIMAX	100.0%	100.0%	0.0%	0	100.0%	99.9%	0.1%	0	100.0%	99.9%	0.1%	1	3.7%	98.3%	68.1%	0
ALCBN-LN.D.MAXIMIN	100.0%	100.0%	0.0%	0	100.0%	99.9%	0.1%	0	100.0%	99.9%	0.1%	1	3.7%	98.3%	68.1%	0
ALCBN-LN.D.LAPLACE	100.0%	100.0%	0.0%	0	100.0%	99.9%	0.1%	0	100.0%	99.9%	0.1%	1	3.7%	98.3%	68.1%	0
HE-GENG.S.MINIMAX	87.5%	100.0%	8.8%	86	100.0%	99.9%	0.1%	25	T1				T1			
HE-GENG.S.ENTROPY	87.5%	100.0%	8.8%	86	100.0%	99.9%	0.1%	25	T1				T1			
HE-GENG.D.MINIMAX	100.0%	100.0%	0.0%	0	100.0%	99.9%	0.1%	0	100.0%	99.9%	0.1%	74	3.7%	98.3%	68.1%	0
HE-GENG.D.ENTROPY	100.0%	100.0%	0.0%	0	100.0%	99.9%	0.1%	0	100.0%	99.9%	0.1%	74	3.7%	98.3%	68.1%	0
HE-GENG-LCC.S.MINIMAX	87.5%	100.0%	8.8%	86	100.0%	99.9%	0.1%	25	T1				T1			
HE-GENG-LCC.S.ENTROPY	87.5%	100.0%	8.8%	86	100.0%	99.9%	0.1%	25	T1				T1			
HE-GENG-LCC.D.MINIMAX	100.0%	100.0%	0.0%	0	100.0%	99.9%	0.1%	0	100.0%	99.9%	0.1%	5	3.7%	98.3%	68.1%	0
HE-GENG-LCC.D.ENTROPY	100.0%	100.0%	0.0%	0	100.0%	99.9%	0.1%	0	100.0%	99.9%	0.1%	5	3.7%	98.3%	68.1%	0
HE-GENG-LN.S.MINIMAX	87.5%	100.0%	8.8%	2	100.0%	99.9%	0.1%	6	100.0%	99.9%	0.1%	2	77.8%	99.6%	15.7%	30
HE-GENG-LN.S.ENTROPY	87.5%	100.0%	8.8%	2	100.0%	99.9%	0.1%	6	100.0%	99.9%	0.1%	2	77.8%	99.6%	15.7%	30
HE-GENG-LN.D.MINIMAX	100.0%	100.0%	0.0%	0	100.0%	99.9%	0.1%	0	100.0%	99.9%	0.1%	5	3.7%	98.3%	68.1%	0
HE-GENG-LN.D.ENTROPY	100.0%	100.0%	0.0%	0	100.0%	99.9%	0.1%	0	100.0%	99.9%	0.1%	5	3.7%	98.3%	68.1%	0
LLC1.THR	0.0%	100.0%	70.7%	82	0.0%	100.0%	70.7%	88	0.0%	100.0%	70.7%	90	0.0%	100.0%	70.7%	147
LLC1.ALPHA	100.0%	49.8%	35.5%	82	100.0%	50.3%	35.2%	88	100.0%	50.6%	35.0%	90	96.3%	51.7%	34.2%	147
LLC1.FDR	100.0%	51.3%	34.4%	82	100.0%	51.4%	34.3%	88	100.0%	51.6%	34.2%	90	96.3%	52.8%	33.5%	147
LLC2.THR	T1				T1				T1				T1			
LLC2.ALPHA	T2				T2				T2				T2			
LLC2.FDR	T2				T2				T2				T2			
LLC3.THR	0.0%	100.0%	70.7%	82	10.0%	100.0%	63.6%	88	0.0%	100.0%	70.7%	90	0.0%	100.0%	70.7%	147
LLC3.BOOTSTRAP	100.0%	67.2%	23.2%	82	100.0%	72.0%	19.8%	88	100.0%	69.4%	21.7%	90	98.2%	78.6%	15.2%	147
LLC2-F1.THR	T1				T1				T1				T1			
LLC2-F1.ALPHA	T2				T2				T2				T2			
LLC2-F1.FDR	T2				T2				T2				T2			
LLC2-F2.THR	T1				T1				T1				T1			
LLC2-F2.ALPHA	T2				T2				T2				T2			
LLC2-F2.FDR	T2				T2				T2				T2			
LLC3-F2.THR	T4				T4				T4				T4			
LLC3-F2.BOOTSTRAP	T4				T4				T4				T4			
UNIV-LLC1.THR	0.0%	100.0%	70.7%	82	0.0%	100.0%	70.7%	88	0.0%	100.0%	70.7%	90	0.0%	100.0%	70.7%	147
UNIV-LLC1.ALPHA	100.0%	99.7%	0.3%	82	100.0%	99.7%	0.3%	88	100.0%	99.3%	0.5%	90	94.4%	99.5%	3.9%	147
UNIV-LLC1.FDR	87.5%	100.0%	8.8%	82	100.0%	99.9%	0.1%	88	100.0%	100.0%	0.0%	90	90.7%	100.0%	6.6%	147
UNIV-LLC2.THR	0.0%	100.0%	70.7%	82	0.0%	100.0%	70.7%	88	0.0%	100.0%	70.7%	90	0.0%	100.0%	70.7%	147
UNIV-LLC2.ALPHA	100.0%	99.7%	0.3%	82	100.0%	99.7%	0.3%	88	100.0%	99.3%	0.5%	90	94.4%	99.5%	3.9%	147
UNIV-LLC2.FDR	87.5%	100.0%	8.8%	82	100.0%	99.9%	0.1%	88	100.0%	100.0%	0.0%	90	90.7%	100.0%	6.6%	147
UNIV-LLC3.THR	100.0%	98.3%	1.2%	82	100.0%	98.6%	1.0%	88	100.0%	98.3%	1.2%	90	87.0%	97.7%	9.3%	147
UNIV-LLC3.BOOTSTRAP	100.0%	98.0%	1.4%	82	100.0%	98.6%	1.0%	88	100.0%	97.8%	1.5%	90	88.9%	98.3%	8.0%	147
UNIV-LLC2-F1.THR	0.0%	100.0%	70.7%	5	0.0%	100.0%	70.7%	7	50.0%	99.9%	35.4%	6	0.0%	100.0%	70.7%	12
UNIV-LLC2-F1.ALPHA	75.0%	99.7%	17.7%	5	100.0%	99.1%	0.6%	7	100.0%	99.7%	0.2%	6	96.3%	96.4%	3.6%	12
UNIV-LLC2-F1.FDR	62.5%	99.9%	26.5%	5	100.0%	99.8%	0.1%	7	100.0%	99.8%	0.1%	6	72.2%	99.5%	19.7%	12
UNIV-LLC2-F2.THR	37.5%	99.9%	44.2%	4	0.0%	100.0%	70.7%	6	83.3%	99.8%	11.8%	5	0.0%	100.0%	70.7%	11
UNIV-LLC2-F2.ALPHA	50.0%	99.8%	35.4%	4	100.0%	99.0%	0.7%	6	100.0%	99.6%	0.3%	5	98.2%	96.3%	2.9%	11
UNIV-LLC2-F2.FDR	50.0%	99.9%	35.4%	4	100.0%	99.9%	0.1%	6	83.3%	99.8%	11.8%	5	66.7%	99.6%	23.6%	11
UNIV-LLC3-F2.THR	0.0%	100.0%	70.7%	17	0.0%	100.0%	70.7%	27	0.0%	100.0%	70.7%	32	0.0%	100.0%	70.7%	57
UNIV-LLC3-F2.BOOTSTRAP	75.0%	98.6%	17.7%	19	70.0%	99.1%	21.2%	25	83.3%	98.1%	11.9%	30	59.3%	97.5%	28.9%	46
BIOLEARN.NG	75.0%	99.9%	17.7%	82	50.0%	99.9%	35.4%	88	66.7%	99.9%	23.6%	90	90.7%	99.9%	6.6%	147
BIOLEARN.BDE	0.0%	99.8%	70.7%	82	0.0%	99.9%	70.7%	88	16.7%	99.9%	58.9%	90	50.0%	100.0%	35.4%	147

Table C2: Detailed results of experiments for ECOLI network (4 local causal neighborhoods). See Table C1 for explanation of termination/failure codes T1, T2, T3, and T4.

	YEAST (YBL005W)				YEAST (YFL044C)				YEAST (YLR014C)				YEAST (YKL112W)			
Method Name	Sensitivity	Specificity	Distance	N exp	Sensitivity	Specificity	Distance	N exp	Sensitivity	Specificity	Distance	N exp	Sensitivity	Specificity	Distance	N exp
ODLP	66.7%	99.93%	23.6%	1	66.7%	100.0%	23.6%	1	64.5%	100.0%	25.1%	1	61.0%	99.9%	27.6%	1
ALCBN.S.MINIMAX	T1				T1				T1				61.0%	99.9%	27.6%	1
ALCBN.S.MAXIMIN	T1				T1				T1				61.0%	99.9%	27.6%	1
ALCBN.S.LAPLACE	T1				T1				T1				61.0%	99.9%	27.6%	1
ALCBN.D.MINIMAX	6.7%	99.7%	66.0%	0	26.7%	99.9%	51.9%	0	35.5%	99.9%	45.6%	21	5.3%	98.0%	67.0%	0
ALCBN.D.MAXIMIN	6.7%	99.7%	66.0%	0	26.7%	99.9%	51.9%	0	35.5%	99.9%	45.6%	24	5.3%	98.0%	67.0%	0
ALCBN.D.LAPLACE	6.7%	99.7%	66.0%	0	26.7%	99.9%	51.9%	0	35.5%	99.9%	45.6%	20	5.3%	98.0%	67.0%	0
ALCBN-LN.S.MINIMAX	66.7%	99.9%	23.6%	1	66.7%	100.0%	23.6%	1	64.5%	100.0%	25.1%	1	61.0%	99.9%	27.6%	1
ALCBN-LN.S.MAXIMIN	66.7%	99.9%	23.6%	1	66.7%	100.0%	23.6%	1	64.5%	100.0%	25.1%	1	61.0%	99.9%	27.6%	1
ALCBN-LN.S.LAPLACE	66.7%	99.9%	23.6%	1	66.7%	100.0%	23.6%	1	64.5%	100.0%	25.1%	1	61.0%	99.9%	27.6%	1
ALCBN-LN.D.MINIMAX	6.7%	99.7%	66.0%	0	26.7%	99.9%	51.9%	0	35.5%	99.9%	45.6%	1	5.3%	98.0%	67.0%	0
ALCBN-LN.D.MAXIMIN	6.7%	99.7%	66.0%	0	26.7%	99.9%	51.9%	0	35.5%	99.9%	45.6%	1	5.3%	98.0%	67.0%	0
ALCBN-LN.D.LAPLACE	6.7%	99.7%	66.0%	0	26.7%	99.9%	51.9%	0	35.5%	99.9%	45.6%	1	5.3%	98.0%	67.0%	0
HE-GENG.S.MINIMAX	T1				T1				T1				T1			
HE-GENG.S.ENTROPY	T1				T1				T1				T1			
HE-GENG.D.MINIMAX	6.7%	99.7%	66.0%	0	26.7%	99.9%	51.9%	0	35.5%	99.9%	45.6%	44	5.3%	98.0%	67.0%	0
HE-GENG.D.ENTROPY	6.7%	99.7%	66.0%	0	26.7%	99.9%	51.9%	0	35.5%	99.9%	45.6%	44	5.3%	98.0%	67.0%	0
HE-GENG-LCC.S.MINIMAX	T1				T1				T1				T1			
HE-GENG-LCC.S.ENTROPY	T1				T1				T1				T1			
HE-GENG-LCC.D.MINIMAX	6.7%	99.7%	66.0%	0	26.7%	99.9%	51.9%	0	35.5%	99.9%	45.6%	1	5.3%	98.0%	67.0%	0
HE-GENG-LCC.D.ENTROPY	6.7%	99.7%	66.0%	0	26.7%	99.9%	51.9%	0	35.5%	99.9%	45.6%	1	5.3%	98.0%	67.0%	0
HE-GENG-LN.S.MINIMAX	70.0%	100.0%	21.2%	13	66.7%	100.0%	23.6%	5	64.5%	100.0%	25.1%	11	23.0%	98.6%	54.5%	99
HE-GENG-LN.S.ENTROPY	70.0%	100.0%	21.2%	13	66.7%	100.0%	23.6%	5	64.5%	100.0%	25.1%	11	23.0%	98.6%	54.5%	99
HE-GENG-LN.D.MINIMAX	6.7%	99.7%	66.0%	0	26.7%	99.9%	51.9%	0	35.5%	99.9%	45.6%	2	5.3%	98.0%	67.0%	0
HE-GENG-LN.D.ENTROPY	6.7%	99.7%	66.0%	0	26.7%	99.9%	51.9%	0	35.5%	99.9%	45.6%	2	5.3%	98.0%	67.0%	0
LLC1.THR	0.0%	100.0%	70.7%	328	0.0%	100.0%	70.7%	215	0.0%	100.0%	70.7%	220	0.0%	100.0%	70.7%	804
LLC1.ALPHA	T2				T2				T2				T2			
LLC1.FDR	T2				T2				T2				T2			
LLC2.THR	T1				T1				T1				T1			
LLC2.ALPHA	T2				T2				T2				T2			
LLC2.FDR	T2				T2				T2				T2			
LLC3.THR	T4				T4				T4				T4			
LLC3.BOOTSTRAP	T4				T4				T4				T4			
LLC2-F1.THR	T1				T1				T1				T1			
LLC2-F1.ALPHA	T2				T2				T2				T2			
LLC2-F1.FDR	T2				T2				T2				T2			
LLC2-F2.THR	T1				T1				T1				T1			
LLC2-F2.ALPHA	T2				T2				T2				T2			
LLC2-F2.FDR	T2				T2				T2				T2			
LLC3-F2.THR	T4				T4				T4				T4			
LLC3-F2.BOOTSTRAP	T4				T4				T4				T4			
UNIV-LLC1.THR	0	100.0%	70.7%	328	0.0%	100.0%	70.7%	215	0.0%	100.0%	70.7%	220	0.0%	100.0%	70.7%	804
UNIV-LLC1.ALPHA	86.7%	92.9%	10.7%	328	93.3%	95.3%	5.8%	215	90.3%	95.4%	7.6%	220	90.0%	84.4%	13.1%	804
UNIV-LLC1.FDR	86.7%	92.9%	10.7%	328	93.3%	95.3%	5.8%	215	90.3%	95.4%	7.6%	220	90.0%	84.4%	13.1%	804
UNIV-LLC2.THR	0.0%	100.0%	70.7%	328	0.0%	100.0%	70.7%	215	0.0%	100.0%	70.7%	220	T1			
UNIV-LLC2.ALPHA	70.0%	99.4%	21.2%	328	73.3%	99.7%	18.9%	215	67.7%	99.6%	22.8%	220	T2			
UNIV-LLC2.FDR	53.3%	100.0%	33.0%	328	66.7%	100.0%	23.6%	215	61.3%	100.0%	27.4%	220	T2			
UNIV-LLC3.THR	73.3%	97.8%	18.9%	328	73.3%	98.7%	18.9%	215	83.9%	98.5%	11.5%	220	76.0%	89.5%	18.5%	804
UNIV-LLC3.BOOTSTRAP	0.0%	100.0%	70.7%	328	60.0%	99.1%	28.3%	215	61.3%	99.4%	27.4%	220	0.0%	100.0%	70.7%	804
UNIV-LLC2-F1.THR	3.3%	100.0%	68.4%	10	0.0%	100.0%	70.7%	6	0.0%	100.0%	70.7%	8	T1			
UNIV-LLC2-F1.ALPHA	36.7%	99.6%	44.8%	10	93.3%	97.7%	5.0%	6	90.3%	97.7%	7.0%	8	T2			
UNIV-LLC2-F1.FDR	3.3%	100.0%	68.4%	10	66.7%	99.9%	23.6%	6	61.3%	99.8%	27.4%	8	T2			
UNIV-LLC2-F2.THR	3.33%	100.0%	68.4%	9	0.0%	100.0%	70.7%	7	0.0%	100.0%	70.7%	7	T1			
UNIV-LLC2-F2.ALPHA	63.3%	98.2%	26.0%	9	93.3%	97.6%	5.0%	7	90.3%	97.7%	7.0%	7	T2			
UNIV-LLC2-F2.FDR	6.7%	100.0%	66.0%	9	60.0%	99.8%	28.3%	7	64.5%	99.8%	25.1%	7	T2			
UNIV-LLC3-F2.THR	T4				0.0%	100.0%	70.7%	63	0.0%	100.0%	70.7%	60	T4			
UNIV-LLC3-F2.BOOTSTRAP	T4				26.7%	99.1%	51.9%	63	41.9%	98.6%	41.1%	57	T4			
BIOLEARN.NG	T3				T3				T3				T3			
BIOLEARN.BDE	T3				T3				T3				T3			

Table C3: Detailed results of experiments for YEAST network (4 local causal neighborhoods). See Table C1 for explanation of termination/failure codes T1, T2, T3, and T4.

Appendix D. Assessment of Various Edge Orientation Strategies

To evaluate the accuracy of edge orientation, five orientation methods were tested in two datasets (REGED and ECOLI). All orientation experiments were conducted on the unoriented skeleton discovered by the PC algorithm from observational data. The following five orientation methods were tested:

(1) observational: Edge orientation was determined using constraint-based orientation rules specified in the PC algorithm. This orientation method applied on top of the unoriented PC skeleton is equivalent to the PC algorithm. Notice that some edges may be left unoriented.

(2) experimental: This is a classic orientation approach, and it involves manipulating a variable and assessing its statistical association with the undirected neighbors in order to determine the orientation. For the implementation of this approach, variables with the largest number of undirected neighbors were prioritized for manipulation in order to minimize the number of required experiments (Meganck et al., 2006). Specifically the approach was implemented as follows: (a) Select the vertex with the largest number of undirected neighbors. Denote this variable as X, and its undirected neighbors $Y_1, ..., Y_i, ...Y_n$; (b) Manipulate variable X. (c) For every undirected neighbor Y_i, orient edge as $X \rightarrow Y_i$, if there is a statistically significant association between X and Y_i at $\alpha = 0.05$. Otherwise, orient edge as $Y_i \rightarrow X$; (d) repeat steps (a)-(c) until all edges are oriented.

(3) experimental: For every unoriented edge $X - -Y$in the skeleton, manipulate X and assess the association between X and Y, denoted as A_{XY}. Similarly, manipulate Y and assess the association between X and Y, denoted as A_{YX}. The larger is A_{XY} (or A_{YX}), the stronger is association. If $A_{XY} > A_{YX}$, orient edge as $X \rightarrow Y$, otherwise orient edge as $Y \rightarrow X$;

(4) observational + experimental: apply observational method (1) and orient the rest of the unoriented edges with the experimental method (2);

(5) observational + experimental: apply observational method (1) and orient the rest of the unoriented edges with the experimental method (3).

The results of experiments described above are given in Table D1. The accuracy of orientation is defined as the number of correctly oriented edges divided by the number of correctly inferred edges in the skeleton (i.e. evaluated only with respect to correctly inferred edges by the PC algorithm). In both datasets, the observational orientation had an accuracy that is close to or worse than random (55.2% for REGED and 40.9% for ECOLI). On the other hand, both experimental orientation methods yielded much higher and non-random accuracies up to 100% for REGED dataset and up to 91.2% for ECOLI dataset. Performing observational orientation before experimental orientation reduces the number of experiments as expected, however this also reduces the accuracy. These results indicate that although PC orientation is theoretically sound, experimental orientation methods provide better orientation accuracy.

REGED

Orientation Method	# of edges in the gold-standard	# of edges in the skeleton	# of correctly inferred edges in the skeleton	# of oriented edges in the skeleton	# of correctly inferred edges in the skeleton that are also oriented	# of correctly oriented edges in the skeleton	# of experi-ments	**Accuracy of orientation***
(1) observational	1148	6324	1137	6073	942	520	0	55.2%
(2) experimental	1148	6324	1137	6324	1137	1116	645	98.2%
(3) experimental	1148	6324	1137	6324	1137	1137	1000	100.0%
(4) observational+experimental	1148	6324	1137	6324	1137	712	143	62.6%
(5) observational+experimental	1148	6324	1137	6324	1137	715	336	62.9%

ECOLI

Orientation Method	# of edges in the gold-standard	# of edges in the skeleton	# of correctly inferred edges in the skeleton	# of oriented edges in the skeleton	# of correctly inferred edges in the skeleton that are also oriented	# of correctly oriented edges in the skeleton	# of experi-ments	**Accuracy of orientation***
(1) observational	3632	12091	1660	11964	1595	653	0	40.9%
(2) experimental	3632	12091	1660	12091	1660	1348	1206	81.2%
(3) experimental	3632	12091	1660	12091	1660	1514	1565	91.2%
(4) observational+experimental	3632	12091	1660	12091	1660	718	62	43.3%
(5) observational+experimental	3632	12091	1660	12091	1660	718	152	43.3%

***** Computed only over edges that have been correctly inferred in the skeleton.

Table D1: Comparison of accuracy for various edge orientation methods.

Appendix E. Publicly Available Software Implementations of the Core Methods

Algorithm	Implementation	Link to Publicly Available Software
ODLP*	Matlab	http://ccdlab.org/odlp.html
ALCBN	-	Can be requested from the authors of Meganck et al., 2006
HE-GENG	R	http://www.math.pku.edu.cn:8000/people/view.php?uid=heyb&showdetail=1
LLC	R	LLC1: Can be requested from the authors of Eberhardt et al., 2010 LLC2: https://docs.google.com/file/d/0B7pSUZzmhZ33VnZjdG8xaUVIZDg/edit?pli=1 LLC3: https://docs.google.com/file/d/0B7pSUZzmhZ33b1Zfb3l6XzMwQzQ/edit
BIOLEARN	Java	http://www.c2b2.columbia.edu/danapeerlab/html/biolearn.html

Table E1: Publicly available software implementation of different algorithms

Appendix F. Description of the TIE* and iTIE* algorithms

The TIE* and iTIE* algorithms are described in detail in (Statnikov et al., 2013). Before we review the algorithms below, we note that TIE* and iTIE* were originally introduced for discovery of all Markov boundaries of the target variable T. However, TIE* is also suitable for discovery of all local causal pathways of T consistent with the data when it is used with the Markov boundary induction algorithm Semi-Interleaved HITON-PC; see proof of Theorem 1 in Appendix G for discussion. Similarly, iTIE* which is derived by modifying Semi-Interleaved HITON-PC can be also used for discovery of all local causal pathways of T consistent with the data. When there is no multiplicity of local causal pathways, TIE* and iTIE* will be equivalent to Semi-Interleaved HITON-PC and will output all and only members of the true local causal pathway of T. When the multiplicity is present, the union of Markov boundaries output by TIE* or iTIE* (i.e., all local causal pathways of T consistent with the data) will contain all variables that constitute the true local causal pathway of T and other variables that contain equivalent information about T.

Next, we present the generative TIE* algorithm. This generative algorithm describes a family of related but not identical algorithms which can be seen as instantiations of the same broad algorithmic principles. The pseudo-code of the TIE* generative algorithm is provided in Figure F1. On input TIE* receives (i) a dataset $\mathbb{D}$ (a sample from distribution $\mathbb{P}$) for variables $\boldsymbol{V}$, including a target variable T; (ii) a single Markov boundary induction algorithm $\mathbb{X}$; (iii) a procedure $\mathbb{Y}$ to generate datasets $\mathbb{D}^e$ from the so-called embedded distributions that are obtained by removing subsets of variables from the full set of variables $\boldsymbol{V}$ in the original distribution $\mathbb{P}$; and (iv) a criterion $\mathbb{Z}$ to verify Markov boundaries of T. The inputs $\mathbb{X}, \mathbb{Y}, \mathbb{Z}$ are selected to be suitable for the distribution at hand and should satisfy admissibility rules stated in (Statnikov et al., 2013) for correctness of the algorithm. The algorithm outputs all Markov boundaries of T that exist in the distribution $\mathbb{P}$.

To further facilitate understanding of the TIE* algorithm, we provide in Figure F2 a concrete and specific instantiation of TIE*. Finally, we present in Figure F3 the algorithm iTIE*.

Generative algorithm **TIE***

Inputs:

- dataset $\mathbb{D}$ (a sample from distribution $\mathbb{P}$) for variables ***V***, including a target variable *T*;
- Markov boundary induction algorithm $\mathbb{X}$;
- procedure $\mathbb{Y}$ to generate datasets from the embedded distributions;
- criterion $\mathbb{Z}$ to verify Markov boundaries of *T*.

Output: all Markov boundaries of *T* that exist in $\mathbb{P}$.

1. Use algorithm $\mathbb{X}$ to learn a Markov boundary ***M*** of *T* from the dataset $\mathbb{D}$ for variables ***V*** (i.e., in the original distribution $\mathbb{P}$)
2. Output ***M***
3. Repeat
4. Use procedure $\mathbb{Y}$ to generate a dataset $\mathbb{D}^e$ from the embedded distribution by removing a subset of variables ***G*** from the full set of variables ***V*** in the original distribution (also denoted as $\mathbb{D}(\boldsymbol{V} \setminus \boldsymbol{G})$).
5. Use algorithm $\mathbb{X}$ to learn a Markov boundary $\boldsymbol{M}_{new}$ of *T* from the dataset $\mathbb{D}^e$
6. If $\boldsymbol{M}_{new}$ is a Markov boundary of *T* in the original distribution according to criterion $\mathbb{Z}$, output $\boldsymbol{M}_{new}$
7. Until all datasets $\mathbb{D}^e$ generated by procedure $\mathbb{Y}$ have been considered.

Figure F1: TIE* generative algorithm

An example of instantiated algorithm **TIE***

Inputs: dataset $\mathbb{D}$ (a sample from distribution $\mathbb{P}$) for variables ***V***, including a target variable *T*.

Output: all Markov boundaries of *T* that exist in $\mathbb{P}$.

1. Use algorithm Semi-Interleaved HITON-PC to learn a Markov boundary ***M*** of *T* from the dataset $\mathbb{D}$ for variables ***V*** (i.e., in the original distribution $\mathbb{P}$)
2. Output ***M***
3. Repeat
4. Generate a dataset $\mathbb{D}^e = \mathbb{D}(\boldsymbol{V} \setminus \boldsymbol{G})$ from the embedded distribution by removing from the full set of variables ***V*** in the original distribution the smallest subset ***G*** of the so far discovered Markov boundaries of *T* such that:
   (i) ***G*** was not considered in the previous iterations of this step, and
   (ii) ***G*** does not include any subset of variables that was previously removed from ***V*** to yield a dataset $\mathbb{D}^e$ when $\boldsymbol{M}_{new}$ was found not to be a Markov boundary of *T* in the original distribution (per step 6)
5. Use algorithm Semi-Interleaved HITON-PC to learn a Markov boundary $\boldsymbol{M}_{new}$ of *T* from the dataset $\mathbb{D}^e$ (i.e., in the embedded distribution)
6. If $T \perp \boldsymbol{M} \mid \boldsymbol{M}_{new}$, then $\boldsymbol{M}_{new}$ is a Markov boundary of *T* in the original distribution and it is output by the algorithm
7. Until all datasets $\mathbb{D}^e$ generated in step 4 have been considered.

Figure F2: An example of instantiated TIE* algorithm.

Algorithm **iTIE***

Input: dataset $\mathbb{D}$ (a sample from distribution $\mathbb{P}$) for variables ***V***, including a target variable *T*.

Output: multiple Markov boundaries of *T* that exist in $\mathbb{P}$.

Phase I: Forward

1. Initialize Θ with an empty set
2. Initialize ***M*** with an empty set
3. Initialize the set of eligible variables ***E*** ← ***V*** \ *T*
4. Repeat
5. *Y* ← $\text{argmax}_{X \in \boldsymbol{E}}$ Association(*T*, *X*)
6. ***E*** ← ***E*** \ *Y*
7. If there is no subset ***Z*** ⊂ ***M*** such that $T \perp Y \mid \boldsymbol{Z}$ then
8. ***M*** ← ***M*** ∪ *Y*
9. Else if ***Z*** exists and the following relations hold: $T \not\perp Y$, $T \not\perp \boldsymbol{Z}$, $T \perp \boldsymbol{Z} \mid Y$
10. Record in Θ that *Y* and ***Z*** contain equivalent information with respect to *T*
11. Until ***E*** is empty

Phase II: Backward

12. For each *X* ∈ ***M***
13. If there is a subset ***Z*** ⊂ ***M*** \ *X* such that $T \perp X \mid \boldsymbol{Z}$ then
14. ***M*** ← ***M*** \ *X*

Phase III: Construction of multiple Markov boundaries

15. Compute the Cartesian product of target information equivalency relations for subsets of ***M*** that are stored in Θ to construct multiple Markov boundaries of *T*
16. Output multiple Markov boundaries of *T*

Figure F3: iTIE* algorithm

Appendix G. Proof of Correctness of ODLP

Theorem 1 *ODLP is sound under the following sufficient assumptions: (i) TIE near-faithfulness (as a relaxation of local adjacency faithfulness to allow for target information equivalency relations); (ii) causal Markov condition; (iii) local causal sufficiency; (iv) acyclicity of the data-generative graph; and (v) correctness of statistical decisions.*

Proof First, we remind the readers that under DAG-faithfulness, the Markov boundary is unique and consists of children, parents, and spouses of T. i.e., the Markov boundary contains all members of the local causal pathway of T (consisting of parents and children of T), plus spouses that are not children of T. The latter spouses are marginally or conditionally independent of T unlike members of the local causal pathways of T. Under DAG-faithfulness, the Semi-Interleaved HITON-PC algorithm can discover all members of the local causal pathway of T (Aliferis et al., 2010a,b). However under TIE near-faithfulness, this algorithm will output a local causal pathway consistent with the data, which may or may not contain parents and children of T.

We have previously established that an admissible instantiation of the generative algorithm TIE* can correctly discover all Markov boundaries of the target variable T (see Theorem 10 in (Statnikov et al., 2013)). When TIE* is instantiated with the Markov boundary inducer Semi-Interleaved HITON-PC, it will identify in step 1 all local causal pathways of T consistent with the data (Statnikov et al., 2013). The latter requires that members of all local causal pathways consistent with the data are marginally and conditionally dependent on T (except for violations of the intersection property that lead to equivalence relations), which is satisfied given assumptions of this theorem, in particular TIE near-faithfulness. Therefore, all members of the true local causal pathway will be contained in the output of TIE* in step 1.

Similarly, it can be shown that iTIE* will identify in step 1 all local causal pathways consistent with the data (and therefore all members of the true local causal pathway) given assumptions of this theorem and an additional requirement that all equivalence relations in the underlying distribution follow from equivalence relations of individual variables. The latter requirement is one of sufficient assumptions for iTIE* correctness (Statnikov et al., 2013).

Before we proceed with the remainder of the proof, we examine the contents of equivalence clusters formed in step 3. Given three types of variables of interest (causes, effects, and passengers) there are the following options for contents of the cluster: (1) causes; (2) causes and effects; (3) causes and passengers; (4) causes, effects and passengers; (5) effects; (6) effects and passengers; and (7) passengers. It can be shown by examples that options (1)-(5) are possible and consistent with assumptions of this theorem. On the other hand, options (6) and (7) cannot take place in the settings of this theorem.

Next we prove correctness of identification of effects, direct effects, other effects ("other effects" are effects that are not identified as direct effects, they could be indirect effects or both direct and indirect effects at the same time), causes, direct causes, other causes ("other causes" are causes that are not identified as direct causes, they could be indirect causes or both direct and indirect causes at the same time), and passengers within the variable set $\boldsymbol{V}$, which is the union of all variables that participate in the local causal pathways

of T consistent with the data. Given that all members of the true local causal pathway are contained in the set $\boldsymbol{V}$, the correct identification of direct effects and direct causes within the set $\boldsymbol{V}$ implies that ODLP is sound.

Identification of effects and direct/other effects: Based on the assumption of correctness of statistical decisions and the definition of causation, all effects of T are correctly identified by performing an experiment on T (step 4) and considering as effects all variables $\boldsymbol{E} \subseteq \boldsymbol{V}$ that change as a result of that experiment (step 5). Identification of direct/other effects is performed within the subset E. We distinguish here three cases:

1. An equivalence cluster contains one variable X, which is an effect (step 9.a). Then X has to be a direct effect. Otherwise, based on causal Markov condition and correctness of statistical decisions, X will not belong to $\boldsymbol{E}$ because X will be rendered statistically independent of T conditioned on a subset of variables from any local causal pathway of T consistent with the data during execution of TIE* in step 1.

2. An equivalence cluster contains multiple variables, out of which only one variable X (effect) has neither been identified yet as other effect nor as direct effect and all other effect variables have been identified as other effects (step 9.a). Then, similarly to the previous case, X has to be a direct effect. Otherwise, a cluster will only have an indirect but no direct effect which cannot happen based on the assumptions of this theorem and the methodology of constructing equivalence clusters by utilizing TIE* in steps 1-3.

3. An equivalence cluster contains multiple variables, out of which two or more effect variables have neither been identified as other effects nor as direct effects. The algorithm proceeds to execution of steps 9.b-9.d, whose correctness follows from the definition of causation and the assumption of correctness of statistical decisions.

Identification of causes and direct/other causes: Since we have already identified the set of effects $\boldsymbol{E}$, identification of causes (and direct/other causes) is performed within the set of variables $\boldsymbol{V} \setminus \boldsymbol{E}$. We distinguish here three cases:

1. An equivalence cluster contains one unmarked variable X (step 6.a). Since X is unmarked, it is not an effect. Then X has to be a direct cause. Otherwise, based on causal Markov condition and correctness of statistical decisions, X will not belong to $\boldsymbol{V} \setminus \boldsymbol{E}$ because X will be rendered statistically independent of T conditioned on a subset of variables from any local causal pathway of T consistent with the data during execution of TIE* in step 1.

2. An equivalence cluster contains multiple variables, out of which only one variable X has not been marked yet and all other variables have been identified as passengers and/or effects (step 6.a). Again, since X is unmarked, it is not an effect. Then, similarly to the previous case, X has to be a direct cause. Otherwise, a cluster will either have only effects and passengers or effects, passengers, and an indirect cause. None of these cases can happen based on the assumptions of this theorem and the methodology of constructing equivalence clusters by utilizing TIE* in steps 1-3.

3. An equivalence cluster contains multiple variables, out of which two or more variables have not been marked yet. The algorithm proceeds to execution of steps 6.b-6.d, whose correctness follows from the definition of causation and the assumption of correctness of statistical decisions.

Identification of passengers: Based on the assumption of correctness of statistical decisions and the definition of causation, passengers are correctly identified in step 6.d. More

specifically, all variables marked as passengers in that step have been previously unmarked (and therefore are not effects of T) and are not on the causal path to T (and therefore are not causes of T). ■

Appendix H. More on ODLP's Experimental Strategy and its Efficiency

Consider an example network shown in Figure H1.a. Variables A, B, C, D, and E contain equivalent information about the target variable T and cannot be distinguished with observational data. Without any prior knowledge about the causal role of A, B, C, D, and E, we will first need to manipulate T to determine that none of the above 5 variables is an effect of T. Therefore, they can be either causes or passengers. If we manipulate C, we will realize that D and E change but T does not change due to manipulation of C. Therefore, C, D, E are all passengers and we do not need to manipulate D and E (we saved 2 experiments). Next we manipulate A and observe that it leads to changes in T (and B, C, D, and E) and thus it is a cause of T. Finally, we can manipulate B and observe that it leads to changes only in T and thus it is a direct cause. So, in total we performed 4 experiments (manipulate T, C, A, and B in order). However, if we did not choose C early on for manipulations, we could end up doing up to 6 experiments (manipulate T, E, D, C, A, and B in order) to identify the local causal pathway. In fact, it is not possible to conduct fewer than four single-variable experiments in this example, and thus the sequence of experiments T, C, A, B is optimal. The only problem is that we do not know the graphical structure when we perform experiments, and thus we need to resort to heuristics to manipulate first variables that are likely to yield savings in experiments (step 6.b of the ODLP algorithm; see Figure 3).

Consider another example network shown in Figure H1.b. Variables A, B, C, D, E, F, and J contain equivalent information about the target variable T and cannot be distinguished with observational data. Assume that we have already manipulated T, A, and B, and now we are deciding what variable to manipulate next. Manipulation of T, A, and B provided us with partial information on topological (causal) order of variables. Specifically, we know that (i) no variable is downstream of T (from manipulating T), (ii) B, C, D, E, F, J, and T are downstream of A (from manipulating A), and (iii) D, E, F, J, and T are downstream of B (from manipulating B). As discussed in the text, one possibility is to use a partial network-based heuristic that chooses a variable that has the highest topological order relative to T. As established from constraints learned from experimental data, variable C has the highest topological order and has not been manipulated yet. Manipulation of C allows to immediately identify the local causal pathway because D, E, F, and J will change and T will not change due to manipulation of C, thus C, D, E, F, and J are all passengers. In summary we conducted 4 experiments, while alternative strategies will take up to 8 experiments. To see the expected efficiency of the above heuristic function, we can revisit this example and assume that we do not have knowledge to manipulate A and B first. In this case, we will identify the local causal pathway in 4 experiments with probability 6.67% using the above heuristic and with probability 2.86% without the heuristic and performing random selection of variables for manipulation (in step 6.b of the ODLP algorithm; see Figure 3).

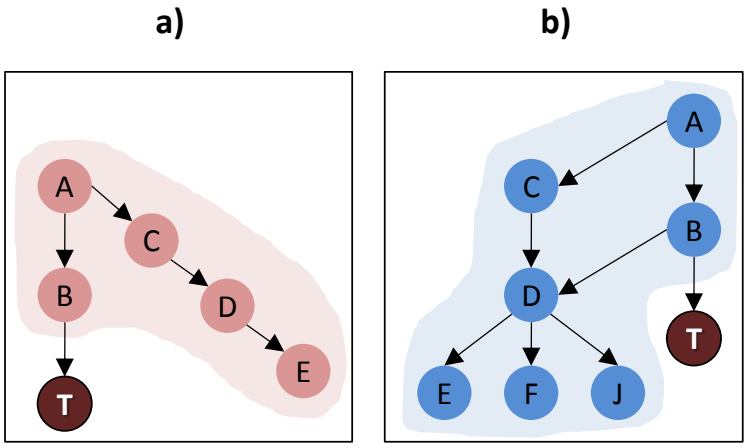

Figure H1: Two causal networks used to illustrate ODLPs experimental strategy and its efficiency. Variables are shown with circles, and edges represent direct causal influences. The target variable is T. Variables that are shown with the same color contain the same information about the target (they are target information equivalent).

References

Constantin F Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D Koutsoukos. Local causal and markov blanket induction for causal discovery and feature selection for classification part i: Algorithms and empirical evaluation. *The Journal of Machine Learning Research*, 11:171–234, 2010a.

Constantin F Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D Koutsoukos. Local causal and markov blanket induction for causal discovery and feature selection for classification part ii: Analysis and extensions. *The Journal of Machine Learning Research*, 11:235–284, 2010b.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 289–300, 1995.

Yoav Benjamini and Daniel Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics*, pages 1165–1188, 2001.

David Maxwell Chickering. Optimal structure identification with greedy search. *The Journal of Machine Learning Research*, 3:507–554, 2003.

Diego Colombo and Marloes H Maathuis. Order-independent constraint-based causal structure learning. *The Journal of Machine Learning Research*, 15(1):3741–3782, 2014.

Povilas Daniusis, Dominik Janzing, Joris Mooij, Jakob Zscheischler, Bastian Steudel, Kun Zhang, and Bernhard Schölkopf. Inferring deterministic causal relations. *arXiv preprint arXiv:1203.3475*, 2012.

Edward R Dougherty and Marcel Brun. On the number of close-to-optimal feature sets. *Cancer Informatics*, 2:189, 2006.

Frederick Eberhardt, Patrik O Hoyer, and Richard Scheines. Combining experiments to discover linear cyclic models with latent variables. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 185–192, 2010.

Clark N. Glymour and Gregory F. Cooper. *Computation, Causation, and Discovery.* AAAI Press ; MIT Press, Menlo Park, California, 1999.

Isabelle Guyon, Constantin F Aliferis, Gregory F Cooper, André Elisseeff, Jean-Philippe Pellet, Peter Spirtes, and Alexander R Statnikov. Design and analysis of the causation and prediction challenge. In *WCCI Causation and Prediction Challenge*, pages 1–33, 2008.

Yang-Bo He and Zhi Geng. Active learning of causal networks with intervention experiments and optimal designs. *Journal of Machine Learning Research*, 9(11), 2008.

Patrik O Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. In *Advances in Neural Information Processing Systems*, pages 689–696, 2009.

Antti Hyttinen, Frederick Eberhardt, and Patrik O Hoyer. Causal discovery for linear cyclic models with latent variables. *on Probabilistic Graphical Models*, page 153, 2010.

Antti Hyttinen, Frederick Eberhardt, and Patrik O Hoyer. Learning linear cyclic causal models with latent variables. *The Journal of Machine Learning Research*, 13(1):3387–3439, 2012.

Dominik Janzing, Joris Mooij, Kun Zhang, Jan Lemeire, Jakob Zscheischler, Povilas Daniušis, Bastian Steudel, and Bernhard Schölkopf. Information-geometric approach to inferring causal directions. *Artificial Intelligence*, 182:1–31, 2012.

Jan Lemeire. *Learning Causal Models of Multivariate Systems and the Value of it for the Performance Modeling of Computer Programs.* PhD thesis, Vrije Universiteit Brussel, 2007.

Daniel Marbach, Thomas Schaffter, Claudio Mattiussi, and Dario Floreano. Generating realistic in silico gene networks for performance assessment of reverse engineering methods. *Journal of Computational Biology*, 16(2):229–239, 2009.

Stijn Meganck, Philippe Leray, and Bernard Manderick. Learning causal bayesian networks from observations and experiments: A decision theoretic approach. In *Proceedings of the Third International Conference on Modeling Decisions for Artificial Intelligence*, MDAI'06, pages 58–69, Berlin, Heidelberg, 2006. Springer-Verlag.

Joshua Menke and Tony R Martinez. Using permutations instead of student's t distribution for p-values in paired-difference algorithm comparisons. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 2, pages 1331–1335. IEEE, 2004.

Joris Mooij, Oliver Stegle, Dominik Janzing, Kun Zhang, and Bernhard Schölkopf. Probabilistic latent variable models for distinguishing between cause and effect. In *Advances in Neural Information Processing Systems*, pages 1687–1695, 2010.

Kevin P Murphy. Active learning of causal bayes net structure, 2001.

Richard E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, illustrated edition edition, April 2003. ISBN 0130125342.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann Publishers, 1 edition, September 1997. ISBN 9781558604797.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, New York, NY, USA, 2nd edition, 2009. ISBN 052189560X, 9780521895606.

Dana Pe'er, Aviv Regev, Gal Elidan, and Nir Friedman. Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, 17, 2001.

Jose M Peña, Roland Nilsson, Johan Björkegren, and Jesper Tegnér. Towards scalable and data efficient learning of markov boundaries. *International Journal of Approximate Reasoning*, 45(2):211–232, 2007.

Joseph Ramsey, Jiji Zhang, and Peter L Spirtes. Adjacency-faithfulness and conservative causal inference. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-2006)*, pages 401–408, 2006.

Thomas Richardson. A discovery algorithm for directed cyclic graphs. In *Proceedings of the Twelfth International Conference on Uncertainty in Artificial Intelligence*, pages 454–461. Morgan Kaufmann Publishers Inc., 1996.

Thomas Richardson and Peter Spirtes. *Automated Discovery of Linear Feedback Models*, chapter 7, pages 254–302. MIT Press, 1999.

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A Lauffenburger, and Garry P Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

Thomas Schaffter, Daniel Marbach, and Dario Floreano. Genenetweaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics*, 27(16):2263–2270, 2011.

Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, and Antti Kerminen. A linear non-gaussian acyclic model for causal discovery. *The Journal of Machine Learning Research*, 7:2003–2030, 2006.

Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, Prediction, and Search*, volume 81. MIT press, 2000.

Alexander Statnikov and Constantin F Aliferis. Analysis and computational dissection of molecular signature multiplicity. *PLoS Computational Biology*, 6(5):e1000790, 2010.

Alexander Statnikov, Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. *Causal explorer: A matlab library of algorithms for causal discovery and variable selection for classification*, volume 2, page 267. Microtome Publishing, 2010.

Alexander Statnikov, Mikael Henaff, Nikita I Lytkin, and Constantin F Aliferis. New methods for separating causes from effects in genomics data. *BMC Genomics*, 13, 2012.

Alexander Statnikov, Jan Lemeir, and Constantin F Aliferis. Algorithms for discovery of multiple markov boundaries. *The Journal of Machine Learning Research*, 14(1):499–566, 2013.

Simon Tong and Daphne Koller. Active learning for structure in bayesian networks. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'01, pages 863–869, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2006a.

Ioannis Tsamardinos, Alexander R Statnikov, Laura E Brown, and Constantin F Aliferis. Generating realistic large bayesian networks by tiling. In *FLAIRS Conference*, pages 592–597, 2006b.

Jianxin Yin, You Zhou, Changzhang Wang, Ping He, Cheng Zheng, and Zhi Geng. Partial orientation and local structural learning of causal networks for prediction. In *WCCI Causation and Prediction Challenge*, pages 93–105, 2008.

Kun Zhang and Aapo Hyvärinen. Distinguishing causes from effects using nonlinear acyclic causal models. In *Journal of Machine Learning Research, Workshop and Conference Proceedings (NIPS 2008 causality workshop)*, volume 6, pages 157–164, 2008.