University of Amsterdam

MSc Artificial Intelligence
Master Thesis

Order-Based Causal Analysis of Gene Expression Data

---

by
Silvan de Boer
12158054

November 20, 2020

48 ECTS
November 2019 - November 2020

Supervisor:
Prof Dr J Mooij

Assessor:
Dr P Forré

Informatics Institute

Abstract

Artificial intelligence (AI) is a field of science that attempts to automate human intelligence. In the last decades, the statistical approach to this automation has made tremendous progress, especially with deep learning in subdomains like computer vision and natural language processing. However, the statistical approach is starting to reach limits.

One limitation of purely statistical AI is that it has no understanding of cause and effect. On the intersection of statistical and symbolic AI, the field of causality offers a framework to model causal assumptions. This framework allows for a formal analysis of cause and effect in data.

The focus of this thesis is to infer causal relations in a gene perturbation dataset [Kemmeren et al., 2014]. This dataset contains measurements of the expression of genes in yeast bacteria, under normal circumstances and when a gene is knocked out. The dataset is sparse and high-dimensional, which makes the task particularly challenging.

We attempt to improve a simple and efficient inference algorithm called Local Causal Discovery (LCD), because its performance is near state-of-the-art on this dataset [Versteeg and Mooij, 2019]. The method relies on an exogenous context variable to predict ancestral relations.

Our hypothesis is that there is some implicit causal order among the genes, which can be used to inform the context and improve the performance of LCD. We extensively investigate algorithms to estimate such order. An algorithm is chosen that uses TrueSkill [Herbrich et al., 2007], which was originally developed to rate a skill level of players based on game outcomes.

Although the order estimation seems appropriate, we do not succeed to improve the performance of the LCD method. We thoroughly analyse the properties of the method and compare it with the original LCD version. Directions for future research are suggested to help further develop causal inference on this challenging dataset.

Contents

1 Introduction

1.1 Positioning the Thesis in the Field of AI

Artificial intelligence (AI) is a field of science that attempts to automate human intelligence. This comprises a philosophical component. What do we consider to be human intelligence? In fact, this notion changes as AI advances. Once, chess grandmasters were attributed superior intelligence. Since the world champion Garry Kasparov was beaten by a computer in 1997, mastering chess has downgraded to learning an impressive skill for a human. This leaves us to wonder whether all human intelligence can be automated, and what even distinguishes intelligence from other skills if it can be automated.

Psychology and biology are another dimension of AI. Knowledge of the brain can help automate its functions. The neurological structure of the brain has proven to be a very productive metaphor for the neural network, a widely used mathematical model that is the cornerstone of deep learning. On the side of psychology, studying human learning and problem solving may inform the design of AI algorithms. Reinforcement learning is full of metaphors of agents in a world learning from their experience.

Finally, to apply artificial intelligence in the real world, we use computers to efficiently execute algorithms and perform tasks. This requires information technology to represent human knowledge in a machine, and mathematics to formulate algorithms such that machines can execute them.

The Computational Domain of AI

Two schools are typically distinguished in this computational domain of AI, aptly described by Van Harmelen and Teije [2019]. The symbolic school is characterized by algorithms using a model of the world that is discrete, compositional, interpretable, and homologous to the structure of the thing it models. Algorithms use deduction to draw conclusions from a model.

The statistical school is typified by functions that are model-free, and constructed using induction. Observations from the world are collected as a dataset. Algorithms make generalizations and discover patterns in the data, which are used to fit a function. This function can then be used to make predictions. The field of machine learning and specifically deep learning is mostly contained in the statistical school.

In the last decade, statistical AI has had unbelievable successes. Increased computational power and a growing body of data allowed deep learning models to shatter the state-of-the-art in AI subfields like computer vision and natural language processing. These rapid developments have inspired incredible optimism. Grace et al. [2018] surveyed AI researchers at some top conferences about the future of AI. These researchers predicted that with 50% chance, machines will be better and more cost-effective than humans at every job within 120 years. That includes building houses, creating inspirational documentaries,

governing countries, and doing AI research.

We find these predictions naive, but do not reject the possibility that technology will ever reach this point. Whatever be the timeline, most would agree that these goals cannot be reached with deep learning alone. Deep learning has already reached theoretical limits that can no longer be overcome with more compute or data. Problems such as adaptability and explainability renewed interest in the symbolic school, and sparked interest in combining the schools.

Causality

One limitation of purely statistical AI is that it has no understanding of cause and effect. Pearl [2009] is an influential researcher in the field of causality. In a recent article, he describes a three-level hierarchy of information [Pearl, 2019]. Each layer can answer a certain type of question, and a higher layer subsumes the lower layers as it can answer their questions as well. We shortly discuss the layers to exemplify this limitation of statistical AI, and to show how the methods in this thesis play into this perspective. Since the Covid-19 pandemic has dominated the news while this thesis was written, we will use it for examples of each layer.

The first level is concerned with associations. This is formalized in statistics with conditional probabilities $\mathbb{P}(X = x|Y = y)$. A typical question would be: "What is the chance that I am infected with Covid ($X = x$), given that I live in Utrecht ($Y = y$)?" Questions in this layer can be answered from observational data alone. Statistical AI is limited to this layer.

The second level is about interventions. This layer cannot be formally described by traditional statistics. Pearl [2009] developed a formalism that contains a new do-operator, and a do-calculus to compute probabilities such as $\mathbb{P}(X = x|do(Y = y), Z = z)$. This formalism allows us to handle questions like "What will happen to the number of cases per capita ($X = x$) in Delft ($Z = z$) if we introduce a lock-down ($do(Y = y)$)? This question cannot be answered with observational data alone. One issue in this case is that Delft has never been in lock-down. The algorithms discussed in this thesis are concerned with this second layer of information.

For completeness, we describe the third level. This level handles counterfactuals, which question the probability of some observation had the circumstances been different, denoted by $\mathbb{P}(y_x|x', y', do(Z = z), W = w)$. A typical question is "What is the chance that I would have been infected with Covid, if I had not worn a face mask in the train yesterday (y_x)? I know that I actually did wear the mask (x') and that I am not infected (y'). The train went to Bovenkarspel ($W = w$) and all other people wore a mask because the government made it mandatory ($do(Z = z)$)." To answer such questions, data alone cannot be enough. For example, it is logically impossible to have infection data of people that really wore a mask, but then actually did not.

1.2 Topic of the Thesis

Discovering the Gene Regulatory Network

In this thesis we investigate a very practical prediction task, that heavily relies on making a distinction between cause and effect. Kemmeren et al. [2014] created a dataset from experiments on yeast bacteria. Using microarray technology, they measured the expression level of over six thousand genes in this bacteria. The expression level indicates how active a gene is. Genes have effect on each other. One gene may up- or down-regulate another gene, meaning that its activity increases or decreases the activity of the other gene. We can construct a regulatory network by representing all relations between all genes in a graph. In order to understand the functioning of the yeast bacteria, we wish to gain knowledge about this network. In this thesis, we investigate a method to predict the effects of each gene.

To predict if one gene affects another, it is not enough to know if they are statistically associated. If we only know that two genes X and Y are associated, it may be that X has effect on Y, Y has effect on X, some other gene has an effect on both X and Y, or some combination is the case. This brings us to the realm of causality, because we need to define causal assumptions that are required to distinguish these cases.

In fact, the dataset provides a good first assumption. 262 experiments are reported in which one gene was knocked out (made inactive), which we can interpret as an intervention. Besides using these data points as an important source of information, we also use it to evaluate the predictions.

The dataset poses three important challenges, that make it interesting for this thesis. Because the number of genes is large, complete algorithms are computationally expensive and almost infeasible. Moreover, there is very little data. Specifically, for each intervention there is only one data point, and the number of genes is larger than the total number of data points, including purely observational data. Lastly, evaluation is not trivial. In the biology literature, there is only some consensus about a relatively small number of relations.

We take the work of Versteeg and Mooij [2019] as a starting point, since they report a state-of-the-art causal inference algorithm on this dataset (ICP), and an incomplete efficient algorithm that approaches a similar performance, called Local Causal Discovery (LCD). We focus on LCD, because it is more efficient. LCD constructs an external context variable that represents background knowledge that can be used to infer causal relations. Versteeg and Mooij [2019] construct this in the most straightforward way, encoding if a data point is pure observation or whether an intervention took place. We try to make the context variable more informative.

We could define a hierarchy among the genes in which causes precede effects. We hypothesize that this implicit causal order can be used to inform the LCD context variable. A knock-out on a gene that is earlier in the order, is more likely to affect some gene than one later in the order.

This poses a challenge of estimating this causal order in the genes. We investigated this problem extensively.

Moreover, when we predict the effects of some gene, we use the intervention data of that gene to evaluate the predictions. This means we cannot use it to infer its position in the order. We require a different estimation method to infer the position of a single gene in an existing order, and propose a straightforward method for this.

The combination of these two estimation methods yields a context variable that we use for LCD in our experiment. We call the method containing these three steps *order-based local causal discovery*, and analyse its performance on the gene perturbation dataset.

1.3 Structure of the Thesis

We start this thesis in Chapter 2 with a description of the modeling framework, and how causal assumptions are encoded in this framework. The chapter is concluded with a concise review of causal inference methods. The dataset is described and analysed in Chapter 3.

Following this, we develop and justify the order-based LCD method. Chapter 4 details the experiments and analysis that determine how we estimate variable order. Order-based LCD is described in detail in Chapter 5. The method requires us to estimate the position of a test variable in the order. A short explanation and analysis is included of our position estimation method.

Chapter 6 describes the experiment in which order-based LCD is applied to the Kemmeren et al. [2014] dataset. The implementation of our method and the baselines is also shortly discussed. The results and analysis of this experiment is described in Chapter 7, followed by our conclusions and suggestions for future work in Chapter 8.

2 Background

2.1 Modelling Framework

Structural Causal Model

Throughout this thesis we will assume that the data is generated by a Structural Causal Model (SCM) $\mathcal{M}$. This modelling framework is widely used in the field of causality, and is very flexible (see e.g. Pearl [2009] and Peters et al. [2017] for the variety of methods).

A distinction is made between endogenous variables and exogenous variables. Endogenous variables are known, either by measurement (data $\boldsymbol{X}$) or by design of some inference method (e.g. context variables $\boldsymbol{C}$ in LCD). They are represented by an index set $\mathcal{I}$. Exogenous variables are latent, a typical example is noise variables $\boldsymbol{N}$. Exogenous variables are represented by an index set $\mathcal{J}$.

The causal mechanism $\boldsymbol{f}$ of a SCM is a function that describes how all variables relate to each other. It maps a product space of all variables to a product space of the endogenous variables.

The components of the causal mechanism f_i usually do not depend on all variables, but rather on a small subset that we call the parents $\text{PA}(i)$ of variable X_i. The augmented graph $\mathcal{H}_\mathcal{M}$ represents these child-parent relations with directed edges between variable nodes.

The exogenous variables are modelled with a product probability measure $\mathbb{P}_{\boldsymbol{\mathcal{E}}}$, since their values are not measured. Data can then be sampled from a SCM by sampling from this measure and (iteratively) applying the functions f_i to compute the values of the endogenous variables.

Usually, an incomplete definition of SCMs suffices, consisting of the structural equations of the endogenous variables and the density function of the exogenous variables, indicated below with $\boldsymbol{X}$ and $\boldsymbol{E}$ respectively:

$$\mathcal{M} : \begin{cases} X_i &= f_i(\boldsymbol{X}_{\text{PA}(i)\cap\mathcal{I}}, \boldsymbol{E}_{\text{PA}(i)\cap\mathcal{J}}) \\ p_{\boldsymbol{E}} &= \prod_{j\in\mathcal{J}} p_{E_j} \end{cases}$$

In a practical setting we are unable to infer the real structure of the exogenous variables. A useful graphical representation of a SCM is the graph $\mathcal{G}_\mathcal{M}$, which is an abstraction of the augmented graph $\mathcal{H}_\mathcal{M}$. Only the endogenous variables are nodes in this graph. The relations among endogenous variables are still represented by directed edges ($i \to j$). Variables that are confounded by an exogenous variable (i.e. share an exogenous ancestor) are connected with a bidirected edge[1] ($i \leftrightarrow j$).

[1] The same representation of confounding is used when we marginalize over a subset of endogenous variables.

Causal Assumptions and Interventional Data

If one observes two variables X and Y, and measures a dependence among them, it is impossible to say if X causes Y or the other way around. More formally, one cannot infer the causal direction from a probability measure $\mathbb{P}_{\{X,Y\}}$ alone. This is why we have to rely on *causal assumptions* [Pearl, 2009]. Some common assumptions are discussed in Section 2.2. Section 2.3 describes inference methods that rely on these assumptions.

One assumption particularly relevant to this thesis is related to the method of data acquisition. A distinction is made between *observational data* and *interventional data*. Observational data is gathered without interference with the system. We assume that there is an underlying SCM, and every data point is a sample from it. The sampling distribution approximates the observational distribution $\mathbb{P}_{\boldsymbol{X}}$. It is theoretically impossible to infer any causal statements without further assumptions.

Interventional data is gathered while we interfere with the system. Every data point is measured while an *intervention* is performed. Formally, this intervention is modelled as a manipulation of the causal mechanism $\boldsymbol{f}$ subject to some constraints or assumptions. This may render the causal inference problem theoretically possible.

A concrete example is the *perfect intervention*. A perfect intervention sets a variable X_i to a fixed value ξ_i, denoted as $\mathrm{do}(X_i = \xi_i)$. This removes all the dependence of X_i on its parents $\mathrm{PA}_{\mathcal{H}}(i)$. The adapted SCM induces a different, interventional distribution $\mathbb{P}_{\boldsymbol{X}|\mathrm{do}(X_i=\xi_i)}$. Pearl [2009] developed a do-calculus that can be used to make causal inferences from observational and intervenional data. As a simple example, take a system of two variables that are related as $X \to Y$. From the observational data we only know that X and Y are dependent. However, if we have access to distributions $\mathbb{P}_{\{X,Y\}|\mathrm{do}(X=x)}$ and $\mathbb{P}_{\{X,Y\}|\mathrm{do}(Y=y)}$ we can see that intervening on Y does not affect X, whereas intervening on X does affect Y. We conclude that X causes Y.

Markov Property

The Markov Property is a very common assumption that links the SCM to conditional independence relations (CIRs) in the data. The property follows from the definition of the SCM, so it does not add a restriction to our modelling.

The notion of *d-separation* is used to infer CIRs. We say that two variables X and Y are d-separated by a conditioning set of variables $\boldsymbol{C}$, if all walks in $\mathcal{G}_{\mathcal{M}}$ from X to Y are d-blocked by $\boldsymbol{C}$. This is denoted as $X \perp^d_{\mathcal{G}} Y \,|\, \boldsymbol{C}$. On each walk we will consider if the variables are a *collider*, that is: if the adjacent edges of the walk point towards it ($... \to Z \leftarrow ...$). A walk is *d-blocked* in three cases:

1. X or Y are in $\boldsymbol{C}$.
2. The walk contains a non-collider Z that is in $\boldsymbol{C}$.

3. The walk contains a collider Z that is not in $\boldsymbol{C}$, and none of its descendents are in $\boldsymbol{C}$ either.

Consider the graph in Figure 1. By case 1, X_1 blocks the walk from X_1 to X_3. X_2 blocks this walk if it is in the conditioning set $\boldsymbol{C}$ by case 2. According to case 3, the walk from X_2 to X_4 is blocked if neither X_3 nor X_5 are in $\boldsymbol{C}$.

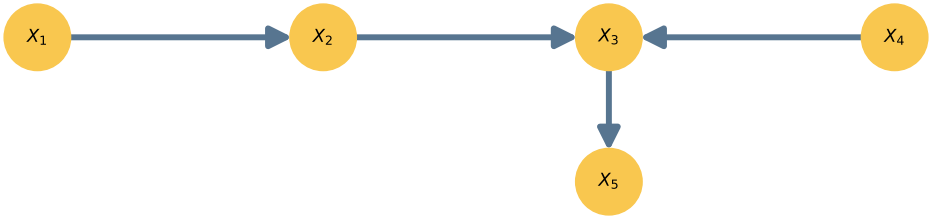

Figure 1: Graph of five random variables.

The Global Markov Property links d-separation to conditional independence:

$$\boldsymbol{A} \overset{d}{\underset{\mathcal{G}}{\perp}} \boldsymbol{B} \,|\, \boldsymbol{C} \implies \boldsymbol{A} \underset{\mathbb{P}_{\boldsymbol{X}}}{\perp\!\!\!\perp} \boldsymbol{B} \,|\, \boldsymbol{C}$$

The Markov Property with d-separation is only valid for SCMs with an acyclic graph.[2] A generalization of d-separation was developed by Forré and Mooij [2017] that applies to the cyclic case as well under certain assumptions.

Different graphs may satisfy the same set of d-separations. Therefore, some observational distribution $\mathbb{P}_{\boldsymbol{X}}$ may satisfy the Global Markov Property with respect to different graphs. The set of graphs that induces the same observational distribution as some graph $\mathcal{G}$ is called its Markov Equivalence Class $\text{MEC}(\mathcal{G})$. For acyclic graphs without latent confounding, it is shown that two graphs are equivalent if they have the same skeleton and immoralities [Verma and Pearl, 1991]. The skeleton is the set of edges when we disregard their direction, an immorality is a local structure $A \to B \leftarrow C$ in which A and C are not directly connected.

Causal inference methods that use only CIRs in observational data cannot infer more than the MEC of the graph, because this is all the information that is in the CIRs in the observational distribution $\mathbb{P}_{\boldsymbol{X}}$.

Causal Inference Tasks

Different tasks can be distinguished within causal inference. In this thesis we focus on methods that infer parts of the augmented graph $\mathcal{H}$. These methods are not directly used to infer quantitative causal effects. However, the do-calculus of Pearl [2009] can in many cases be used to combine measured

[2] It is also valid for some restricted cases of cyclic SCMs, cf. Forré and Mooij [2017].

conditional distributions and a (partially) infered graph to make quantitative statements.

Generally, we can distinguish between global and local inference methods. Global methods aim to infer as much as possible from graph $\mathcal{G}$. If the data is observational, these methods infer (an instance of) the MEC. An example is the SGS algorithm [Spirtes et al., 2000], which naively checks the CIRs among all combinations of variables to infer the MEC. Interventional data enables some methods to infer more features of the graph, like Greedy Interventional Equivalence Search by Hauser and Bühlmann [2012] that infers a so-called interventional MEC.

Local methods are specialized to find only some elements of the graph, usually trading completeness for efficiency. Commonly, these methods find some direct or ancestral causal relations (directed paths in $\mathcal{G}$), like Local Causal Discovery [Cooper, 1997] which tests for one specific pattern of CIRs among three variables.

2.2 Principles of Causal Inference

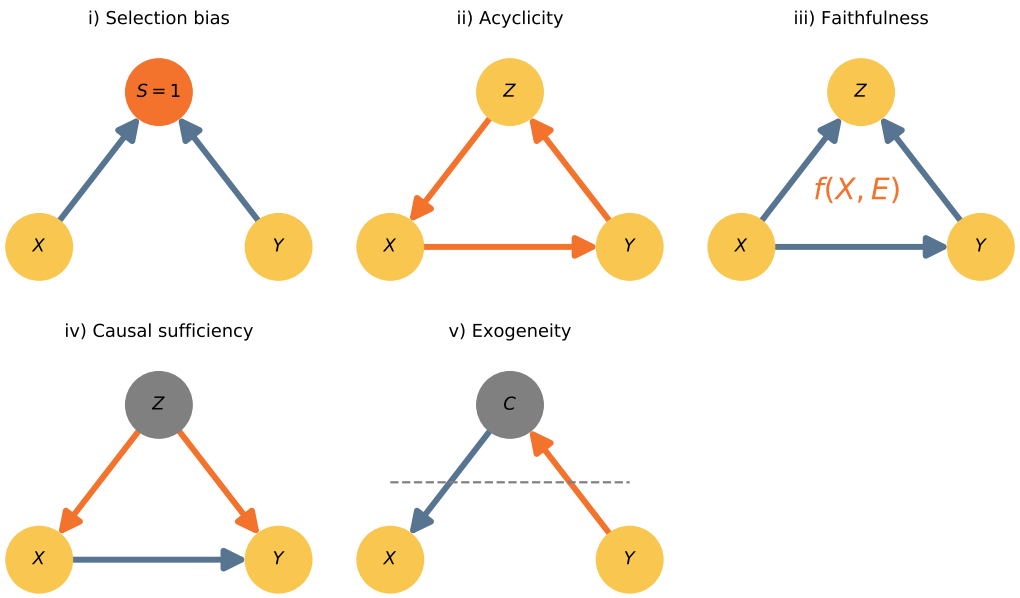

Figure 2: Violations of causal assumptions indicated by orange nodes or edges. Violation of faithfulness (iii) depends on the functional dependence between the variables, for example if $X \perp\!\!\!\perp Z$.

Now that the general modelling framework of the SCM is established, it is time to define the most important additional assumptions that enable many causal inference methods. These assumptions are generally restrictions on the SCM.

Reichenbach's common cause principle

An important insight in causal inference is Reichenbach's common cause principle [Reichenbach, 1956]. It states that correlation is always the result of some causation. When two variables correlate, either one causes the other, or there is a third variable causing both.

This is quite a strong assumption. It relies on i.i.d. sampling of the data, and a good approximation of the probability distribution. One should be cautious for spurious correlations. These can result from a search over correlations between many variables (without type I error control), or from overseeing a time dependence of the variables (violation of i.i.d. assumption).

Moreover, the principle relies on the important assumption of **unbiased data selection**. Selection bias can be present when data selection is based on the value of some unobserved variable that is the effect of some observed variables. Take for example the graph in Figure 2i. Suppose that X and Y happen to correlate in the specific case that $S = 1$, an example of a spurious correlation. If we only have data with this value of S, we could erroneously infer a causal relation between X and Y.

This assumption is required for many causal inference methods. Take for example Local Causal Discovery (LCD) [Cooper, 1997], which depends on some local pattern of CIRs to infer an ancestral relation. If CIRs can be the result of selection bias, the pattern is not sufficient anymore to infer the ancestral relation.

Faithfulness

Faithfulness is a very common assumption. It reverses the implication of the Markov Property, such that conditional independence implies d-separation in the graph:

$$\boldsymbol{A} \underset{\mathbb{P}_{\boldsymbol{X}}}{\perp\!\!\!\perp} \boldsymbol{B} \,|\, \boldsymbol{C} \implies \boldsymbol{A} \overset{d}{\underset{\mathcal{G}}{\perp}} \boldsymbol{B} \,|\, \boldsymbol{C}$$

Many causal inference methods use this assumption to restrict the set of possible graphs by measuring conditional dependences and independences. The test used to measure dependences and independences relies on additional assumptions, which we jointly call the assumption of a **perfect independence test**. For example, the partial correlation test relies on normality of the data. When the data and assumptions are sufficient to infer the MEC of the graph, we say that the MEC is **identifiable**.

Faithfulness may be violated. Take the graph in Figure 2iii. Some causal mechanism may result in an independence between X and Z, even though Z is functionally dependent on X and an intervention on Y would show this.

It may be problematic that faithfulness requires that all CIRs in the data are in the graph. Especially when the graph is large and contains cycles, there may be hypothesis testing errors [Uhler et al., 2013].

An implication of faithfulness is **causal minimality**. A distribution satisfies causal minimality with respect to some graph if it is Markov to the graph, but not to any proper subgraph.

One method that relies on faithfulness is Accounting for Strong Dependences (ASD) [Hyttinen et al., 2014]. All dependence and independence relations are encoded as soft constraints in an Answer Set Programming (ASP) solver, along with constraints that determine how CIRs relate to the graph structure.

Causal sufficiency

The presence of latent (unobserved) variables can make it harder to identify parts of SCMs. Latent confounders that affect multiple variables influence the CIRs. Some methods rely on the **causal sufficiency** assumption that there are no latent confounders. Figure 2iv shows a violation of this assumption, where Z is latent.

Inductive Causation (IC) [Verma and Pearl, 1991] checks for every pair of variables X and Y that are dependent if there is a set of variables $\boldsymbol{S}$ that renders them conditionally independent: $X \perp\!\!\!\perp Y \mid \boldsymbol{S}$. If no such set exists, there must be a direct causal relation, because by Reichenbach's principle and causal sufficiency there cannot be another way in which they are dependent.

Acyclicity

The **acyclicity** assumption restricts the class of graphs to directed mixed graphs[3] (DMGs). Inference under this assumption is typically easier, because the class of possible graphs is much smaller. Nevertheless, some methods can be generalized when the concept of σ-separation is introduced to replace d-separation (e.g. Mooij et al. [2016]).

One advantage of assuming acyclicity is that it is possible to define some ordering in the variables. In this order, variables precede their descendents in the graph. Greedy Equivalence Search (GES) [Chickering, 2002] is an example of a method that exploits this property. The search for the MEC is transformed to a search for an order that corresponds to the MEC. A greedy search algorithm is used to move between permutations of the order to efficiently find the correct MEC.

Exogeneity

Some methods distinguish between two types of endogenous variables: the system variables $\boldsymbol{X}$ and the context variables $\boldsymbol{C}$. The context is seen as external to the system. This means that according to the **exogeneity**[4] assumption, no

[3] DMGs are like DAGs, with bidirected edges that indicate latent confounding.

[4] Note that 'exogenous' might refer to observed, endogenous context variables, or unobserved exogenous variables as represented by $\mathcal{H}$

edges in the graph can point from a system variable to a context variable.

This assumption is violated in the graph in Figure 2v. We observe an effect on X when we intervene on Y. If we assume that C is exogenous, we could erroneously conclude that there is a direct relation $Y \to X$. In this case however, C mediates this relation.

Typically, the context variables are used by the modeller to encode some causal background knowledge, such as the type of intervention. Invariant Causal Prediction (ICP) [Peters et al., 2016] leverages the exogeneity assumption to compare the conditional distribution of a variable given a set of potential parents, across values of the context. If the context itself is not a direct parent of the variable, this conditional distribution should be invariant, which allows ICP to infer causal relations.

2.3 Causal Discovery Methods

In this section we give an impression of the diversity of causal discovery, by describing some well-known methods in more detail. We give a concise description of the algorithm and the assumptions that it relies on.

Constraint-Based Causal Discovery

Constraint-based approaches derive constraints from the data, and use it to restrict the class of possible underlying causal graphs. Causal inference is treated as a constraint-satisfaction problem. Generally, CIRs are chosen as constraints, which can be used to derive the separation of variables and the orientation of edges in the graph. We first describe three global methods IC, PC and FCI, and then three local methods Y-Structures, LCD and ICP.

Inductive Causation (IC) was introduced by Verma and Pearl [1991] and generally describes how we can induce a Partially Directed Acyclic Graph[5] (PDAG) from conditional independences in the data. The algorithm follows two steps: inducing the skeleton and orienting the edges. For every pair of variables X and Y, we check whether there is an edge in the PDAG. Using the faithfulness assumption, we add an edge if there is no separating set of variables $\boldsymbol{S}_{XY}$ that makes X and Y conditionally independent ($\nexists \boldsymbol{S}_{XY} : X \perp\!\!\!\perp Y \mid \boldsymbol{S}_{XY}$). One easy step of edge orientation uses the separation criterion of colliders. If two non-adjacent variables X and Y share a neighbour Z that is not in $\boldsymbol{S}_{XY}$, it must be a collider on the path and we induce edge orientation $X \to Z \leftarrow Y$. Application of additional orientation rules lead us to a maximally oriented PDAG, which describes the MEC of graphs that induce the joint data distribution. One early set of such rules was described by Spirtes et al. [2000] in the SGS algorithm, which was named after the authors.

IC is quite limited, because it relies on several assumptions about the underlying SCM (e.g. causal sufficiency), and its naive implementation is costly

[5] A PDAG is a graph without directed cycles, edges may be directed or undirected

due to the search over all separating sets $\boldsymbol{S}_{XY}$.

PC, named after its inventers Peter Spirtes and Clark Glymour [Spirtes and Glymour, 1991], reduces the cost of naive IC. A systematic algorithm finds the separating sets $\boldsymbol{S}_{XY}$ in polynomial time. Starting from a fully-connected graph, edges are systematically removed by considering separating sets of increasing cardinality, and only taking into account the variables that neighbour X and Y. For example, first edges are removed between variable pairs that are independent given the empty set (cardinality 0). Then, edges are removed between remaining adjacent variable pairs that are independent given one of their neighbours. Already some possible separating sets can be skipped here, because edges were removed in the previous step. Therefore, as we consider larger possible separating sets, the number of neighbours to choose from decreases.

Fast Causal Inference (FCI) extends PC to allow for selection bias and latent confounding. Dropping the causal sufficiency assumption means that it should consider some possible separating sets $\boldsymbol{S}_{XY}$ that contain variables not adjacent to X or Y (specifically, their ancestors). FCI is a feasible algorithm for datasets with many variables when the underlying graph is sparse and bidirected edges (i.e. confounded variables) are not too much chained together. It was first introduced by Spirtes et al. [1999], and gradually developed since then. A modern version named FCI+ by Claassen et al. [2013] is relatively fast. It finds the Partially Ancestral Graph (PAG)[6] in polynomial time.

Y-structures are a pattern in a PAG, which has information about ancestral relations between four random variables. Mooij et al. [2015] showed that a set of independence tests can be used to infer some ancestral relations from observational data. This local method can be used to find an incomplete set of ancestral relations in the underlying SCM.

Specifically, we marginalize over four random variables X, Y, Z and U. Then we perform four statistical tests: $Z \perp\!\!\!\perp Y \,|\, X$, $Z \not\perp\!\!\!\perp Y$, $Z \not\perp\!\!\!\perp U \,|\, X$ and $Z \perp\!\!\!\perp U$. If all tests are positive, there are only two possible PAGs representing the marginalization over the four variables, which are called Y-structure (a term coined for a score-based method by Mani [2006]) and Extended Y-structure. In both PAGs, we can use the backdoor criterion [Pearl, 2009] to infer that $p(Y \,|\, \mathrm{do}(X = x)) = p(Y \,|\, X)$.

Mooij et al. [2015] investigate the performance effect of adding more independence tests. By adding two more tests, the Y-structure remains the only possible PAG. After that, more redundant independence tests can be added. Adding more tests necessarily reduces recall, but precision can improve if the extra tests eliminate false positives. Interestingly, in their experiments on synthetic data, maximum precision is not always achieved with the minimum or maximum number of tests.

An interesting insight from Mooij et al. [2015] is that the faithfulness assumption becomes more problematic as the number of random variables grows.

[6] The PAG is a graph that can represent latent confounding.

The data is sampled from a SCM, which makes the faithfulness assumption very reasonable. However, it appears that the marginalized data can be almost faithless to the supposed PAG.

Local Causal Discovery (LCD) is a local method that looks at ancestral causal relations in the context of a changing environment. In the data, we marginalize over three variables (C, X, Y), one of which is exogenous (C). Next, we perform three statistical tests: $C \perp\!\!\!\perp Y \mid X$, $C \not\perp\!\!\!\perp X$ and $X \not\perp\!\!\!\perp Y$. Cooper [1997] proves that there is a relationship $X \to Y$ if these tests are positive.

This method allows for latent confounders, and assumes that there is no selection bias. Note that this method can find only a subset of ancestral relations. The method is proven by listing all possible causal graphs of three variables in which the context variable is not caused by endogenous variables. Out of the 32 networks in which X causes Y, LCD is only able to identify this relation in 3 cases. The others are not identified, because there are other configurations that induce the same independence relations. For example, if C causes Y not only through X, but via another path as well, we will not find that X causes Y.

Invariant Causal Prediction (ICP) is a more general local method that also uses the context of a changing environment. In its original formulation by Peters et al. [2016] it outputs a subset of parents of a target variable, assuming causal sufficiency and acyclicity. Faithfulness is not required. Mooij et al. [2016] show that when faithfulness is assumed, we no longer require the causal sufficiency and acyclicity assumptions, and the algorithm outputs ancestors.

When we model the context (e.g. experimental setting) as a variable that is not a direct cause of the target variable, then the conditional distribution of the target variable given its direct causes is invariant to the value of the context variable. This property is used by ICP to find parent or ancestor relations in data from different contexts.

In a naive implementation, we would have to investigate every possible parent set for every target variable, making the complexity exponential in the number of variables. Since (direct) causes tend to correlate with their effect, it is reasonable to only consider variables that have a relatively high correlation with the target variable. Such an approximation was used by Peters et al. [2016] and Meinshausen et al. [2016] to apply ICP to the dataset of Kemmeren et al. [2014] with over six thousand variables.

Score-Based Causal Discovery

Score-based approaches provide an alternative to the methods discussed above. They search for the causal graph that optimizes some loss function, often based on independences in the data. Two global methods are discussed below.

Accounting for Strong Dependences (ASD) is a method that focusses on graphs satisfying dependences in the data, whereas most methods focus on the independences. The groundwork of the method was done by Hyttinen

et al. [2014]. Dependence and independence relations in the data are encoded as soft constraints in ASP (Answer Set Programming), together with rules that determine that the solution should be a valid graph. The solution is then found by an ASP solver, that minimizes the loss computed as the sum of the weights of the constraints that are not satisfied. This method can only be applied to problems with a small number of variables, because the number of dependence and independence relations quickly becomes very large.

Magliacane et al. [2016] compute weights based on the p-value of the conditional independence test and the significance level, such that strong dependences obtain a larger weight than independences. They also provide a method to compute a confidence score for single features of the graph (like an ancestral relation) by solving two optimization problems and subtracting the losses. In one problem, the presence of the feature is added as a hard constraint, in the other the absence is added as a hard constraint.

As an implementation of their Joint Causal Inference (JCI) framework, Mooij et al. [2016] adapt the ASD method to include interventional data. In their adaptation, some constraints are added to the ASP problem that encode the JCI assumptions for interventional data.

Greedy Equivalence Search (GES) was first introduced by Meek [1997] and further detailed by Chickering [2002]. A search for the Markov equivalence class (MEC) is performed in two phases. The graph is initialized without edges. In the first phase, we move between MECs by adding edges. In each step, we consider the set of MECs that are one edge addition away from the current MEC. Reversing a covered edge[7] does not change the MEC, so we need to consider every combination of covered edge reversals, followed by a single edge addition, followed again by covered edge reversals. We score the MECs and move to the one with the highest score. Meek [1997] proposed the Bayesian scoring criterion to score DAGs in a MEC. Under some conditions this can be approximated by the Bayesian Information Criterion (BIC).

Once we reach a local maximum, Chickering [2002] shows that the MEC contains the generative distribution. The second phase is analogous to the first, but this time single edges are removed. Chickering [2002] shows that this final MEC is a perfect map of the generative distribution.

Hauser and Bühlmann [2012] generalize the method to datasets with interventional data (GIES). They introduce the interventional MEC as the object of the search. The algorithm is adapted by introducing the turning phase. Repeated application of the three phases leads to the interventional MEC underlying the generative distribution. They note that the space of interventional MECs is more fine-grained, meaning that this type of data leads to improved identifiability.

[7] An edge $X \to Y$ is covered if X and Y have the same parents (and Y has X as parent as well)

Hybrid methods

Finally, there are also methods that rely on both constraints and the optimisation of some score. SP is the most relevant example, and is described below.

Sparsest Permutation (SP) is a global method proposed by Raskutti and Uhler [2018] that searches for the MEC in a space of orders of random variables, using observational data. The order is a permutation of variables in which ancestors precede descendents. Given an order and a method to infer dependence and independence relations, we can infer a DAG that satisfies causal minimality. The SP algorithm searches over the DAGs infered from all possible permutations and selects the DAG with the smallest number of edges. Raskutti and Uhler [2018] show that this algorithm is consistent, which means that it finds a DAG in the correct MEC as sample size tends to infinity. This consistency relies on an assumption that is a weaker version of faithfulness.

Solus et al. [2017] propose a greedy algorithm (GSP) that increases the number of variables that can be handled from about 10 to hundreds. The only sacrifice is that a somewhat stronger assumption is required, which is still weaker than faithfulness. They introduce the DAG associahedron, a sub-polytope of the permutohedron which indicates possible directions for the search. These directions are based on covered edges.

Wang et al. [2017] adapt the algorithm further to take interventional data into account (IGSP). The score is now the sum over the individual scores for each intervention distribution. This score is a weighted sum of the number of edges in the inferred graph, excluding edges towards the intervened variable, and a maximum likelihood score of the interventional data given some model assumptions.

3 Data

3.1 Data Source

We evaluate our methods on a biological dataset. The DNA of a cell contains genes that are involved in many of the cell's functions. They are typically responsible for the production of a protein. The first step in this process is to copy its information to a Messenger RNA (mRNA) strand. The amount of mRNA that is measured in an experiment indicates how active a specific gene is, or how high its *expression level* is.

Genes interact to fulfill a plethora of cell functions. For example, the expression of one gene might up- or down-regulate the expression of some other gene. This interaction is regulated by biochemical processes.

For a variety of reasons, it is interesting to know how genes interact precisely, that is: what the regulatory network looks like. By jointly measuring the expression of a large set of mRNA strands we obtain an mRNA profile. Collecting a set of these profiles allows us to model the joint distribution of mRNA expression and the causal relations among them.

Specifically, we use mRNA profiles from Kemmeren et al. [2014]. They measured a profile of 6.182 genes in cells of the yeast species *Saccharomyces cerevisiae* (baker's yeast), using DNA microarray technology (Figure 3). The dataset consists of 262 observation samples obtained from unaltered *wild type* cells, and 1.479 intervention samples obtained from *mutant* cells where one gene was deactivated.[8]

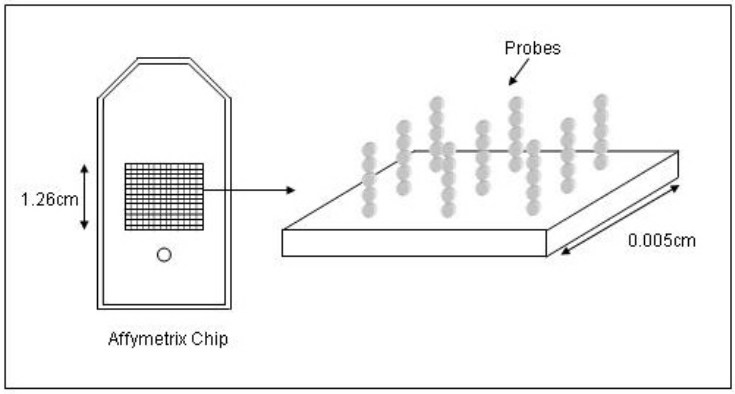

Figure 3: Illustration of a typical microarray chip. Every small square can be used for an experiment to measure gene expression levels.[9]

Both the observational and interventional profiles are reported relative to some average wild type profile. Kemmeren et al. [2014] report that the intervention profiles are compared to a set of 428 wild type profiles. Gene expression is measured as fluorescent intensity in the experiments. In this thesis, we work

[8] A newer version of the dataset exists with 1.484 intervention samples.
[9] Source: http://grf.lshtm.ac.uk/microarrayoverview.htm

with the difference in log_2 fluorescent intensity between the data point and the reference wild type data, which indicates deviation from the normal gene expression levels. Because the data already indicates values relative to a norm, we choose to not preprocess the data further.

There are some details of the experiments that might be of relevance in a discussion of underlying assumptions. First of all, the researchers chose to measure only a subset of about 25% of all genes. Selection criteria included whether genes were expected to be involved in regulating other genes, and only genes were selected that do not play a vital role in keeping the cell alive (viability).

Furthermore, the profile resulting from an experiment had to pass a quality control before being admitted to the dataset. Failing this test resulted either in repeating the experiment, or excluding the mutant. Although these checks improve the quality of the data by removing some failed experiments, they may result in some selection bias as well.

Another form of selection bias is inherent in the experimental method. The mutant cells need to reproduce many times until a culture is grown that is large enough to do the measurements. However, cells with certain properties may reproduce quicker or easier, and be overrepresented in the measurement. It is unclear how large this effect may be.

A final factor to consider is that data from previous work of the same institute is included in the dataset, specifically from Lenstra et al. [2011] and Van Wageningen et al. [2010]. The authors note that they could not find any significant differences in the data. Nevertheless, this information can be seen as a context variable, and ignoring it is an explicit modelling assumption.

3.2 Binary Ground-Truth

We suppose that a SCM generates the dataset. Every gene expression is interpreted as an endogenous random variable. We suppose that there is an underlying causal mechanism (function $\boldsymbol{f}$) that models the relations among these gene expressions. The interventional data is generated by a SCM induced by a perfect intervention on the expression of the mutant gene, making its expression level a lot smaller. These interventions, along with the observational data, provide us with information about the SCM.

The values in the interventional data represent deviations from the normal (wild-type) gene expressions that are measured when there is a perfect intervention on one gene (in the mutant). The more these values deviate from zero, the more likely it is that they are in fact deviating as a result of this intervention. We construct a set of binary ground-truth relations by selecting per intervened gene, those genes that respond with an absolute value exceeding some threshold. We interpret the result as a set of causes and (possibly indirect) effects.

Four thresholds are used in this thesis. Kemmeren et al. [2014] used a

threshold of 1.7, stating that lower levels may be biologically relevant, but focussing on robust changes makes it more likely that they are biologically meaningful. We express thresholds in terms of the percentage of relations that are true under them. To evaluate our methods, we use the Kemmeren et al. [2014] threshold which corresponds to 0.1%, along with the weaker thresholds 1% and 0.1% that allow us to show more information about the system, possibly in a more noisy way. These thresholds are shown in the graphs of this chapter.

The binary ground-truth is also used in the order-based LCD method. Order inference uses a 20% threshold, and position inference a 10% threshold. Both were chosen based on performance of these subtasks, which are explained in their corresponding chapters 4 and 5.

Often, we restrict ourselves to the 1.479 intervention genes (i.e. genes that occur in the intervention data as target of a perfect intervention) as possible effects, instead of all 6.182 measured genes. This makes it easier to interpret our metrics, because every relation between these genes is captured by the interventional data. We call this subset of the intervention data the *intervention table* $\boldsymbol{X} \in \mathbb{R}^{1479\times1479}$ where $\boldsymbol{X}_{ij}$ is the relative expression level of gene i in the experiment with gene j knocked out (i.e. intervened upon). When subjected to some ground-truth threshold, we get a matrix of the same dimensions that has value True where gene j is a cause of gene i.

We use the binary ground-truth to analyse the dataset, to inform our algorithm to find an order in the genes, and to evaluate causal methods. Scientific knowledge about some relations could be used to evaluate our methods as well, but it is incomplete and somewhat obscure because it is scattered about many papers with different experimental methods and conditions.

3.3 Properties

Certain properties of the dataset are valuable to interpret our inference challenge, and justify the assumptions of our methods. The most challenging property of the dataset is **sparsity**. The data can be interpreted as a collection of single samples from (slightly) different joint distributions, since there is only one measurement of the genes per intervention. The exception is the observational data, which consists of 262 samples of the same distribution.

Figure 4 shows examples of possible relations between two genes, distinguishing observation and intervention data. The relation between two variables can be visually estimated in these plots. In the middle plot, we see that the genes are correlated. By Reichenbach's principle we infer that there is some causal relation. Gene *YHR087W* has reduced expression when we intervene on gene *YDR074W*, but not the other way around. We may conclude that gene *YDR074W* is an ancestor of gene *YHR087W*. In the right plot we see two genes that are correlated, but do not respond to interventions. Assuming that the data selection was unbiased, this indicates confounding.

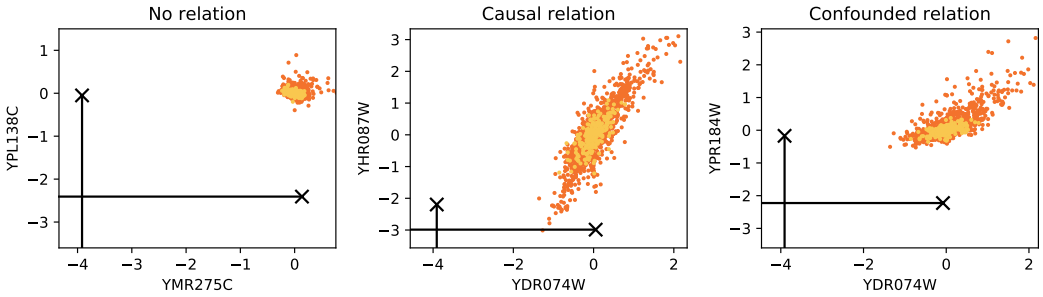

Figure 4: Expression levels of pairs of genes. All observation values are shown in orange, all intervention values in blue. The black crosses show the interventions on the plotted genes, the line shows which gene was intervened on.

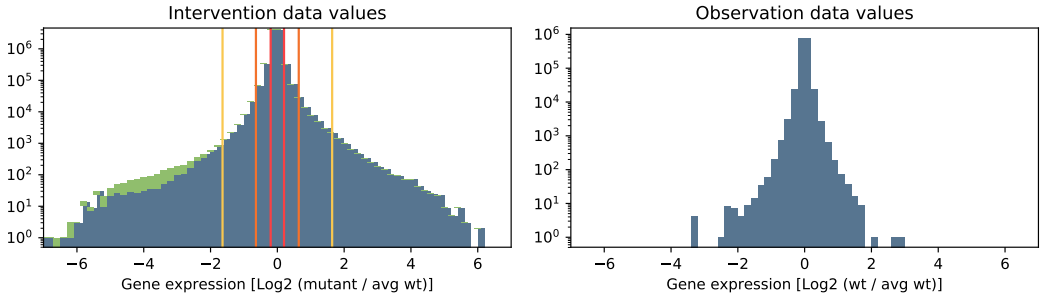

Figure 5: Distribution of expression values in observation and intervention data. The values of the mutant genes themselves are shown in green. The ground-truth thresholds are shown in the intervention plot.

Generally, the intervention data has higher **variance**, because the effect of the intervention propagates to many genes. Figure 5 shows that the interventional data contains more extreme values. Figure 6 compares the variance of genes in the observational and interventional data. In the interventional setting, the variance of genes tends to be about three, in some cases even more than ten times larger.

The partial correlation test is commonly used in the causality literature to test if two variables are dependent or independent[10]. In this thesis we use it to analyse dependences in the data. The test relies on the assumption that the data is **normally distributed**. Figure 7 shows the results of a Shapiro-Wilk normality test. The normality assumption is more valid for the observational data than for the interventional data. This is to be expected, since we model the observational data to be sampled from a single distribution with some noise, and the interventional data from different distributions.

Because the expression of most genes is not normally distributed, the normality assumption is dubitable. However, the independence test that it is used for is very common, and generally considered to be robust to violation of this

[10]When the test fails to reject the hypothesis of dependence, it is commonly concluded that the variables are dependent.

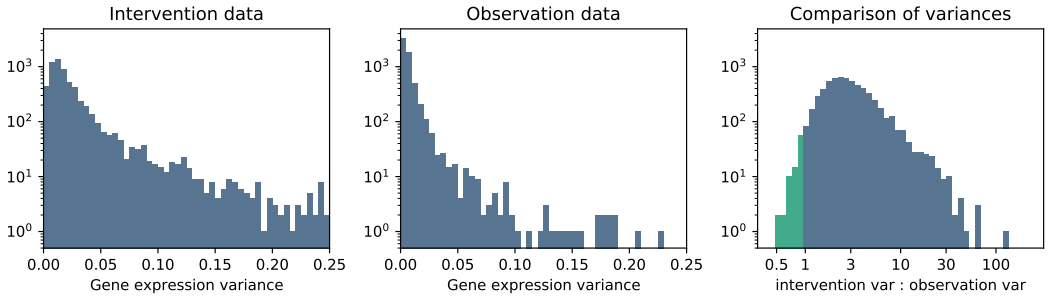

Figure 6: Distribution of variance in the expression values of single genes in the observation and intervention data. The distribution of the ratio between intervention variance and observation variance per gene is shown on the right.

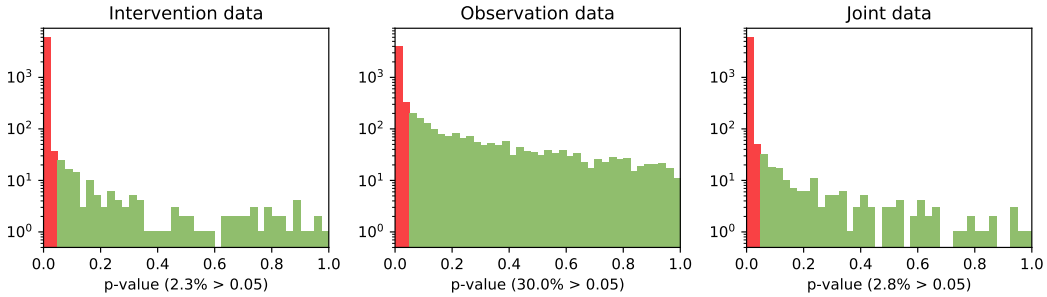

Figure 7: Distribution of p-values from the Shapiro-Wilk normality test per gene. For the intervention and joint data, we sampled 262 values per gene to make the p-values comparable.

very strict assumption. We note that visual inspection of the data as in Figure 4 and the distribution of values in Figure 5 show that the expression distributions are at least similar to the Gaussian shape. Different independence tests can still be interesting because they can capture other forms of independence. In our LCD method, we use a mean-variance test.

The hypothesis of this thesis that there is some implicit order in the genes, requires that there are no **cycles** in the SCM. We construct a graph from the binary ground-truth and compute the number of strongly-connected components (SCCs): the number of variable subsets in which there is a directed path from each variable to each other variable. Combined with the number of variables in SCCs, this gives an indication of the validity of the acyclicity assumption.

The results in Figure 8 show that for any ground-truth threshold, the number of SCCs is small. At the 0.1% threshold (~ 1.7), there are no SCCs and therefore no cycles. At the 1% threshold, there are three SCCs and more than half of variables are in them. All variables are in one big SCC when we use the 10% threshold, which makes sense since we include so many relations.

We conclude that the presence of cycles challenges our hypothesis. In finding some underlying order, we will need to ignore some relations to get an acyclic approximation. In fact, the order inference algorithm that is finally

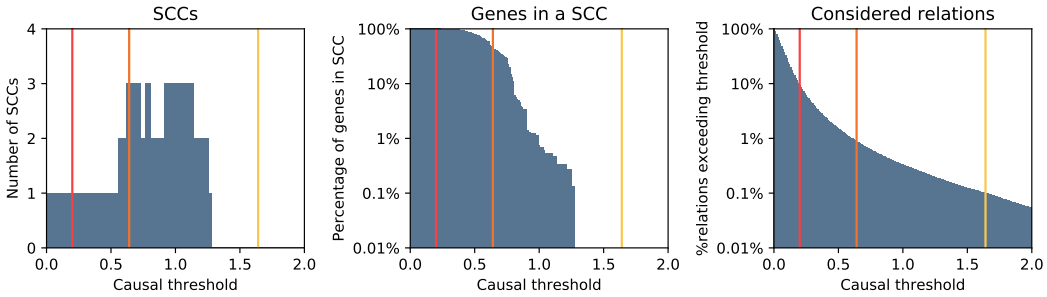

Figure 8: Left: number of SCCs in the graph per ground-truth threshold. Middle: percentage of intervention genes that are in a SCC per threshold. Right: percentage of causal relations that are significant according to the threshold.

used is even based on a 20% threshold (~ 0.128), and designed to break cycles effectively.

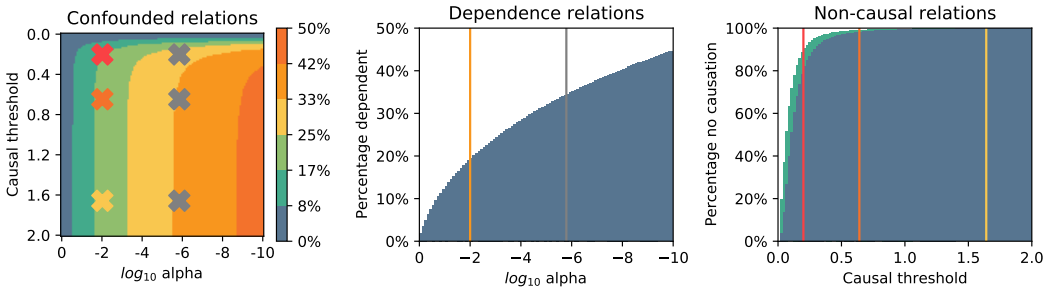

Figure 9: Left: percentage of relations that are only related by confounding, for different binary thresholds, and significance levels. The two crosses show the alpha and thresholds used in this thesis. Middle: percentage of relations that are dependent per significance level. Right: percentage of relations that are not causal per binary ground-truth threshold (inverse of Figure 8-right)

. Relations that do not exist in either direction are indicated by dark blue, relations that do exist in one direction are indicated in light blue.

Next we test the frequency of **confounding** in the data. We could search for common ancestors in the binary ground-truth, but this will not find latent confounding. Therefore, we test for each variable pair if they are dependent by the partial independence test, and not a cause of each other in the ground-truth. Note that we will miss variable pairs in which one variable causes another, while also being confounded. Figure 9 shows that at our ground-truth thresholds, and a significance level of $\alpha = 0.01$, about 20% of all relations among intervention genes is confounded. Note that all ground-truth thresholds are quite strict, so most relations are non-causal. Since about 20% of relations are dependent at the chosen α, this also determines mostly the percentage of confounded variable pairs.

Being a biological system, we suspect that gene expressions are highly interconnected, explaining a high percentage of confounding. However, most of the relations are indirect and have relatively small effect, which makes it

hard to infer the causal structure. Large prevalence of latent confounding is acceptable, since most methods can deal with it. However, the recall of LCD may be quite limited. It cannot identify ancestral relations that are also confounded, meaning that an expected 20% of all relations are already disqualified.

3.4 State-of-the-Art

Due to the sparsity of the data, any kind of causal inference task is challenging. The few methods that have been applied on this data improve over some baseline only in a small number of strongest predictions. ICP is the method with the best performance. LCD has been tested as much faster alternative [Versteeg and Mooij, 2019]. Preselection is commonly used to speed up the algorithms, and might even improve accuracy of the most certain predictions. Three papers are shortly discussed in this subsection.

The ICP algorithm was first proposed by **Peters et al. [2016]**, and tested on multiple datasets including the one of Kemmeren et al. [2014]. Two versions of ICP are designed, one with a test on regression coefficients, the other as faster alternative with an approximate test of residuals. L2-boosting regression [Schapire et al., 1998] is used to preselect variables. Effects are considered true if the absolute intervention value exceeds the upper or lower 1% of the observation values per gene, resulting in about 9.2% of all effects being true. 6 out of the top 8 predictions of both ICP versions are true. A baseline that uses correlation on the pooled data has 2 true predictions in the top 8.

Meinshausen et al. [2016] use ICP in a slightly different way, and evaluate against a different definition of true effects. They introduce a generalized ICP that accounts for latent confounding. They use stability selection [Meinshausen and Bühlmann, 2010] to make a larger number of more fine-grained predictions, visualised in a receiver operating characteristic (ROC) curve. Effects are considered true only if the values of the cause and effect gene are extremes in the joint data, leading to a strict ground-truth set of about 0.1% of all effects. The normal ICP version predicts 7 true effects in the first 9 predictions, and makes not one other true prediction in the top 25. The generalized ICP version predicts 5 true effects in the first 8 predictions and keeps making true predictions at a declining rate after that. A baseline using cross-validated sparse Lasso regression is provided for comparison, but it performs worse than random.

The most recent work on causal inference applied to the Kemmeren et al. [2014] dataset is by **Versteeg and Mooij [2019]**, who combine elements of both papers and investigate LCD as an alternative method that is simpler and faster. Predictions are made using stability selection. L2-boosting regression preselection is used for ICP and as an option for LCD. Two independence tests are compared, but no significant difference in performance is found. True effects are defined by the top x% absolute value in the intervention data. Meth-

ods are compared on a ROC curve, at ground-truth thresholds 10%, 1%, and 0.1%, but the comparative results are the same. In the first 100 predictions, ICP performs best, followed by LCD using preselection. An interesting result is that a baseline using only the preselection method outperforms the methods thereafter. LCD without preselection performs worse than this baseline from the start. ICP predicts about 40, 20, and 10 effects correctly in the first 100 for the respective ground-truth thresholds.

4 Approximating Variable Order

4.1 Methods

The genes in the dataset can be ordered according to their position in the ancestral hierarchy. This order is clearly defined when there are no cycles in the underlying graph. When cycles do exist, we can only infer some order that satisfies many ancestral relations. We hypothesize that an approximation to this underlying topological order can be leveraged to improve causal discovery.

If we ignore the purely observed genes that do not occur as knock-out gene, we are left with a square *intervention table* $\boldsymbol{X} \in \mathbb{R}^{N \times N}$. X_{ij} is the relative expression level of gene i in the experiment with gene j knocked out. If we take into account all intervention data, we have $N = 1479$ genes. This table represents a complete directed graph if it is interpreted as a weighted adjacency matrix. A straightforward way to model the problem of finding order, is to minimize the sum of the absolute weight of the edges that violate the order π, which is represented by a permutation of genes. We call the average absolute weight of violating edges the *penalty* p_π:

$$p(\pi) = \frac{\sum_{\pi_i < \pi_j} |X_{ij}|}{\frac{1}{2}N(N-1)}$$

N is the number of intervention genes. To make this number more interpretable for different intervention tables, we define the *penalty ratio* r_π as the penalty divided by the average absolute value of all relations, excluding values of the knocked-out genes themselves. We present this value as a percentage. A random order is expected to have a penalty ratio of 100%.

$$r(\pi) = \frac{\sum_{\pi_i < \pi_j} |X_{ij}|}{\frac{1}{2}N(N-1)} \frac{N(N-1)}{\sum_{i \neq j} |X_{ij}|} = \frac{2 \sum_{\pi_i < \pi_j} |X_{ij}|}{\sum_{i \neq j} |X_{ij}|}$$

Assuming that the expression values reflect the importance of ancestral relations, we can weigh the edges by the respective expression values. In an unweighted variant, we could add only those edges whose expression value exceeds some threshold. This yields a sparser graph that may still contain cycles. It is important to realise that this thresholding has different meaning and implications than the thresholding on which we may base our evaluation of predictions. In testing predictions, we are interested in meaningful relations. In finding order, we are more pragmatic and could use any threshold that yields the best prediction algorithm.

Minimum Feedback Arc Set Problem

The presence of cycles in the underlying graph is an important challenge to inferring order. Finding the order in an acyclic graph is trivial and can be computed with a complexity linear in the number of nodes and edges (for

example Kahn's algorithm [Kahn, 1962]). One way to find a good ordering in graphs with cycles, is to eliminate edges to make the graph acyclic, without too much damage to the hierarchy. An order is then found in the acyclic subgraph.

This strategy can be modeled as the Minimum Feedback Arc Set (MFAS) problem. The minimum feedback arc set is the smallest set of edges whose elimination makes the graph acyclic. A weighted version of the problem aims to find the set of edges with minimal summed weight whose elimination makes the graph acyclic.

In theory, we could find the optimal solution for the complete directed graph constructed from the intervention table, eliminate the minimal feedback arc set, and use Kahn's algorithm to infer an order in the remaining acyclic subgraph. Unfortunately, the MFAS problem is expensive to solve. In fact, Guruswami et al. [2008] show that even approximating the maximum acyclic subgraph problem with an approximation ratio lower than 0.5 is Unique-Games hard. That is, it is impossible to get a polynomial-time approximation of the minimal feedback arc set that is better than imposing a random order.

Because of the complexity of the problem, we focus on heuristic methods to eliminate edges, or optimize the order directly with respect to the penalty. We first discuss approaches that use the continuous values of the intervention table, followed by approaches that first binarize the data with some threshold. Finally, we discuss our experiment and results, justifying the algorithms that we use in the causal inference method described in the next chapter. Appendix B contains some further details on the implementation of these approaches, along with some notes about preliminary work to determine their parameters.

Continuous approaches

A straitforward approach is to directly optimize the penalty. Because it is expensive to find the optimal solution, we designed an **evolution strategy** (ES) to search for a good solution[11]. This method iteratively updates a *population* of solutions according to evolutionary principles. A solution in this case is a permutation of variables, which are initialized randomly. Every iteration, we *recombine* pairs of solutions to form new solutions[12]. The penalty of each new solution is computed and the best solutions are kept in the population. We experimented with different recombination methods and different parameters on synthetic data and selected a good setting for the experiments in this chapter.

The population consists of 100 solutions. We use cycle crossover [Oliver et al., 1987] as recombination method, which preserves the absolute position of indices in the permutations. If we have two permutations π^1 and π^2, we first identify all *cycles* $C_k \subseteq \{1...N\}$ such that $\{\pi^1_i\} = \{\pi^2_i\}, i \in C_k$. The indices in a cycle can be swapped between the two permutations. The new solutions

[11] cf. Eiben et al. [2003] for an introduction to evolutionary algorithms.

[12] Usually, recombination is followed by mutation - small random changes to the solution. We found no mutation method with a strong effect on the penalty and decided to leave it out.

are formed by swapping alternating cycles: $\pi_i^1 \leftarrow \pi_i^2, \pi_i^2 \leftarrow \pi_i^1$ for all $i \in C_k, k$ even.

Another approach is inspired by implicit assumption that the underlying graph is acyclic. We interpret the intervention data as a noisy weighted adjacency matrix of this graph, and wish to find an order in accordance with the largest values in the matrix. To find this order, we first use the **Edmonds algorithm** [Edmonds, 1967] to find the spanning arborescence of maximal weight, i.e. a rooted directed tree spanning all nodes such that the values on the selected edges are maximal. For these values we take the absolute intervention values. When we have this arborescence, we can select an order of nodes that satisfies it, for example using Kahn's topological sort algorithm [Kahn, 1962].

The Edmonds algorithm has complexity $O(EV)$. When we use the complete intervention table, this scales with the number of genes to the power 3. We propose a faster **Sparse Edmonds algorithm** that scales with power 2, by allowing only a maximum number of edges to be added per node. In our experiment we use the top 10 edges with highest absolute value.

Discrete approaches

We also investigate some approaches using binarized intervention data. Subjecting the data to a threshold makes it possible to use some methods, at the cost of information loss. First, we applied a very similar **evolution strategy** to the binary data. Solutions are now selected based on a *binary penalty* computed from the binarized data. Because the ES is designed to directly optimize some objective, we see that obfuscating this objective is detrimental to its performance.

A different approach to approximating order is to rank the nodes in a graph. We interpret the binarized intervention data as an adjacency matrix of an unweighted directed graph. Three methods were applied to assign a score to the nodes of the graph, and we infer an order by sorting the nodes by their score and randomly deciding ties. These methods were all designed with a specific application in mind, which means there is no a priori guarantee they will work well on our problem.

Furthermore, the methods are not symmetrical. We can construct a graph with edges from cause to effect and determine the order by sorting one way, or construct the graph with edges from effect to cause and sort reversely. Both approaches will yield a different result.

PageRank [Page et al., 1999] was developed to score the relative importance of web pages, based on hyperlinks. The score represents the probability that a fully randomized user clicking hyperlinks, ends up on some website. The scores can thus be interpreted as a probability mass function over nodes. The score of one node in the graph depends on the score of the parent nodes linking to it, and to how many other nodes the parents link. This last factor is undesirable in our context. If some node has edges to multiple other nodes

that have no other edges, they will likely have a lower score than their parent (see for example Figure 12).

PageRank can be adapted to allow weighted edges. For example, Tyagi and Sharma [2012] use the frequency of clicks per link to determine the probability of moving to another page, instead of a uniform probability over all links.

Where the PageRank score is intended as probability of reaching a website X by random clicking, we loosely interpret it as either the likelihood that a random intervention affects gene X, or reversely that a random effect was caused by an intervention on gene X. A high PageRank score should correspond to a low resp. high position in the causal order[13].

Social Agony was proposed by Gupte et al. [2011] to analyse hierarchy in social networks. Users of a directed social medium are assigned a discrete score that is computed based on who follows who. Formally, the group of users is subdivided as a partially ordered set, because we cannot distinguish between users with the same score. The granularity in our experiments will prove to be so low that it hurts the performance.

The algorithm assigns a rank to every user. Users are said to experience agony when they follow someone of lower rank. This agony is usually computed as the difference of their rank plus one. The algorithm then assigns ranks to users such that the total agony of all users is minimized. A version of Social Agony with weighted edges was proposed by Tatti [2015], in which the agony of each follow-relation is weighted by the corresponding edge weight. We did not test this version here.

In the original formulation, agony is caused by following users of lower rank. We see it as a penalty for a gene that has an effect on a potential cause, i.e. a gene higher in the order. The score then corresponds to an order in which genes minimally affect these potential causes. In the reversed case, agony is a penalty for being affected by a potential effect gene lower in the order, with a score corresponding to an order in which genes are minimally caused by their potential effects.

The final scoring algorithm is **TrueSkill**, proposed by Herbrich et al. [2007]. It is used to assign a skill level to players of an online video game, based on match outcomes. The set of all matches is modeled with a factor graph. The skill of a player s_i is normally distributed with some mean μ_i and variance σ_i^2. The player's actual performance p_i is also normally distributed with his skill as mean, and some constant variance. The match outcome is determined by the difference of the performances d_{ij}. The sum-product algorithm [Kschischang et al., 2001] is used to compute the parameters of each player's skill distribution. The expectation propagation algorithm [Minka, 2001] is used to approximate a distribution of each performance difference d_{ij} as a normal distribution. In ranking the players of a game, it is important that a high rank cannot be achieved with a lucky win against a highly ranked player. Therefore, the TrueSkill score penalizes uncertainty and is defined as $\mu_i - 3\sigma_i$.

[13]Genes that are high in the causal order are expected to be causes of lower genes.

Applied to the gene perturbation data, we interpret a high score as an indication that some gene is a very likely cause of some other genes with high certainty. It should be high in the causal order. In the reverse case, a high score indicates a gene that is very likely an effect of some other genes with high certainty, and should be low in the causal order. No simple adaptation of the algorithm was found with weighted matches, which we could use to include the intervention values.

The last method to be discussed is an algorithm developed by Sun et al. [2017], we will call it **Sun's algorithm**. In the context of crowd-sourced taxonomy graphs, it's aim is to break cycles (which are logically inconsistent in a taxonomy), while preserving the logical structure. This problem is very close to the MFAS problem, which is too expensive to solve exactly. Once the graph is acyclic, it is trivial to infer some variable order.

The algorithm consists of two steps. First, the nodes in the graph are scored. Then, strongly connected components (SCCs) are iteratively broken by removing edges based on the node scores. Sun et al. [2017] compare TrueSkill and Social Agony scoring, and propose three heuristic strategies to break cycles. The *greedy strategy* selects the edge that violates the hierarchy the most, i.e. with the largest difference between node scores. The *forward strategy* selects all outward edges of the node highest in the hierarchy. The *backward strategy* selects all inward edges of the node lowest in the hierarchy. Next to the six configurations that can be made, they provide an ensemble voting method. For each edge we count by how many configurations it is removed. Edges are iteratively removed in order of the number of votes, until there are no more cycles.

Because this method is much more time-consuming then the other ones, we selected two based on a small experiment. The voting method was selected, because it most often removes the smallest number of edges, indicating that it comes closest to approximating a solution to the MFAS problem. The greedy strategy combined with TrueSkill scoring was chosen because it generally performs best out of the individual methods. As mentioned earlier, the superior performance of the TrueSkill scoring is most likely due to the information loss in the discrete scores of Social Agony. We only applied these two methods to a graph with edges from effects to causes. This seems more natural in analogy with the taxonomy application.

4.2 Experimental Results

We compare the algorithms based on the penalty ratio, and computation time. We test each algorithm on samples of the data, varying the sample size N from 2 to 1.479, the full dataset. Every algorithm is tested on the same 5 random samples for each sample size. Some algorithms become very time-consuming to evaluate, and were not tested on larger samples. Note that the variance in penalty and time results from two factors, variance in the sample (especially

for small samples) and variance in the algorithm due to randomness.

Global Results

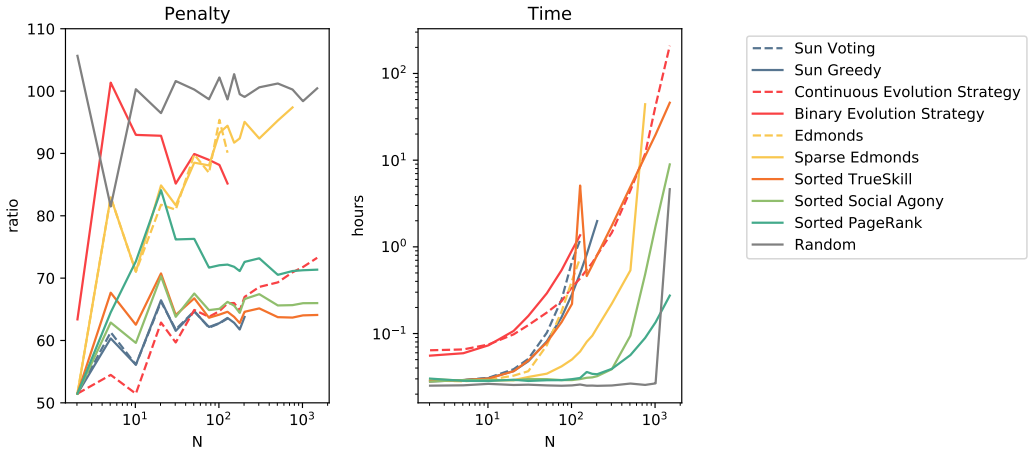

Figure 10: Comparison of penalty ratio and computation time. Averages of results over 5 data samples are shown. Variance generally decreases when sample size increases. It is left out of the graphs to keep them readible.

The comparison of penalty ratio and computation time between all algorithms is shown in Figure 10. At smaller sample sizes, the continuous evolution strategy performs best. This is the only algorithm that attempts to directly optimize the penalty ratio. Because this optimization problem is easy for small samples, this result is to be expected. For larger samples its performance decreases, and the computation tim becomes much longer.

When the samples get larger, the score ordering algorithms and the two versions of Sun's algorithm outperform the others. Of the score ordering algorithms, TrueSkill performs best, followed by Social Agony and PageRank. The order of computation time is reversed, PageRank being the fastest followed by Social Agony and TrueSkill. The two versions of Sun's algorithm seem just a little better than ordered TrueSkill, at the expense of a much longer computation time. Because of this, we select TrueSkill as the most useful algorithm for our causal inference method.

Score Ordering Algorithms

The score ordering algorithms perform well and are relatively fast. Therefore, we look at them in some more detail. These algorithms can be applied in two ways. In the *forward* interpretation, we construct a graph with edges from effect to cause (observed to intervention variable). Variables with a higher score are higher in the causal order. In the TrueSkill model for example, causes are winning from effects and get a higher skill level score. In the *backward* interpretation, edges point from cause to effect. Variables with a higher score

are lower in the causal order. Because the algorithms are not symmetrical, these two interpretations do not yield the same results. The same interpretations can be applied for the Edmonds algorithm, in constructing the (sparse) weighted graph in which the minimal branching is found.

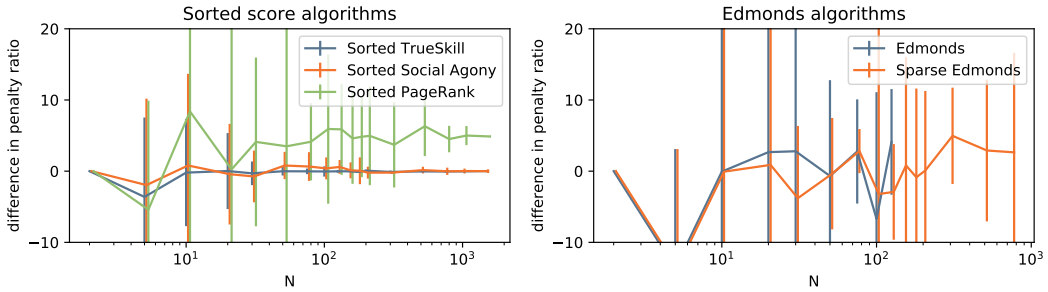

Figure 11: Average difference between Forward and Backward algorithms over 5 samples per sample size. Vertical bars show two standard deviations. A positive difference means that the Backward algorithm has lower penalty.

Figure 11 shows the difference in the penalty ratio between the forward and backward interpretations. For the larger sample sizes, we see that especially PageRank and Sparse Edmonds perform better with the *backward* interpretation (positive difference). TrueSkill, Social Agony and Edmonds show only small differences. In this thesis, whenever the interpretation is not mentioned, we used the *backward* version of these algorithms.

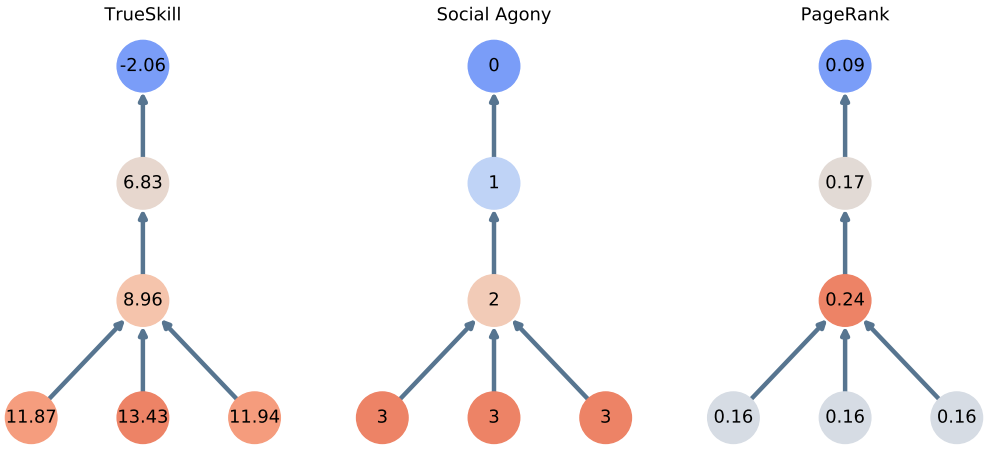

Figure 12: Example scores of backward score sorting algorithms.

Figure 12 shows per algorithm the score it assigns to nodes in a simple graph. The edges point from effect to cause, and are thus reversed for the algorithms. It can be seen that the continuous scores of TrueSkill are somewhat arbitrary. Social Agony is easier to interpret. This specific graph also highlights a weakness of PageRank. The three bottom nodes share the score of their cause, which results in a lower relative score.

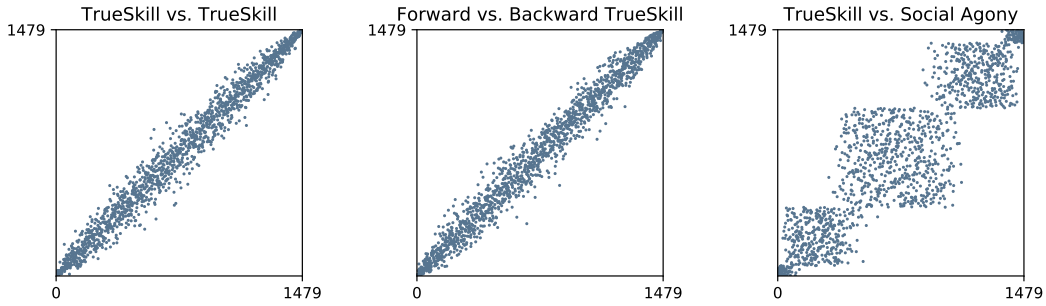

Figure 13: Consistency between order algorithms.

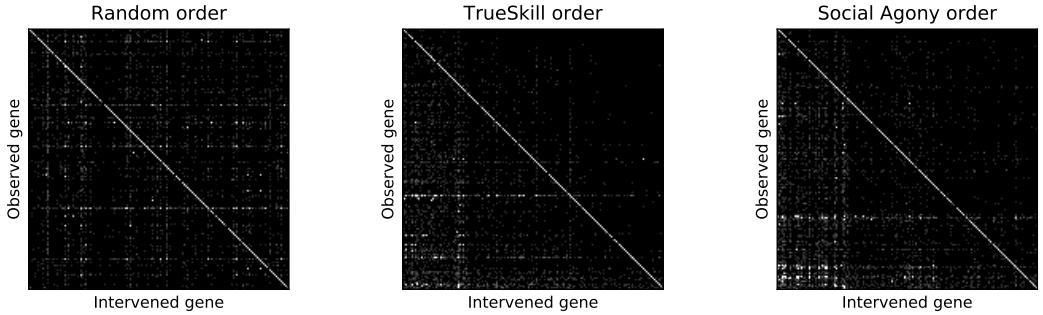

Figure 14: Ordered intervention tables.

A comparison between the infered orders is shown in Figures 13 and 14. Figure 13 compares the one instance of the order found by the TrueSkill algorithm to three other orders. It shows per gene, at what index that gene is placed by the other order. On the left we compare two orders infered using TrueSkill. The variance is only due to randomness in the algorithm. TrueSkill quite consistently puts genes in about the same position in the order. The variance seems to be somewhat lower at the extremes, indicating more certainty about the most obvious causes and effects. In the middle we compare the forward and backward versions of TrueSkill. The figure is very similar, corresponding with the small difference in penalty ratio that we found earlier. On the right we compare TrueSkill with Social Agony. We see that the Social Agony algorithm distinguishes between five discrete scores. The order of genes with the same score is decided randomly. Again, there is more certainty about the extremes of the order. Interestingly, these two algorithms infer a very similar order, Social Agony just has a lower granularity in its scoring. Although this allows TrueSkill to achieve a somewhat better penalty ratio, the Social Agony score may be more informative and useful for causal inference.

A further comparison between these two algorithms is shown in Figure 14. The intervention table is shown with the absolute values grouped into four ranges with corresponding shades of gray. Genes are ordered in the same way on the x-axis and y-axis. High values have the brightest color. On the diagonal we see a bright line, indicating the expression levels of the genes

that are interevened upon. The top-right triangle shows all relations that are violated by the order, and should thus be as dark as possible. Comparing TrueSkill (middle) and Social Agony (right), we see that Social Agony pushes the extreme values as far from the diagonal as possible. These genes fall into the two groups with the highest rank. TrueSkill can make a more detailed distinction between genes. Being able to spread out the most extreme values more, it achieves a better penalty ratio.

5 Causal Discovery using Order-Informed Context

5.1 Local Causal Discovery

Local Causal Discovery (LCD) was originally proposed by Cooper [1997] as a fast, incomplete algorithm. It can determine a direct causal relationship between two variables in a system of three variables. It can also be applied to a larger system by marginalizing over variable triples and testing for ancestral relationships between variables.

LCD is chosen as the causal inference method in this thesis because it is simple, and a good trade-off between performance and computation time. It was applied to the Kemmeren et al. [2014] dataset by Versteeg and Mooij [2019]. They compared LCD to ICP [Peters et al., 2016], presenting a somewhat higher precision in the top predictions with ICP, at the cost of three times the computation time.

Assumptions

The algorithm tests a causal relation $X \to Y$ between two endogenous variables, using a third exogenous variable C (the context). A small set of statistical independence tests restricts the set of possible causal subgraphs with nodes X, Y and C. Under some circumstances, it allowes us to infer that X is an ancestor of Y.

LCD relies on some assumptions. Like many inference algorithms, the starting point is to model the underlying system as a SCM. The context variable needs to be exogenous, such that no system variable can be its cause. Causal faithfulness is assumed to link (in)dependences to the d-separation of variables. To measure these (in)dependences from the data, the dependence and independence tests are taken as oracles, such that their outcomes are taken as the truth. Unbiased data selection and acyclicity are also assumed.

Mooij et al. [2016] show that the acyclicity assumption can be dropped if we adopt a different notion of separation called σ-separation. We will leave this topic at this note, because the theoretical properties of the LCD method are not the focus of this thesis.

Statistical tests

LCD discovers those causal relations, which effect can be measured from the data as $\mathbb{P}_M(Y|do(X = x)) = \mathbb{P}_M(Y|X = x)$. This is the case when there is no confounding between X and Y, no effect of Y on X, and no effect of C on Y that bypasses X. Figure 15 shows the three subgraphs that satisfy the condition. Note that all possible relations between C and X given the exogeneity assumption are allowed.

All subgraphs satisfy one independence and five dependence relations:

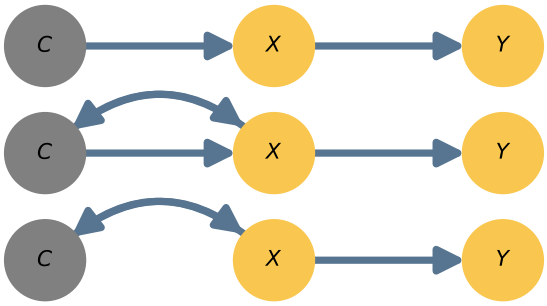

Figure 15: All mixed graphs in which LCD can infer the relation $X \to Y$

1. $C \perp\!\!\!\perp Y \,|\, X$
2. $C \not\perp\!\!\!\perp Y$
3. $C \not\perp\!\!\!\perp X$ (optional)
4. $X \not\perp\!\!\!\perp Y$ (optional)
5. $X \not\perp\!\!\!\perp Y \,|\, C$ (optional)
6. $C \not\perp\!\!\!\perp X \,|\, Y$ (optional)

Cooper [1997] proved that relations 1, 3, and 4 are sufficient to infer the causal relation $X \to Y$, by enlisting all possible subgraphs. Versteeg and Mooij [2019] only used relations 1 and 2, which are also sufficient.

Most LCD implementations only check two or three relations. However, if assumptions are violated or the dataset is small, some nonexistent causal relations might be inferred. One can sacrifice recall for precision by testing some or all of the remaining relations. Cooper [1997] warns specifically that the independence relation ($C \perp\!\!\!\perp Y \,|\, X$) is vulnerable to faithfulness violations, and suggests testing the second relation ($C \not\perp\!\!\!\perp Y$) as well. Triantafillou et al. [2017] aim for high precision and test all six relations.

A common choice of independence test is the two-tailed Fisher z-test [Fisher, 1924], which tests if the partial correlation is zero. In the application of this thesis, this test may be limited, since the context variable is constructed to have only two discrete values. As an intuitive example, we test the relation $C \not\perp\!\!\!\perp X$. The mean of X given $C = 0$ happens to be close to the mean given $C = 1$. This makes it very hard to significantly show a correlation between C and X.

As a potentially better alternative, we use the mean-variance test that is used by Versteeg and Mooij [2019], testing both the means and the variances across context values. The example given above would not fool this test, because there is a difference in variance. Regardless of this intuition, it should be noted that Versteeg and Mooij [2019] show hardly any difference in results due to the test.

When the test rejects the null hypothesis, we conclude (conditional) dependence between the variables. Reversely, when the test fails to reject the null

hypothesis, we conclude (conditional) independence. We accept this dubious method, because it is simple and common in the causality field. We further follow the work of Versteeg and Mooij [2019] by choosing a different significance level for testing the dependence (2) and conditional independence (1) relation. The conditional independence test is made more strict by correcting for multiple testing, with an $\alpha_{indep} = \frac{0.01}{6182}$. It is not as straightforward to correct the test for the dependence relation, since we draw a conclusion when we fail to reject the hypothesis. Therefore, the significance level is kept at $\alpha_{dep} = 0.01$.

5.2 Context Design

The Kemmeren et al. [2014] dataset does not provide clear context variables that are known to be exogenous. Therefore, the modeler can determine how to construct a context variable. Any context that is not informed by the values of the data is allowed, but some designs may be more productive than others. Recall that LCD is sound but not complete. A bad choice of context could result in a low recall.

We can look at context design from the perspective of three sufficient LCD conditions. $X \not\perp\!\!\!\perp Y$ says that LCD only considers relations where dependence between X and Y is shown. This condition is irrelevant for context design. $C \not\perp\!\!\!\perp X$ means that we should choose the context such that it is expected to depend on X. In the case of a single binary context variable, we want the distribution of X to be different depending on the value of C. Lastly, $C \not\perp\!\!\!\perp Y|X$ tells us that any dependence between the context and our potential target Y should disappear once we know the value of X.

Any background knowledge about the data can be used for the context. However, the sparsity of the data makes this task complicated. We would like to encode the target gene of interventions in a categorical context variable. However, there would only be one data point per context value, rendering independence testing impossible. The challenge is to generalize this information.

Original Context Design

The context used by Versteeg and Mooij [2019] is the ultimate generalization of the intervention target information. They introduce a single binary context variable that encodes if the data point is interventional or observational.

Generally, when a gene is knocked out in the system, a different SCM is induced with a different distribution of the variables values. However, the effect of single interventions is typically restricted to a small number of genes that are significantly affected. When we are interested in the effects of some variable X, there may be some intervention data points where X deviates from its normal value due to intervention on itself or its ancestors. However, the significance of these data points may be obfuscated by the large number of intervention data points where X has a normal value. The clusters for $C = 0$ and $C = 1$ could be hardly distinguishable.

Order-based Context Design

We hypothesize that a more specific context is more effective. Given a variable pair (X, Y), the original context makes a distinction whether there is an intervention anywhere in the system. Most of these interventions don't even affect X or Y. We would like a context that makes a distinction between an environment with an intervention that affects X, and an environment without such intervention.

To construct a context based on this idea, we require two adaptations. First, for each variable X we have a separate context variable C_X that we use to test the relation $X \to Y$. Second, we need to estimate whether a given intervention is expected to affect X.

The approach in this thesis is to estimate a causal order of the variables. Assuming acyclicity, interventions on genes later in the order than X cannot affect it. Thus, we set the context to 1 only if the intervention target is earlier in the order than X, or if X is intervened on.

Formally, we look at variable pairs X_i, X_j, and test the causal relation $X_i \to X_j$. For each variable X_i, we have some observation data points and intervention data points, such that $X_i = (X_i^O, X_i^I)$. The elements of the intervention data points $(X_i^I)_k$ are indexed according to the variable that is intervened on. For example, the intervention on X_i itself is $(X_i^I)_i$. The inferred variable order is represented by a permutation π, such that π_i indicates the position of variable i. We now construct the context $C_{X_i} = (C_{X_i}^O, C_{X_i}^I)$ as follows.

$$
\begin{aligned}
(C_{X_i}^O)_k &= 0 \\
(C_{X_i}^I)_k &= \begin{cases} 1 \text{ if } \pi_k \leq \pi_i \\ 0 \text{ if } \pi_k > \pi_i \end{cases}
\end{aligned}
$$

Before we move to the experiments detailed in the next chapter, there is one problem that remains to be tackled. When we wish to predict the effects of variable X_i, we cannot use the intervention data point in which X_i is intervened on. In fact, we use this data point to evaluate our prediction. However, the order inference algorithm requires this data to determine the position of X_i in the order. Therefore, we can only use this algorithm to infer the order of the other variables $X_{\setminus i}$, and need a separate algorithm to infer the position of X_i.

5.3 Estimating Variable Position in an Order

Given the order of variables $X_{\setminus i}$, we wish to estimate the position of variable X_i. For this task we may use all intervention data, except the effects of the intervention on X_i. Some useful information may be found in the effects of the other variables on X_i. The intuition is that variables that significantly affect X_i, should be earlier in the order.

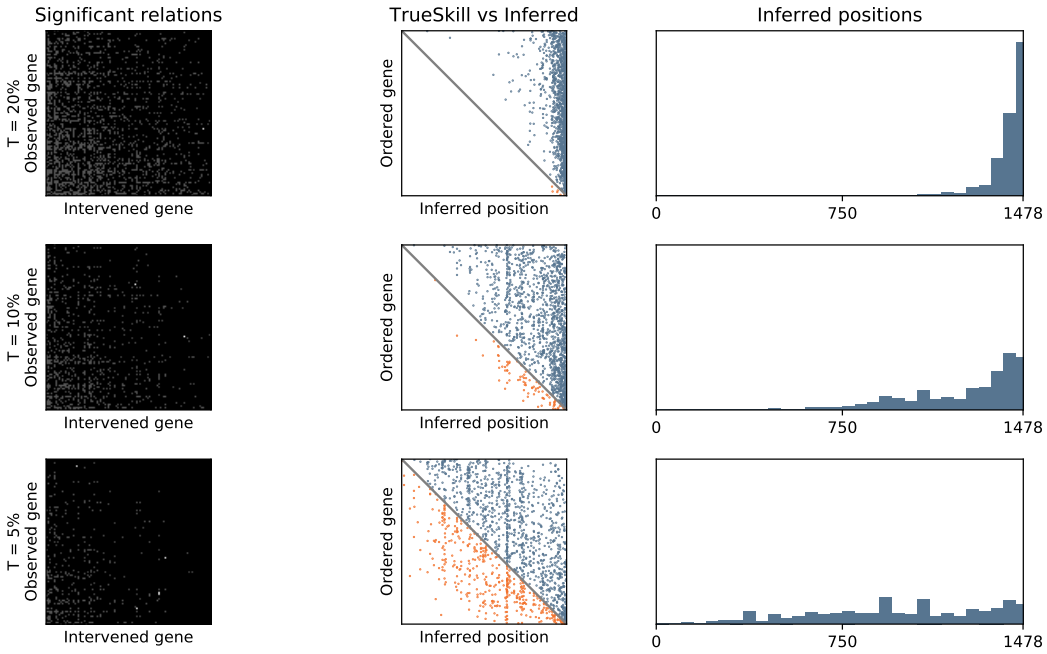

Figure 16: Analysis of test gene position inference for three different thresholds on the absolute expression values.

We need to determine when we consider these effects to be significant. We construct a binary ground-truth in the same way as before, by subjecting the intervention table to a threshold. However, since the task is different we may choose a different threshold. We decide on the threshold using a simple analysis.

The task of estimating the position of a variable X_i is not trivially defined. In our analysis, we first infer the order including X_i. We then remove it from the order and use the available data to estimate its position again. This allows us to evaluate the estimated position by comparing it to its original position. Note that the order of the remaining variables could be different if we would infer it without including X_i. However, we assume that the influence of including X_i is minimal on the order of all other variables.

The algorithm works as follows. We apply a threshold T to the absolute values of the intervention table $\boldsymbol{X} \in \mathbb{R}^{N \times N}$. Like before, $\boldsymbol{X}_{ij}$ is the relative expression level of gene i in the experiment with gene j knocked out. This yields a binary ground-truth of significant effects of each intervention. For each X_i, we look up which interventions affect it significantly according to this ground-truth. Using the inferred order π, we find the position of the latest significant cause of X. The estimate of the position of X_i in this order is directly after this latest cause. This way, all significant causes are earlier in the order.

Figure 16 shows the results of the analysis for ground-truth thresholds 5%, 10%, and 20%. The left column shows the significant relations given the

threshold, where the variables are ordered by the inferred order. Visually, for each gene on the rows, we look up the significant relation furthest to the right. The middle column shows only these last relations, and thus the estimated position for each gene. Note that if the order were correct, and no other order was possible (e.g. if the variables were on a Markov chain), the perfect positions would follow the gray diagonal line. Orange positions are estimated earlier in the order, blue positions later. The right column shows the distribution of estimated positions.

Threshold 5% yields the nicest distribution of estimated positions. Many variables are placed at early positions compared to the results for higher thresholds. However, many of these early positions are earlier than their true position in the inferred order. This is not necessarily wrong, but we cannot verify this. If we make a lot of mistakes, the context becomes meaningless. We do not want to risk this. On the other hand, threshold 20% estimates most positions to be far in the order. This means we approach the original context definition, since most interventions will be earlier than the estimated position of most genes. We choose threshold 10% for our experiments as a good middle way.

6 Experiments

6.1 Experimental Set-up

The Kemmeren et al. [2014] dataset contains the expression levels of 6.182 genes in 262 observation experiments, and as a result of 1.479 knock-out experiments. We describe this dataset as $\mathcal{D} = (\boldsymbol{O}, \boldsymbol{X})$ with observation data $\boldsymbol{O} \in \mathbb{R}^{6182 \times 262}$ and intervention data $\boldsymbol{X} \in \mathbb{R}^{6182 \times 1479}$. $\boldsymbol{O}_{ij}$ is the relative expression level of gene i in the j-th observation, $\boldsymbol{X}_{ij}$ is the relative expression level of gene i in the experiment with gene j knocked out. If we select only the intervention effects on genes that were also object of a knock-out experiment, we are left with the intervention table $\boldsymbol{X}^I \in \mathbb{R}^{1479 \times 1479}$.

Data Folds and Bootstrapping

The task is to predict per knock-out which genes are significantly affected, using the intervention data of the other knock-outs and the observation data. A leave-one-out experimental set-up would use all this data. Since variable preselection and order inference are time consuming, we choose a 5-fold train-test split set-up instead. The effects of each knock-out in a test fold are inferred from the remaining four train folds. In the spirit of this set-up, we also split the observational data, such that there is some variation in the observation data used per test fold.

From previous work on this dataset such as that of Versteeg and Mooij [2019] we know that only the most stable predictions may beat the baselines. We therefore apply the same bootstrapping method. We subsample the train data 100 times, sampling 50% without replacement. The predictions inferred on the subsamples are aggregated, allowing us to sort the predictions by confidence.

We compare two methods of aggregating predictions. A *discrete prediction* counts the number of times a relation was predicted, thus yielding a score between 0 and 100. A *continuous prediction* estimates the confidence per prediction using the necessary condition of LCD that $C \perp\!\!\!\perp Y$. The score is the sum of $-\log p_{C \perp\!\!\!\perp Y}$.

The results of both aggregations are very similar. In the main text of this thesis we will discuss the discrete method, because it is more insightful. When the p-value gets close to 0, the continuous score is capped. In practice this means that the strongest predictions mainly have capped scores and are comparable to the discrete method again.

Evaluation

The intervention data is transformed to obtain a score that indicates the significance of the causal relation. In evaluation, a certain percentage of highest scoring relations is taken as **ground-truth**.

We compare two transformations of the intervention data to score the significance of each relation. The *standardized score* $\boldsymbol{S^S} \in \mathbb{R}^{6182\times1479}$ is taken directly from the work of Versteeg and Mooij [2019], such that we can compare our results. The effects on gene j are normalized using the empirical mean and standard deviation of that gene in the observational data. The absolute value is taken to consider both upregulation and downregulation:

$$S^S_{ij} = \frac{|X_{ij} - \mu_j|}{\sigma_j}$$

This score prioritizes that every gene should occur as an effect in any ground-truth set, when it is possible that some genes never occur as a cause.

A large standard deviation in the expression of some gene is likely to mean that the gene is significantly affected by many other genes. We introduce a simpler *absolute score* that does not imply a prioritization of including all genes as an effect. This is the same score used in the order-based LCD algorithm:

$$S^A_{ij} = |X_{ij}|$$

Furthermore, we consider two **filters** on the set of predictions. Kemmeren et al. [2014] mention that the knock-out genes were selected based on their role in regulating gene expression. This implicates that the group of knock-out genes differs qualitatively from the remaining genes. Therefore, we consider a filter of only effects on those genes.

For the second filter we use the inferred order and gene positions. We only evaluate relations that explicitly comply to the inferred order and positions. Note first that this leaves only a small number of ground-truth relations, since the distribution of inferred positions is skewed to the right, having relatively little candidate descendants. This second filter might provide some insights on the performance on those causal relations that our LCD algorithm is most explicitly concerned with.

6.2 Method Implementation

The order-based LCD algorithm is implemented as an adaptation of the LCD algorithm by Versteeg and Mooij [2019]. Where they use the original context that distinguishes between observation and intervention data, we insert our order and position inference algorithms to construct an order-based context for each test variable. We compare our results to the same baselines used in their paper. The method implementation and baselines are shortly discussed here.

All methods start with variable preselection by L_2-boosting [Schapire et al., 1998]. This preselection saves significant computation time. It works so well, that it is taken as a baseline method. Versteeg and Mooij [2019] show that performance of LCD is improved when this preselection is applied, so at least for the strongest predictions we do not sacrifice performance.

L_2-boosting selects at most 8 test variables that are most predictive for each effect variable, using the joint observation and intervention data. For LCD testing a relation $X \to Y$, this means that at most 8 causes X from the test split are considered for each effect Y. L_2-boosting iteratively applies least squares regression to the residuals, eliminating variables that are least predictive.

To compute the order-based context, our LCD algorithm first infers an order on the variables in the train split using TrueSkill. Next, the position of each gene X in the test split is individually estimated. From the order and the gene position a context C_X is computed and the relation $X \to Y$ is tested for all Y's of which X is a preselected variable.

In the rare case that there are hardly any data points with $C_X = 1$, we skip the tests with X. We do this when there are less than 10 of such data points. If we would infer more spread out gene positions, this setting may need more investigation. Our preliminary results showed that the performance decreases slightly with a higher restriction of 30 and 300.

For comparison, we show the LCD and ICP results from Versteeg and Mooij [2019] that use the mean-variance test. The performance of their LCD is shown with and without L_2-boosting, to make it clear that boosting helps. Their L_2-boosting baseline results are also shown. Note that we evaluate these results not only on their standardized ground-truth score, but also on an absolute score, resulting in somewhat different graphs.

L_2-boosting is implemented with the `glmboost` function in the `MBoost R` package.[14] ICP is implemented using the `InvariantCausalPrediction R` package.[15]. For the order inference algorithm, we use the `trueskill Python` package.[16]

[14] https://cran.r-project.org/web/packages/mboost/index.html
[15] https://cran.r-project.org/web/packages/InvariantCausalPrediction/index.html
[16] https://pypi.org/project/trueskill/

7 Results and Analysis

7.1 Receiver Operating Characteristic Curves

We first analyse the performance of order-based LCD on the Kemmeren et al. [2014] dataset using the receiver operating characteristic (ROC) curve. This graph shows the number of true positives versus false positives of a binary classifier as a prediction threshold is varied. Good methods rise quickly and make many good predictions before starting to make mistakes. Specifically, we fix a binary ground-truth that defines the true ancestral relations between genes. The aggregated scores resulting from the bootstrapped experiment are thresholded to select predictions that are tested to the ground-truth.

Compared to Other Methods

In the top row of Figure 17 we show the ROC curves of order-based LCD, ICP which is considered the best method on this data, the L_2-boosting baseline and a random baseline. We computed the curves for the 10%, 1%, and 0.1% strongest relations according to the absolute score. The discrete predictions of the methods are used. The ROCs of other combinations of these settings, including standardized scores and continuous predictions, can be found in Appendix B. The conclusions hold for all settings.

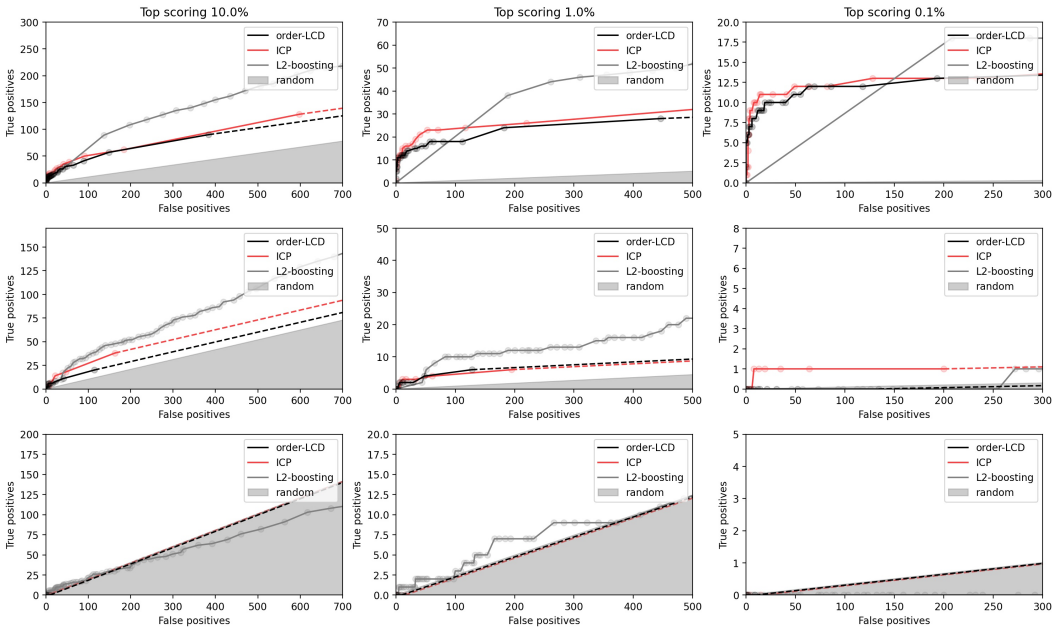

Figure 17: ROC curves of order-based LCD compared to other methods. Columns use different ground-truth thresholds, rows different subsets of the relations that are evaluated. A dashed line indicates that a method resorts to random guessing.

Order-LCD and ICP beat the L_2-boosting baseline in their most confident predictions. The first 10 predictions are generally very good. After this point

the methods quickly start making many mistakes. This indicates the hardness of this prediction task.

Most importantly, we see that order-based LCD approaches the performance of ICP. This is a desirable result, because order-based LCD is a more efficient method than ICP and may therefore be used as a fast approximation. Note that this conclusion was initially drawn about LCD by Versteeg and Mooij [2019], which inspired the adaptation of the method in this thesis. A comparison to this LCD method follows in the next subsection.

Besides these ROC curves, two more settings were tested in which only a subset of ancestral relations is considered. These graphs can be found in the second and third row of Figure 17. The *intervention table* ROC curve is only concerned with relations between genes that occur as knock-out in the data. These predictions may be considered as a different task, as this subset of genes may be qualitatively different from the rest since it was chosen by Kemmeren et al. [2014] based on some biological criteria.

We may expect that order-based LCD behaves differently on this subset of ancestral relations. The inferred order is purely based on this intervention table, and the order inference algorithm is chosen based on analysis of this table. However, the results are not qualitatively different. Only the first few predictions of order-based LCD and ICP beat the L_2-boosting baseline.

The *order-compliant relations* ROC curve contains only those relations that are explicitly not in violation of the order inferred by order-LCD. This small subset of ancestral relations may be used to test whether order-LCD performs relatively well on those relations that are most in line with our hypothesis. Do the genes comply to some implicit order, and can this be used to inform the context variable of LCD? Unfortunately, very little predictions in this subset are made by order-based LCD or ICP. It may be that this small subset of relations happen to be harder to prove, or that order-based LCD fails to show the relations that it was designed for. The fact that the L_2-boosting baseline performs about as good, or worse than the random baseline on this subset supports the first explanation.

Compared to LCD Methods

Figure 18 shows ROC curves of all methods related to LCD using discrete predictions and an absolute ground-truth. ROC curves with the other settings and evaluated on subsets of the relations are in Appendix B.

The most important result is that the ROC curves of the original LCD and our order-based LCD are very similar. This means that we failed in our attempt to improve on the original LCD by informing the context with some gene ordering. We analyse the small difference between these two methods in the end of this section.

Figure 18 also shows the results of an ablation study. We randomized the inferred order, the inferred gene positions, and both together, to investigate to what extend these additions to the original LCD contribute to the performance

of order-based LCD. Although order-based LCD is not beaten by its randomized versions in the strongest predictions, the difference is never more than a few true positives. We have to conclude that we failed to show a significant effect of our order inference and our position inference. This may show that the approach is not effective on this dataset, or that order inference or position inference should be implemented differently to benefit LCD. Since position inference was not as extensively developed in this thesis, the problem may lie there.

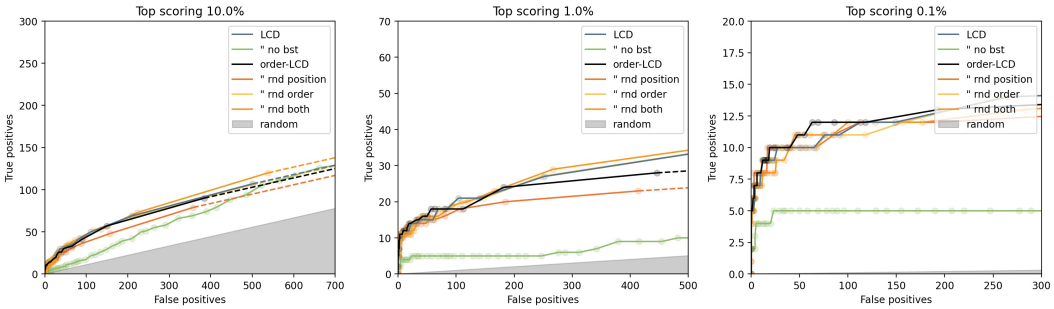

Figure 18: ROC curves of order-based LCD compared to other LCD methods. Columns use different ground-truth thresholds. A dashed line indicates that a method resorts to random guessing.

7.2 Accumulated Regression Deviation

We provide another approach to evaluate the performance of the algorithms to show that our conclusions extend beyond binary ground-truths and ROC curves. When some method predicts an ancestral relation, we use the data in the train split to predict a precise expression value of the knock-out.

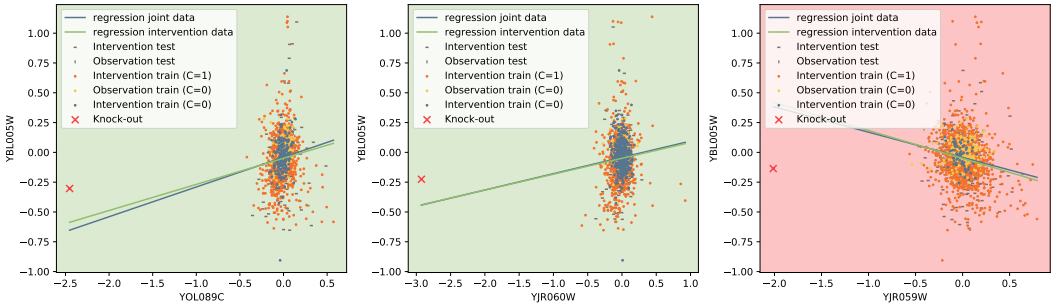

Figure 19: Visualization of regression deviation. Linear regression is applied to the joint or intervention train data. The deviation is the vertical distance between the regression line and the red knock-out cross.

Figure 19 shows how this is computed. Using the joint data or only the intervention data, we apply linear regression and use it to predict a deviation at the knock-out value of the cause gene. Note that we are cheating a little

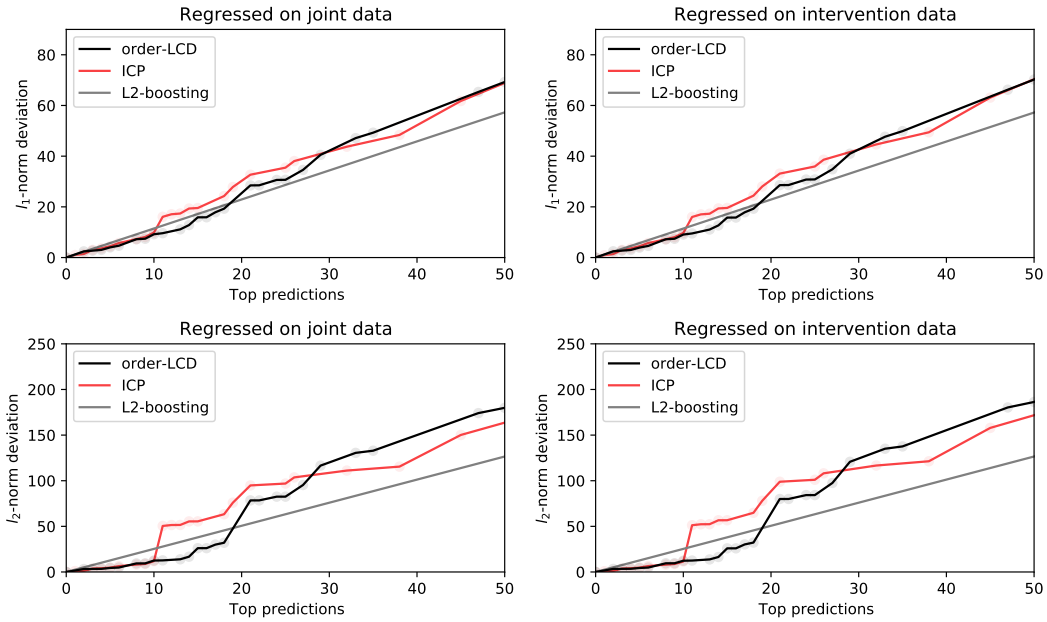

Figure 20: Accumulated regression deviation of order-based LCD compared to other methods. Columns use different data to regress on, rows a different norm to add the accumulated deviations.

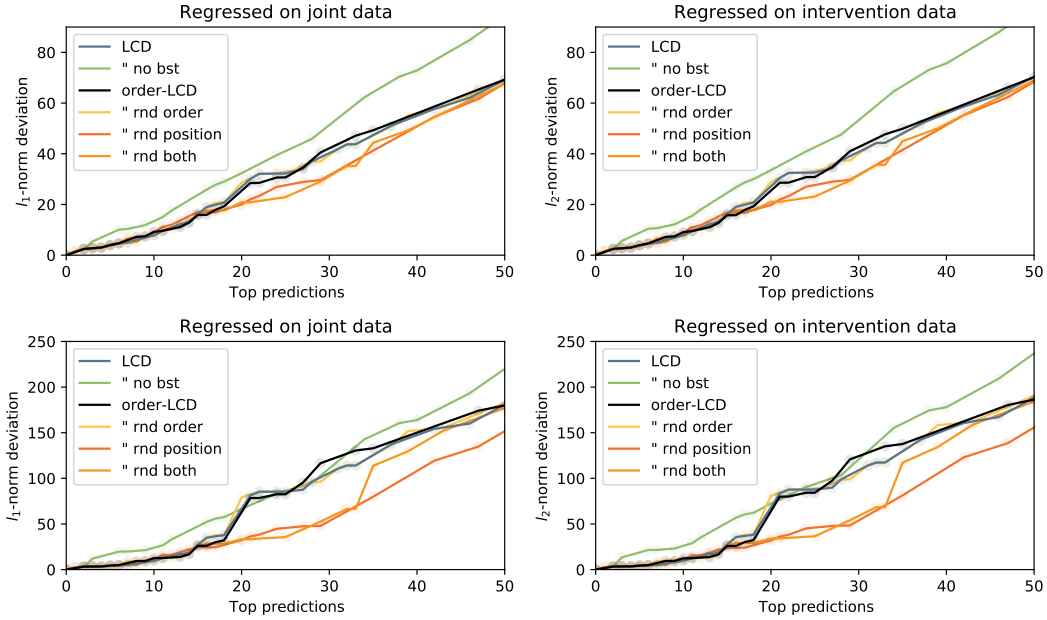

Figure 21: Accumulated regression deviation of order-based LCD compared to other LCD methods. Columns use different data to regress on, rows a different norm to add the accumulated deviations.

here, since this knock-out value is not in the train split. If the results of this approach were great, we could redefine the metric to see if the results are robust. For example, we could assume a standard knock-out value of about -2.5.

To compare the performance of the most confident predictions of different algorithms, we create accumulated regression deviation graphs. These accumulate the deviations of the top discrete predictions using the l_1-norm or l_2-norm.[17] Better methods have graphs with low accumulated deviation for the top predictions.

Figure 20 shows the graphs for order-based LCD compared to ICP and the L_2-boosting baseline. Again, both methods beat the baseline for the strongest predictions in all settings. Interestingly, order-based LCD seems to take longer to degrade than ICP. This result was not further investigated in this thesis.

The graphs comparing the LCD methods is shown in Figure 21. In the top 20 predictions we see hardly any difference, besides that boosting improves performance. Generally, we again struggle to see a significant difference between the original LCD and order-based LCD.

A surprising result shows up when we consider the ablation experiments. Further in the graph, it seems that randomizing the inferred positions leads to a better performance. Note that we decided to make position inference relatively strict, by putting many genes near the end of the order. The randomized position version indicates that we may expect better performance on this metric if we infer the positions more spread out.

A last remark is that linear regression is related to the independence tests that are used by the algorithms. In the accumulated deviation graphs we observe that there is hardly a difference between regression on joint data and intervention data. This shows that adding observation data changes very little about the regression. A new task could be defined in which a subset of the intervention data is selected for regression. Using the same intuition from this thesis, we may hypothesize that selecting the interventions on ancestors yields a better prediction of knock-out expression.

7.3 Comparison of Order-Based LCD with Original LCD

Specific examples

We investigate qualitatively how the predictions of order-based LCD are related to those of the original LCD. Figure 22 shows the six strongest predictions of order-based LCD that are also predicted by LCD, the top 25 can be found in Appendix B. It can be observed that these relations are also strongly predicted by LCD, and most are correct. The only relation that is wrongly predicted by both methods obtains a higher score from the original LCD, which gives some hope for the order-based method. All relations have relatively little interven-

[17] The l_1-norm is the sum of absolute values, the l_2-norm is the sum of squared values.

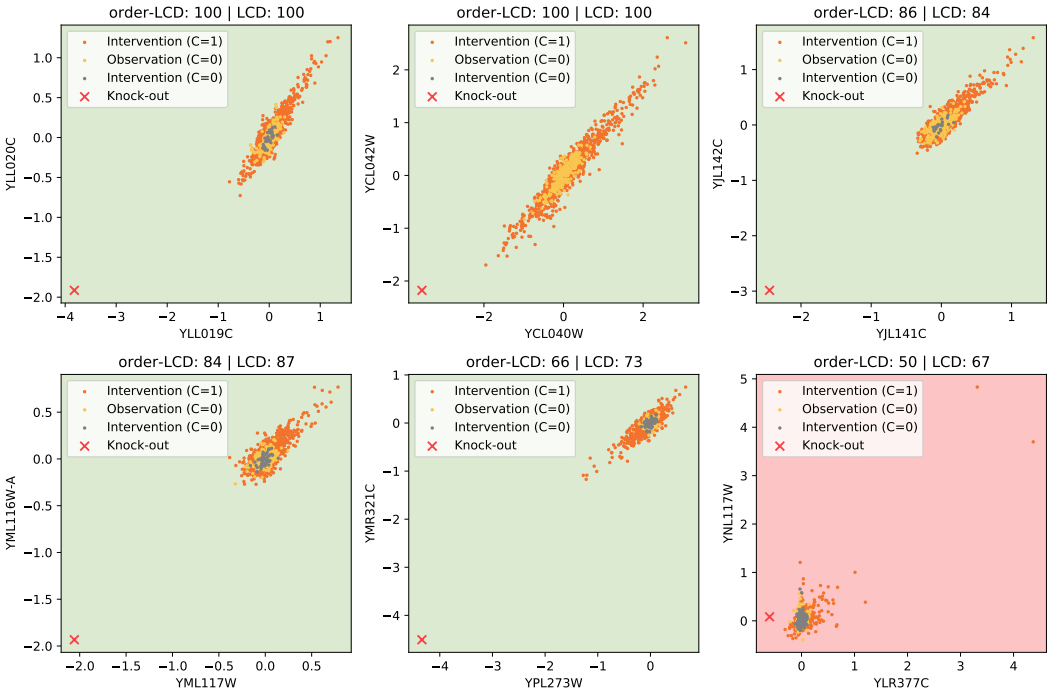

Figure 22: The six strongest predictions by order-based LCD that are also predicted by LCD. The precise scores are shown above the figures. The background color indicates whether the relation is true according to the 10% threshold.

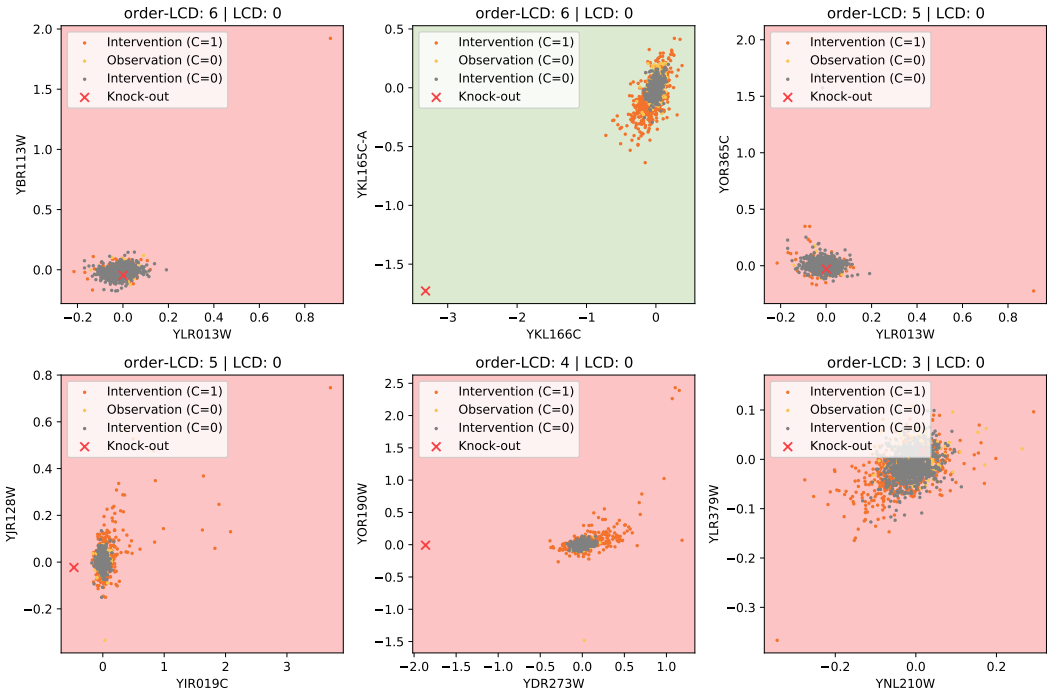

Figure 23: The six strongest predictions by order-based LCD that are not predicted by LCD. The precise scores are shown above the figures. The background color indicates whether the relation is true according to the 10% threshold.

tion data that obtains context value 0, which means that both methods are very similar. In one case, none of the interventions obtain this context value and the methods are effectively identical.

More interesting are the strongest order-based LCD predictions that are not at all predicted by LCD. The top six is shown in Figure 23, and the top 25 in Appendix B. We would hope that the methods behave differently, and that order-based LCD improves over LCD by making some entirely new, confident predictions. This is not what we see. The strongest exclusive prediction has merely a score of 6/100. Interestingly, these relations assign many intervention points the context value 0, and are in that way able to make predictions where the original LCD cannot. Unfortunately, most of these exclusive relations are also wrong, even according to the weak 10% binary ground-truth.

General Similarity

Finally, we take a more general approach in the comparison of order-based LCD and the original LCD, and try to explain why we failed to show a clear difference between the methods.

Figure 24 shows that the distribution of prediction scores assigned by order-based LCD and the original LCD are very similar. Furthermore, only low scores are assigned to exclusive relations, that are predicted by one method but not by the other.

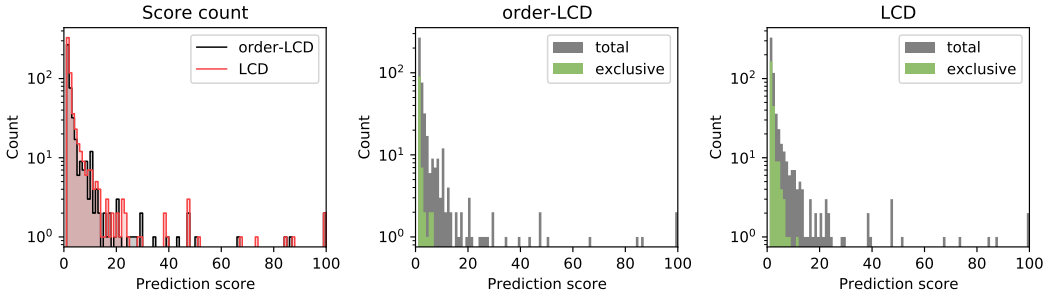

Figure 24: Comparing the distribution of discrete prediction scores of order-based LCD and the original LCD. Left: distribution of scores of both methods. Middle: distribution of order-based LCD, and distribution of the scores assigned to relations not predicted by LCD. Right: same, but comparing LCD to order-based LCD.

Figure 25 further shows that when we sort predictions from strong to weak, both methods end up with a very similar order. The strongest relations of order-based LCD are also the strongest ones of LCD. Weak relations of order-based LCD are also weak according to LCD.

Since we showed in many ways that both methods are very similar. We expect that this is the case because the position inference algorithm is chosen to be too conservative. In Chapter 5 we chose a middle way between the desirable properties of a very spread out distribution of inferred gene positions, and inferred positions that hardly precede the actual position according to

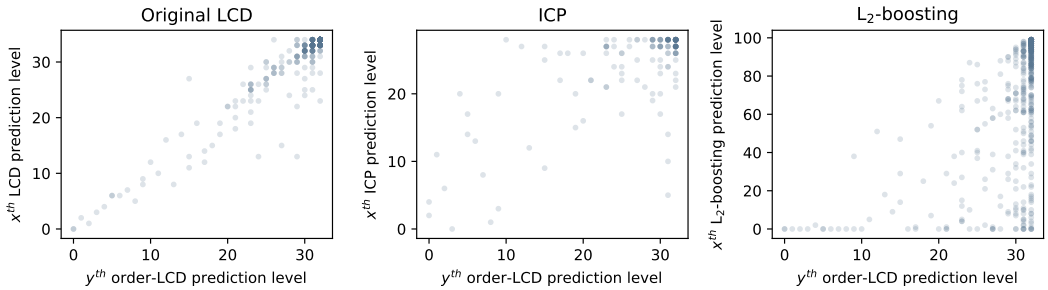

Figure 25: Comparing how predictions of order-based LCD and the original LCD, ICP, and the L_2-boosting baseline are related, going from strongest to weakest predictions. A prediction level contains relations that are assigned the same score, and can therefore not be ordered among themselves.

the order inference algorithm. The chosen middle way still resulted in most positions to be pushed to the end of the order, therefore keeping most of the context values 1. Note that pushing a position all the way to the back means that all context is one, rendering the method identical to the original LCD. Simply put, if we spread out the positions more, then the method becomes more different from the original LCD.

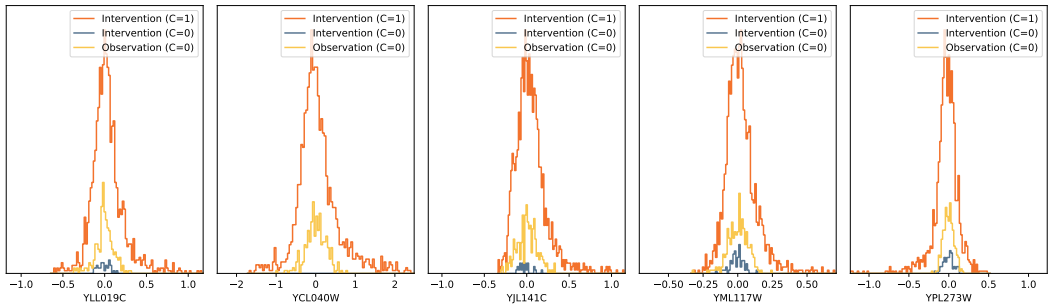

Figure 26: Distribution of the expression level of the cause genes in the strongest predictions of order-LCD. Clusters are made based on whether the data points are observation or intervention data, and based on the context value if the data point is interventional.

Figure 26 shows the distribution of the cause variable expressions in the data, for the five causes in the strongest predictions of order-based LCD.[18] The expressions are separated in three clusters, corresponding to the data source (observation or intervention) and the assigned context value. Ideally, in the cluster with context value 1, we collect those interventions that are affected by the cause variable. In this thesis we attempt to estimate this with an order in the genes. In the figure, we would then like to see that the extremes of the intervention data are assigned $C = 1$, whereas many values in the center are assigned $C = 0$. This would correspond to interventions that clearly affect the cause variable being assigned to the $C = 1$ cluster.

[18] Appendix B shows the top 25 for order-based LCD, original LCD, L_2-boosting and a random selection.

In Figure 26 we fortunately see that no data points in the extremes are assigned to $C = 0$. However, also very little central data points are assigned to $C = 0$. This means that the difference in the data used by the independence tests of order-based LCD and the original LCD are very similar, and we would indeed not expect large differences.

8 Conclusion and Outlook

After a decade that has shown unbelievable successes of statistical AI, purely statistical methods are reaching limits. This warrants a revaluation of symbolic AI, and sparks new interest in the interface of the two schools. Causality arises as a field with the ambitious goal to unveil cause-effect relations with a combination of deduction from datasets and induction from background knowledge.

We took a dataset of gene perturbation experiments and investigated an adaptation of LCD to discover causal relations between genes. Since we found that there was only limited feedback, we hypothesized that a causal order could inform the LCD context variable.

An extensive analysis of methods to estimate variable order from the data showed TrueSkill to be the most effective option. A straightforward method was then analysed to infer the position of a tested gene in the order. Finally, the order was used to construct a context variable for LCD, and order-based LCD was compared to baseline methods.

Several factors make the causal inference problem on this dataset complicated. The data is high-dimensional and very sparse. True labels are not available, so we need to construct some data-based ground-truth to do quantitative evaluation. The underlying biological system is complex. Only gene expressions are measured whereas the processes in the cell involve many other relevant variables, and there is quite some variation in the samples.

Consequently, out of thousands of true ancestral relations in the data, state-of-the-art methods can only be trusted in their top 20 predictions. Unfortunately, in this thesis we were unable to improve on this. Nevertheless, progress was made to develop causal inference methods based on an implicit order in variables, to analyse properties of these methods and justify parameter values, and to thoroughly analyse performance on prediction tasks.

8.1 Contributions

The main contributions of this thesis are listed below.

- Statistical properties of the Kemmeren et al. [2014] dataset were analysed, which can be used to inform causal discovery methods.
- A continuous metric was introduced to evaluate the task of estimating variable order in datasets with single sample interventions.
- Order estimation methods were thoroughly analysed on the Kemmeren et al. [2014] dataset, methods based on the binary ground-truth were found to be most effective.
- A new order-based LCD method was carefully designed and analysed.
- Failure to show significant improvement inspired new ways to analyse and compare inference methods in detail. These evaluation methods and graphs can be used to develop and investigate novel methods in the future.

8.2 Suggestions for Future Work

The insights of this thesis spark many new questions, that can be the basis of future research.

- **Further testing** of the order-based LCD method and easy extensions may show more promising results than we have in this thesis. The method can be generalized for datasets with more samples per intervention target. It may show different behavior compared to the original LCD when the task is easier. An experiment can also be done with ICP and the order-based context.
- An important step in further investigation of the hypothesis of this thesis, is to test the method on **simulated** data. If we fail to artificially construct a dataset on which the method works well, we should not expect it to work well on real-world datasets.
- There are enough parts of the order-based LCD algorithm that may be improved by **gradual development** of the method. The first thing to try is to infer more spread out gene positions, making the method more distinct from the original LCD.
- More **radical changes** of parts of the method are also interesting. We did not succeed to use the continuous data effectively to infer an order for example. Moreover, position inference is probably a weak link in the algorithm right now and may benefit from some more research. Perhaps order and gene positions can be inferred jointly as well.
- A **broader scope** may inspire interesting related methods. Perhaps we can infer a general partial order of variables, or infer specifically per cause variable a set of potential ancestors. It remains a big question to what extend we can make use of our knowledge of the intervention targets. Original LCD does not use this information at all, and order-based LCD only uses it coarsely.
- Lastly, we may investigate some **theoretical** properties of different context variables. Currently we assume that the context variable is exogenous to the system. However, in an indirect way we base it on the data itself. The order and gene positions are directly inferred from the data. What functional relations between the data and the context are allowed? What implicit assumptions do we make?

A Details of Order Inference Methods

This appendix lists the settings of the order inference methods of Chapter 4, and mentions the parameters that were experimented with in preliminary research to ensure good results. Consult the main chapter for details on the content of the algorithms.

Edmonds Algorithm

The Edmonds algorithm was implemented using the `Edmonds` function in the `networkx Python` package.[19] The algorithm has no parameters.

The sparse version of Edmonds selects the 10 strongest edges per node in the graph. We experimented with 50 nodes as well, which led to much more computational cost and no improvement in penalty ratio.

Evolution Strategy

No external code was used to implement the evolution strategies. The discrete and continuous ESs differ only in their fitness scoring. Continuous ES minimizes the penalty. Discrete ES maximizes the number of relations that do not violate the order, computing a ground-truth with an absolute threshold at 0.7, which is approximately the top 1%.

Solutions are represented as a permutation. The population size (number of solutions kept at one iteration) is 100. The population is randomly initialized.

At every iteration, we combine pairs parents to form a new generation. Parents are selected randomly. We use cyclic recombination for the main results. This algorithm is described in the chapter. We tried direct recombination as well. In this case, we randomly select some positions in the permutation, using a selection probability that we experimented with. The selected positions are moved to the other parent. The positions that are not switched to the other parent, are kept in the same relative order and filled in left-to-right. The same happens from the other parent to the first.

We used no mutation in the final ESs, because it increased computation time without clear benefits. However, we did experiment with two mutation algorithms. Mutation applies small random changes to individuals. We tried to swap pairs of positions in the permutation with some probability, that we varied. We also tried to invert random subsequences of fixed length with some probability. Length and probability were varied as well.

We experimented with the probability to apply recombination and mutation as well. In the ESs that we report, the recombination rate is 1 (applied to all parent pairs), and the mutation rate is 0 (never applied). This led to the best results.

Each iteration, an offspring of 400 individuals are created. The 100 with the highest fitness are selected as the new generation. In the case that the new

[19] https://networkx.org/documentation/stable/reference/index.html

generation is worse than the old one, we keep an elite of the best 5 individuals of the last generations, at the expense of the worst 5 in the new selection.

We experimented with different values for population size, offspring size, and elite size.

Every experiment was run for 200 iterations, and stopped if there were no improvements in fitness for 30 iterations. Experiments were often repeated 5 times. We tracked fitness, and an approximation of the average Hamming distance between pairs in the population as a proxy for diversity.

Sorted TrueSkill

The TrueSkill code was taken from the `trueskill` `Python` package.[20] A graph was constructed using the 20% absolute ground-truth.

Updates of the mean and variance variables are iterated 15 times. The final skill is computed per gene as $\mu + 3\sigma$. The algorithm uses a tie probability parameter of 0.05.

Sorted Social Agony

The Social Agony code was taken from the code of Sun's algorithm, but is also made available by the authors of the corresponding paper.[21] A graph was constructed using the 20% absolute ground-truth. The algorithm has no parameters.

Sorted PageRank

The PageRank code was taken from the `pagerank` function in the `networkx` `Python` package.[22] A graph was constructed using the 20% absolute ground-truth. The alpha parameter was set to 0.85. The maximum number of iterations was 100.

Sun's algorithm

The code of Sun's algorithm is shared by the authors.[23] A graph was constructed using the 20% absolute ground-truth. The parameters of the scoring methods are the same as in the three sorted methods above.

[20] https://pypi.org/project/trueskill/
[21] http://users.ics.aalto.fi/ntatti/software.shtml
[22] https://networkx.org/documentation/stable/reference/index.html
[23] https://github.com/zhenv5/breaking_cycles_in_noisy_hierarchies

B Additional Graphs

ROC Curves Compared to Other Methods

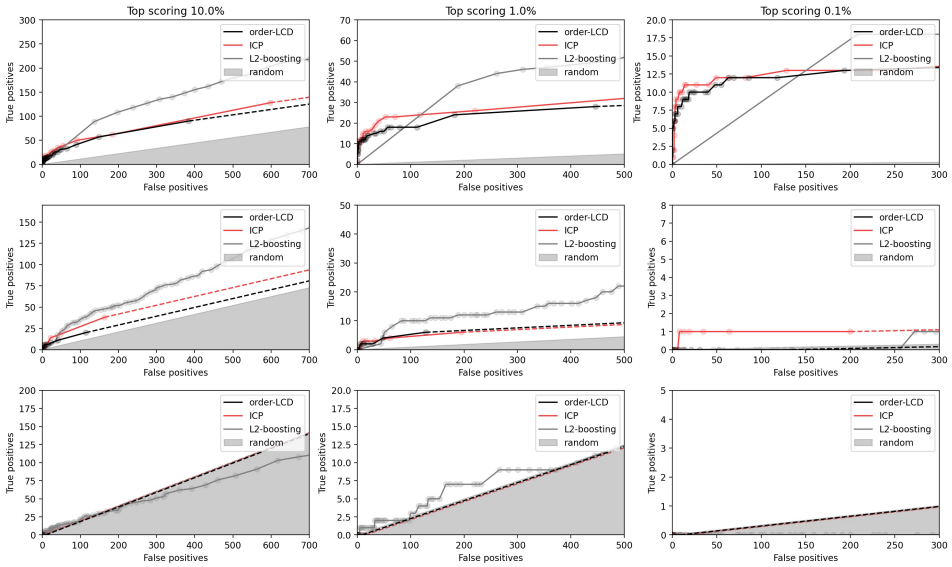

Figure 27: ROC curves of order-based LCD compared to other methods. **Discrete predictions** are evaluated on the **absolute ground-truth**. Columns use different ground-truth thresholds, rows different subsets of the relations that are evaluated. A dashed line indicates that a method resorts to random guessing.

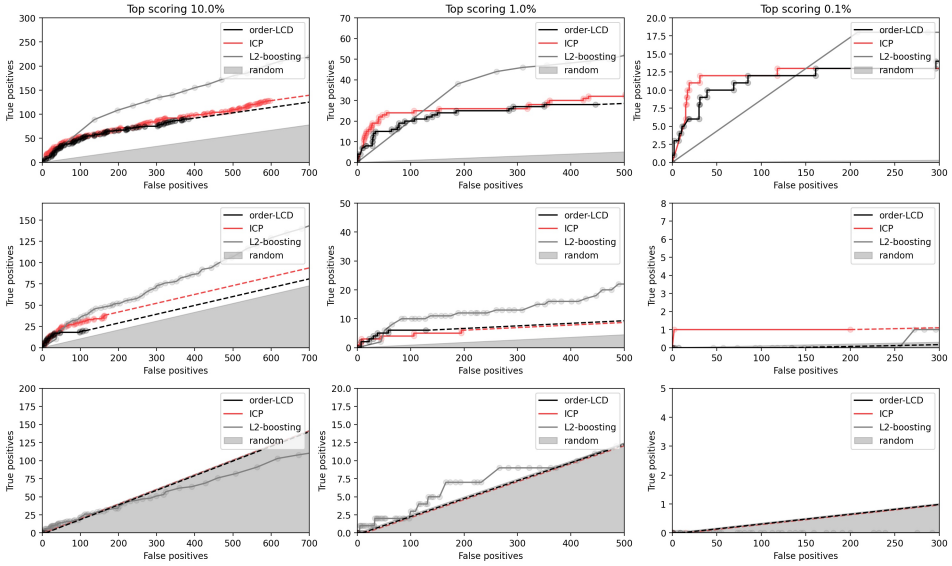

Figure 28: ROC curves of order-based LCD compared to other methods. **Continuous predictions** are evaluated on the **absolute ground-truth**. Columns use different ground-truth thresholds, rows different subsets of the relations that are evaluated. A dashed line indicates that a method resorts to random guessing.

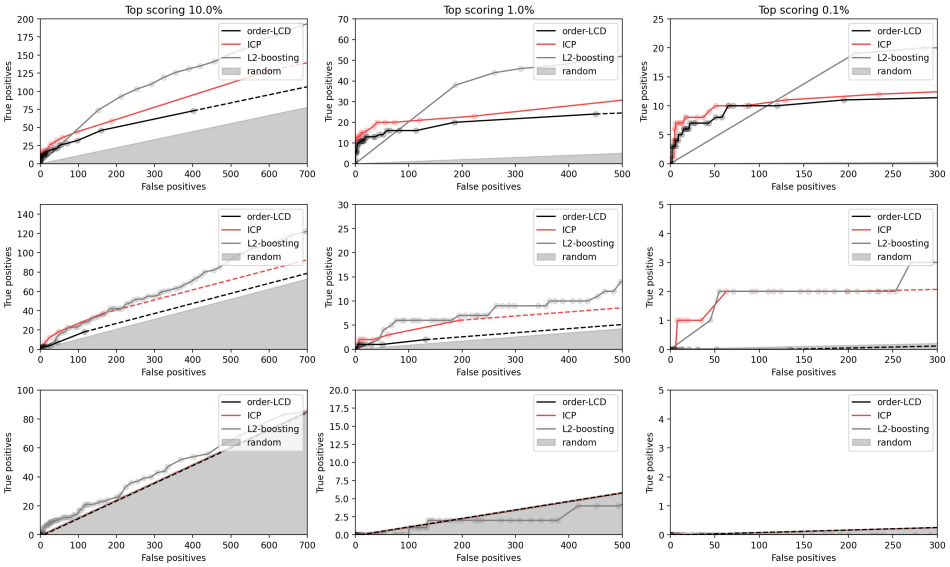

Figure 29: ROC curves of order-based LCD compared to other methods. **Discrete predictions** are evaluated on the **standardized ground-truth**. Columns use different ground-truth thresholds, rows different subsets of the relations that are evaluated. A dashed line indicates that a method resorts to random guessing.

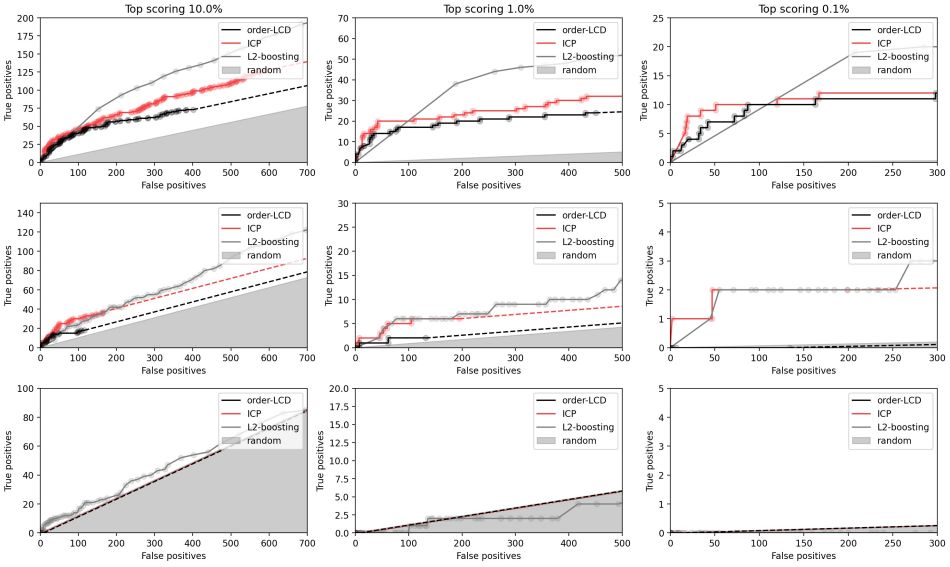

Figure 30: ROC curves of order-based LCD compared to other methods. **Continuous predictions** are evaluated on the **standardized ground-truth**. Columns use different ground-truth thresholds, rows different subsets of the relations that are evaluated. A dashed line indicates that a method resorts to random guessing.

ROC Curves Compared to LCD Methods

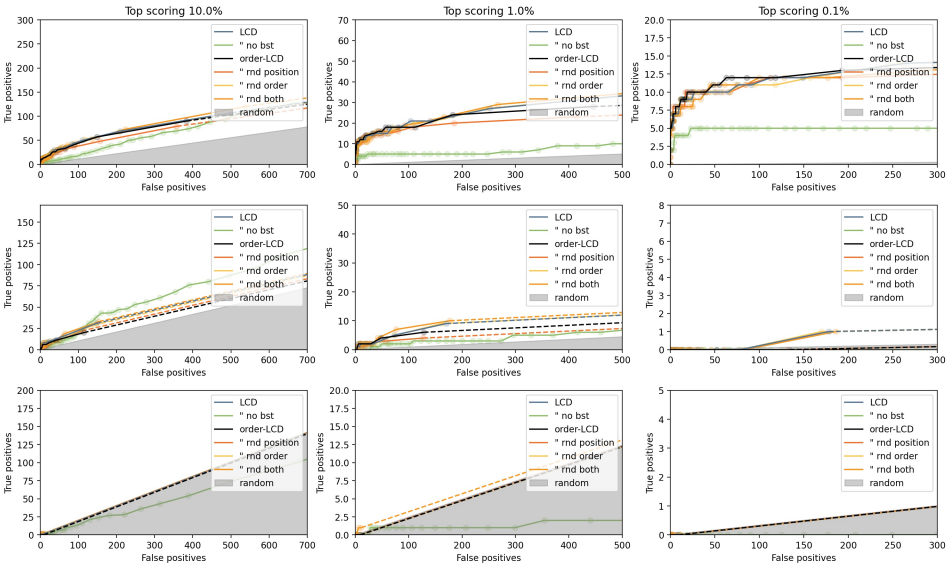

Figure 31: ROC curves of order-based LCD compared to LCD methods. **Discrete predictions** are evaluated on the **absolute ground-truth**. Columns use different ground-truth thresholds, rows different subsets of the relations that are evaluated. A dashed line indicates that a method resorts to random guessing.

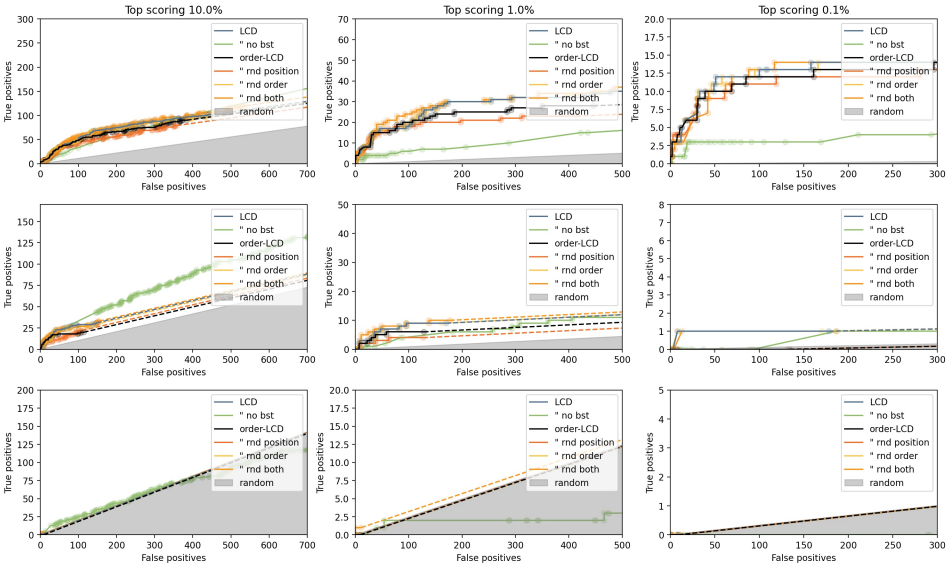

Figure 32: ROC curves of order-based LCD compared to LCD methods. **Continuous predictions** are evaluated on the **absolute ground-truth**. Columns use different ground-truth thresholds, rows different subsets of the relations that are evaluated. A dashed line indicates that a method resorts to random guessing.

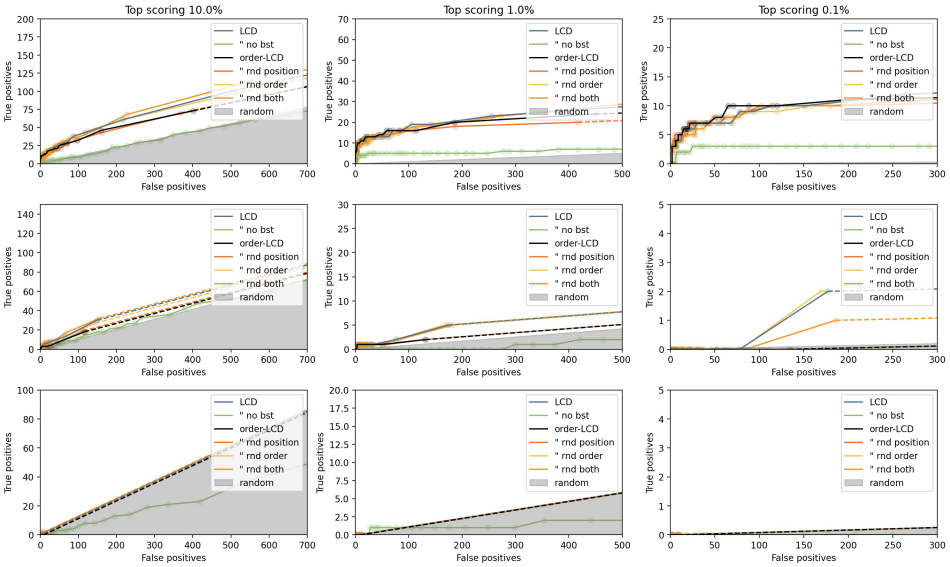

Figure 33: ROC curves of order-based LCD compared to LCD methods. **Discrete predictions** are evaluated on the **standardized ground-truth**. Columns use different ground-truth thresholds, rows different subsets of the relations that are evaluated. A dashed line indicates that a method resorts to random guessing.

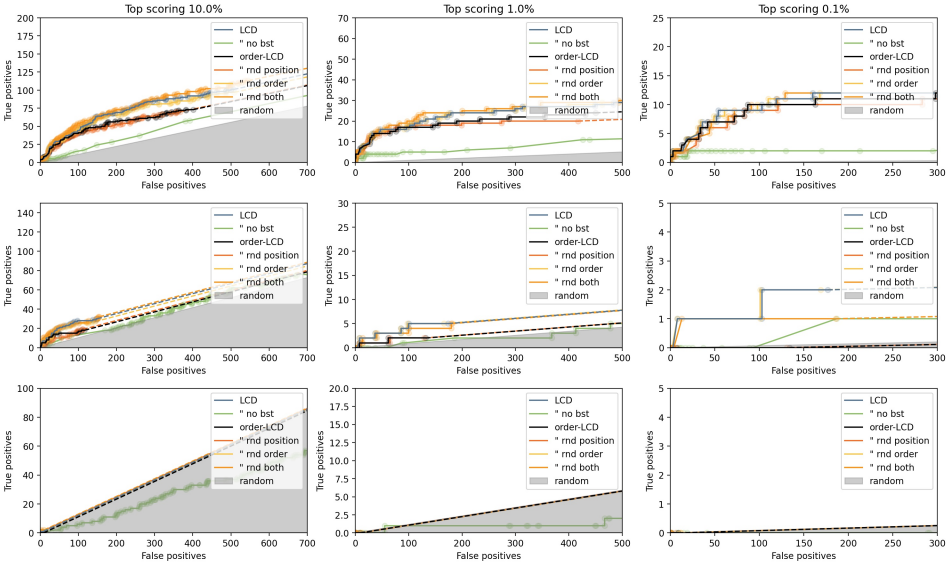

Figure 34: ROC curves of order-based LCD compared to LCD methods. **Continuous predictions** are evaluated on the **standardized ground-truth**. Columns use different ground-truth thresholds, rows different subsets of the relations that are evaluated. A dashed line indicates that a method resorts to random guessing.

Inclusive and Exclusive Order-Based LCD and LCD Relations

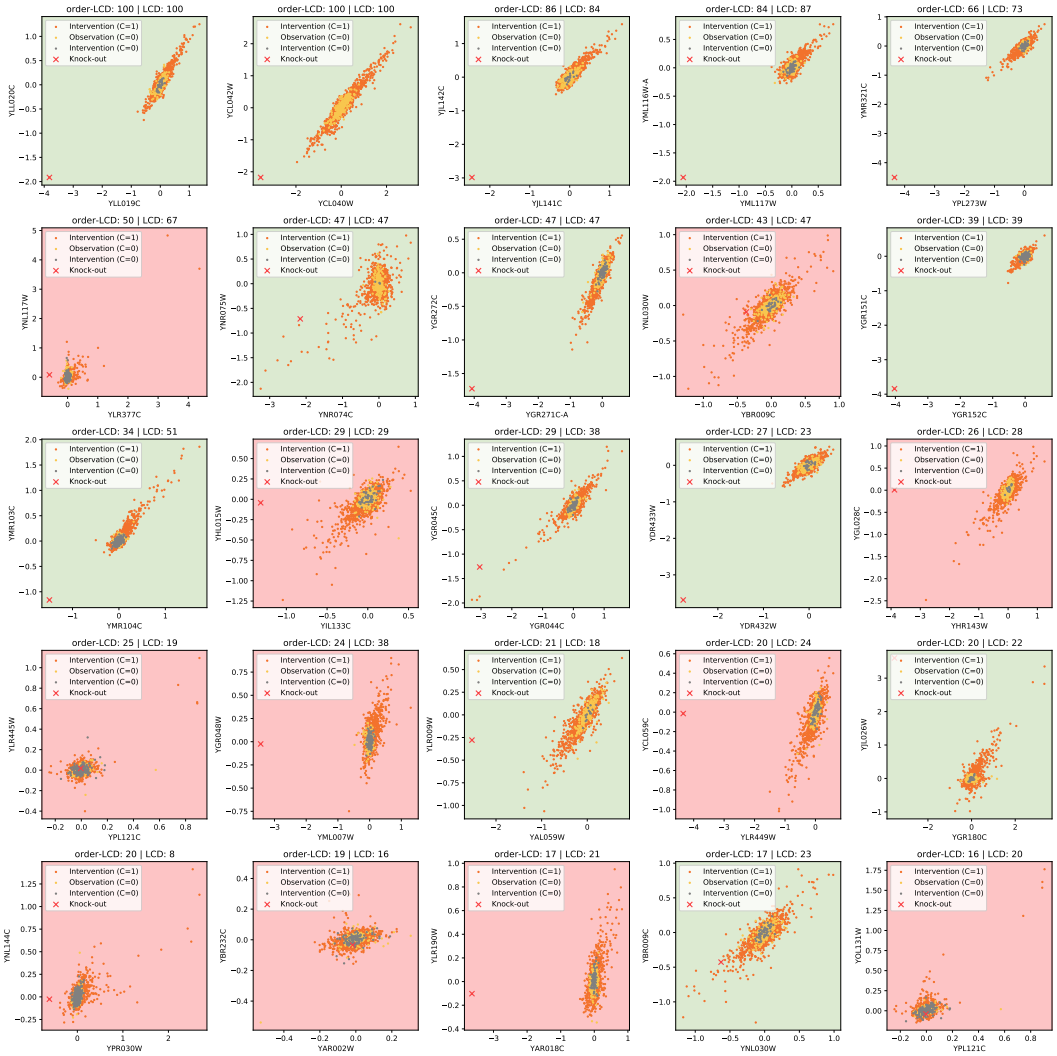

Figure 35: The 25 strongest predictions by **order-based LCD** that are also **included** in the LCD predictions. The precise scores are shown above the figures. The background color indicates whether the relation is true according to the 10% threshold.

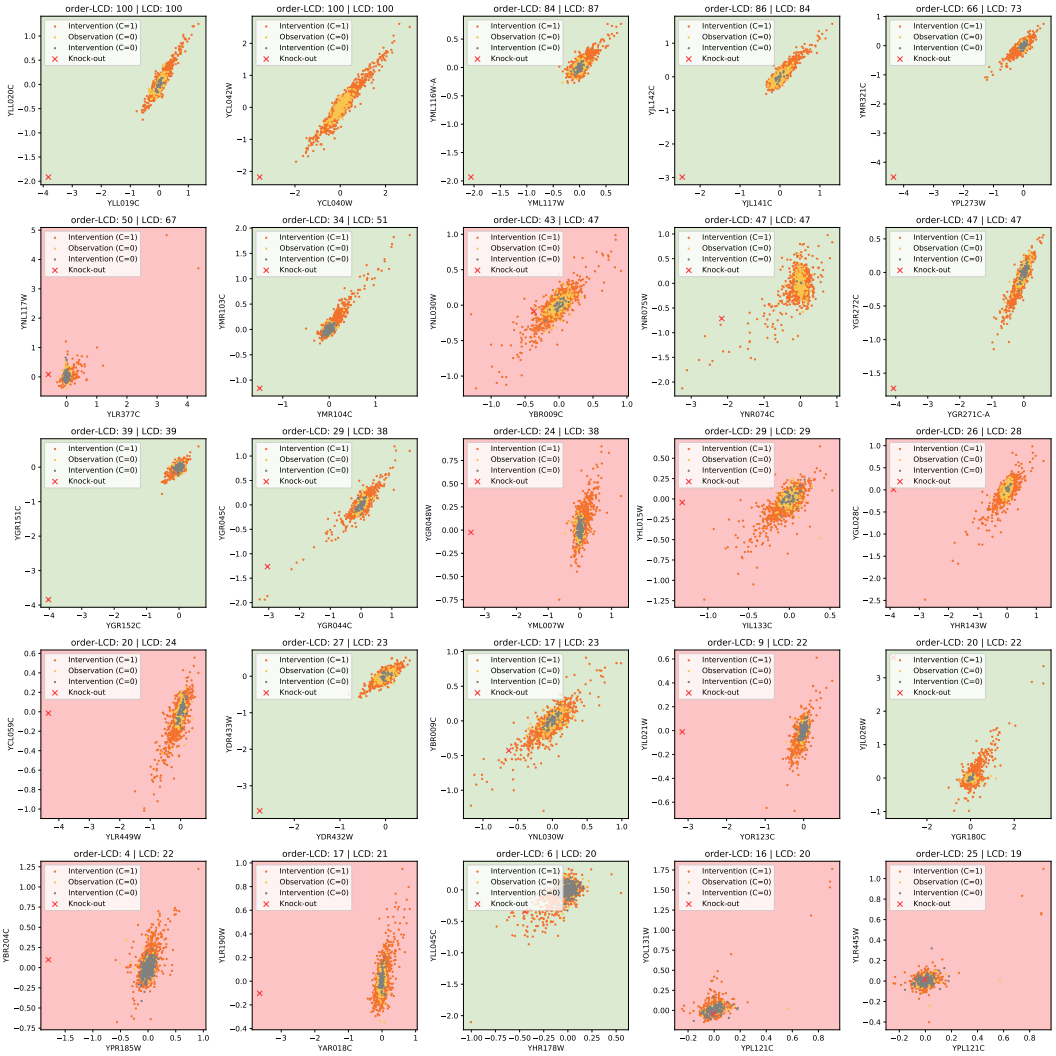

Figure 36: The 25 strongest predictions by **LCD** that are also **included** in the order-based LCD predictions. The precise scores are shown above the figures. The background color indicates whether the relation is true according to the 10% threshold.

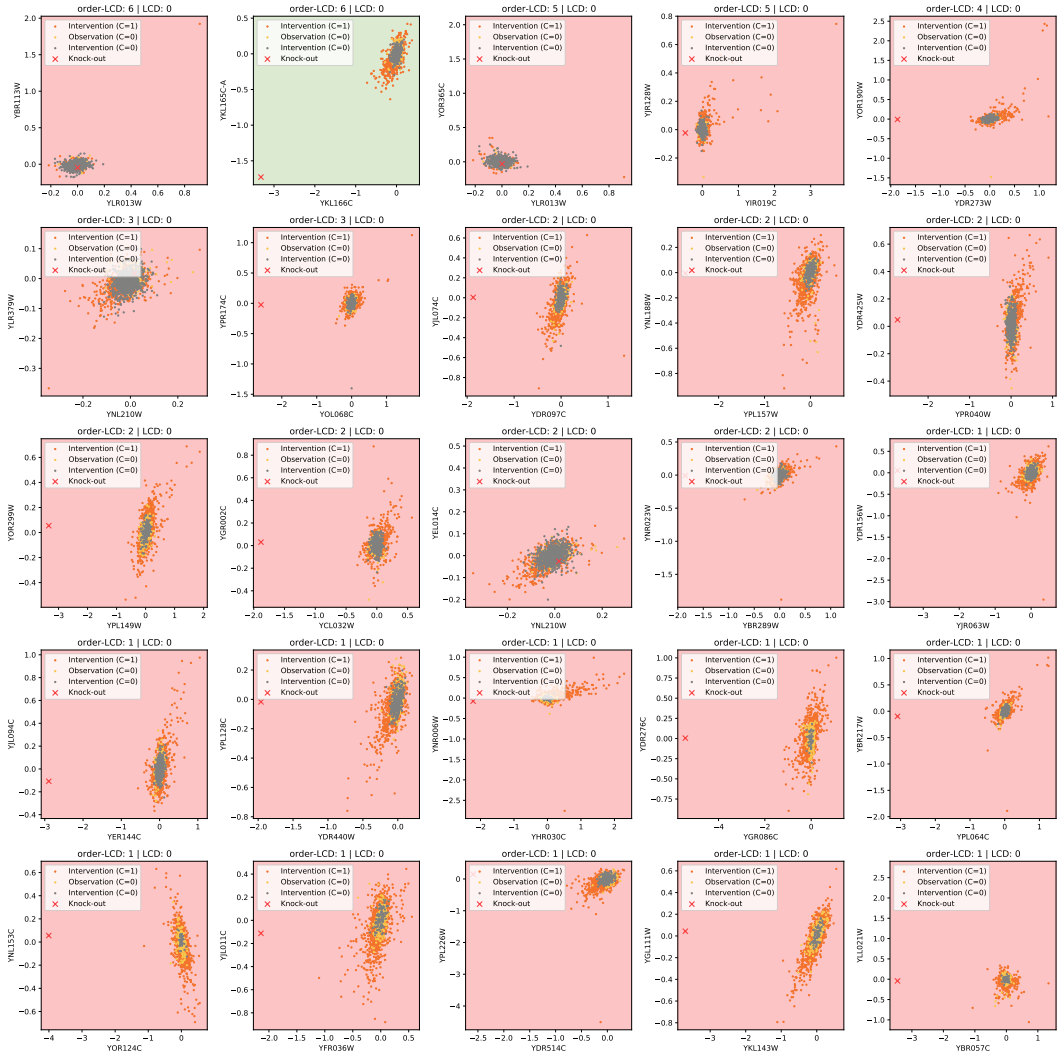

Figure 37: The 25 strongest predictions by **order-based LCD** that are **excluded** in the LCD predictions. The precise scores are shown above the figures. The background color indicates whether the relation is true according to the 10% threshold.

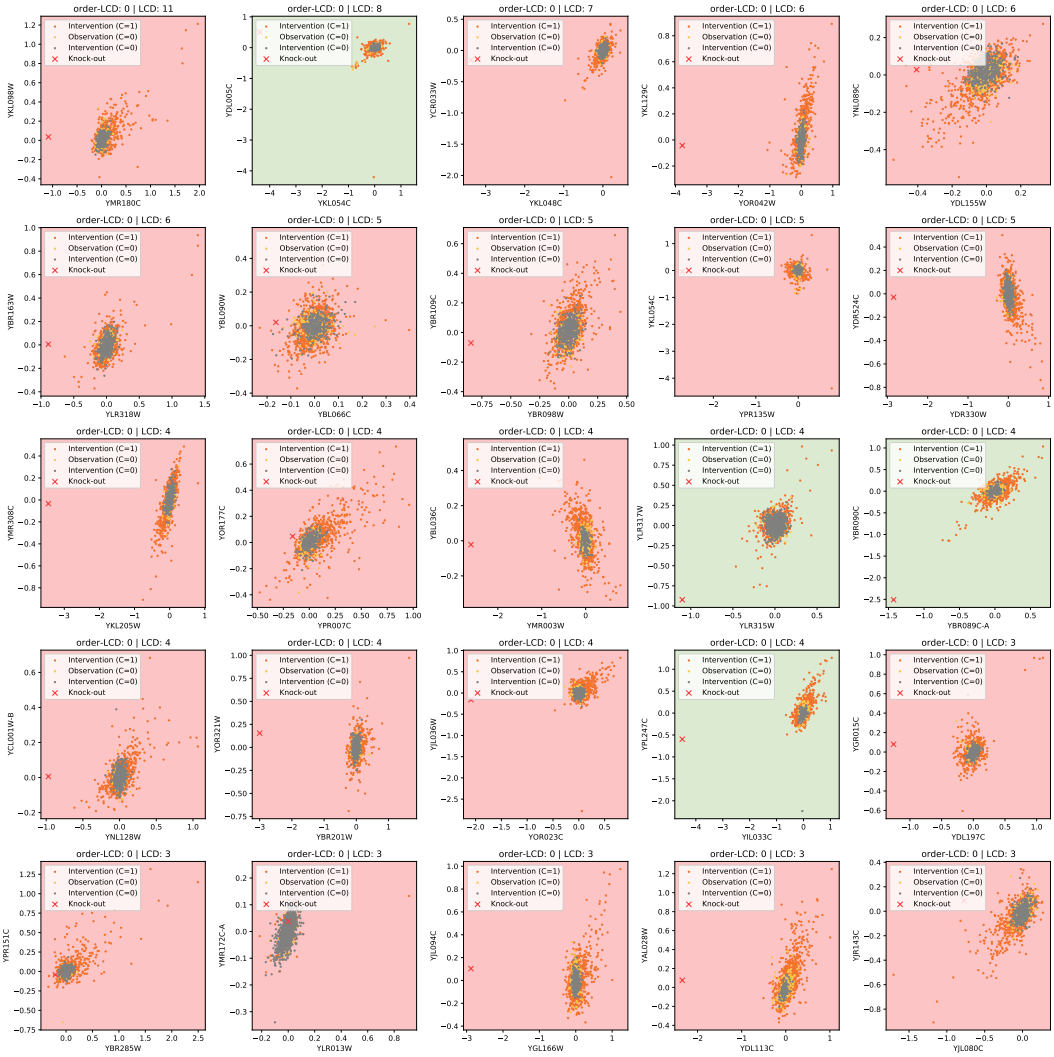

Figure 38: The 25 strongest predictions by **LCD** that are also **excluded** in the order-based LCD predictions. The precise scores are shown above the figures. The background color indicates whether the relation is true according to the 10% threshold.

Expression Levels per Context Group

Figure 39: Distribution of the expression level of the cause genes in the strongest discrete predictions of **order-LCD**. Clusters are made based on whether the data points are observation or intervention data, and based on the context value if the data point is interventional.

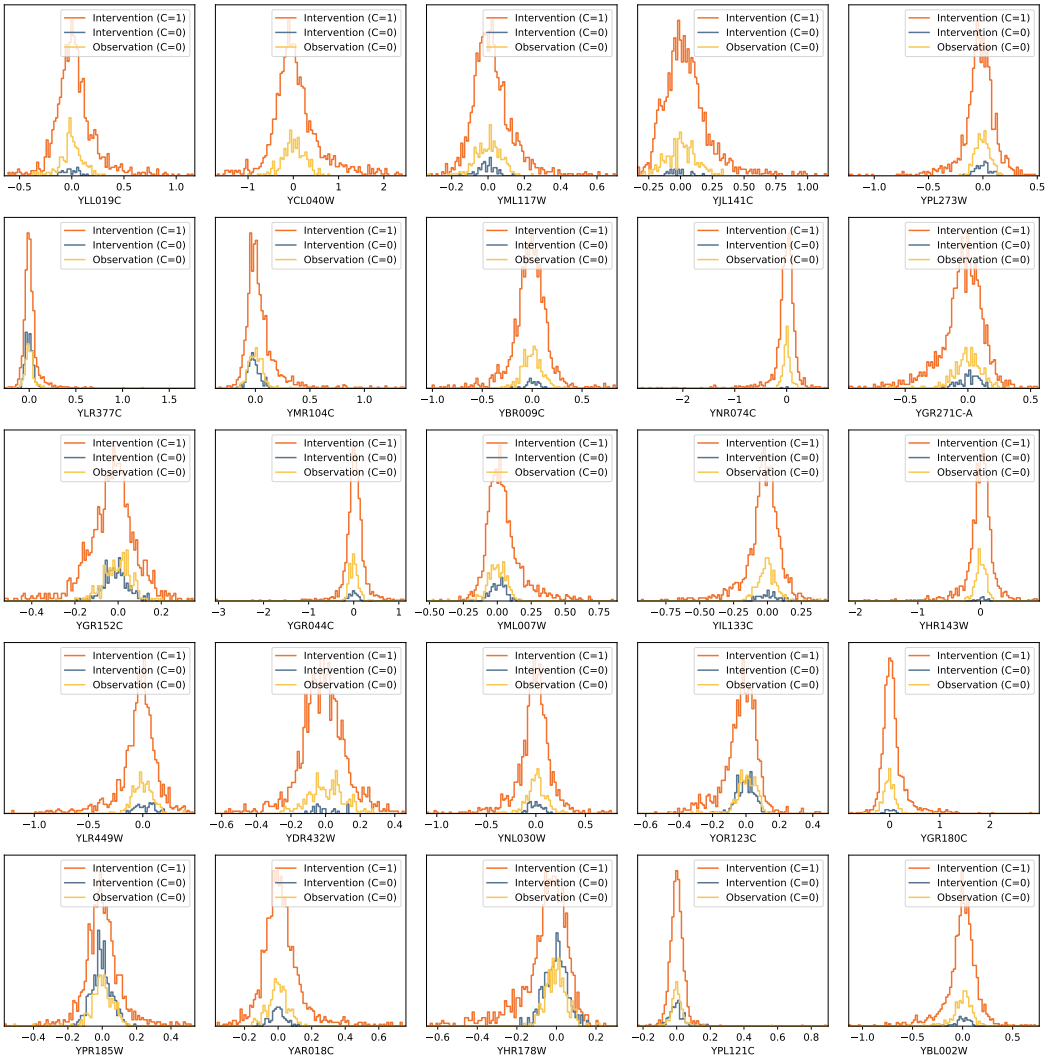

Figure 40: Distribution of the expression level of the cause genes in the strongest discrete predictions of the original **LCD**. Clusters are made based on whether the data points are observation or intervention data, and based on the context value if the data point is interventional.

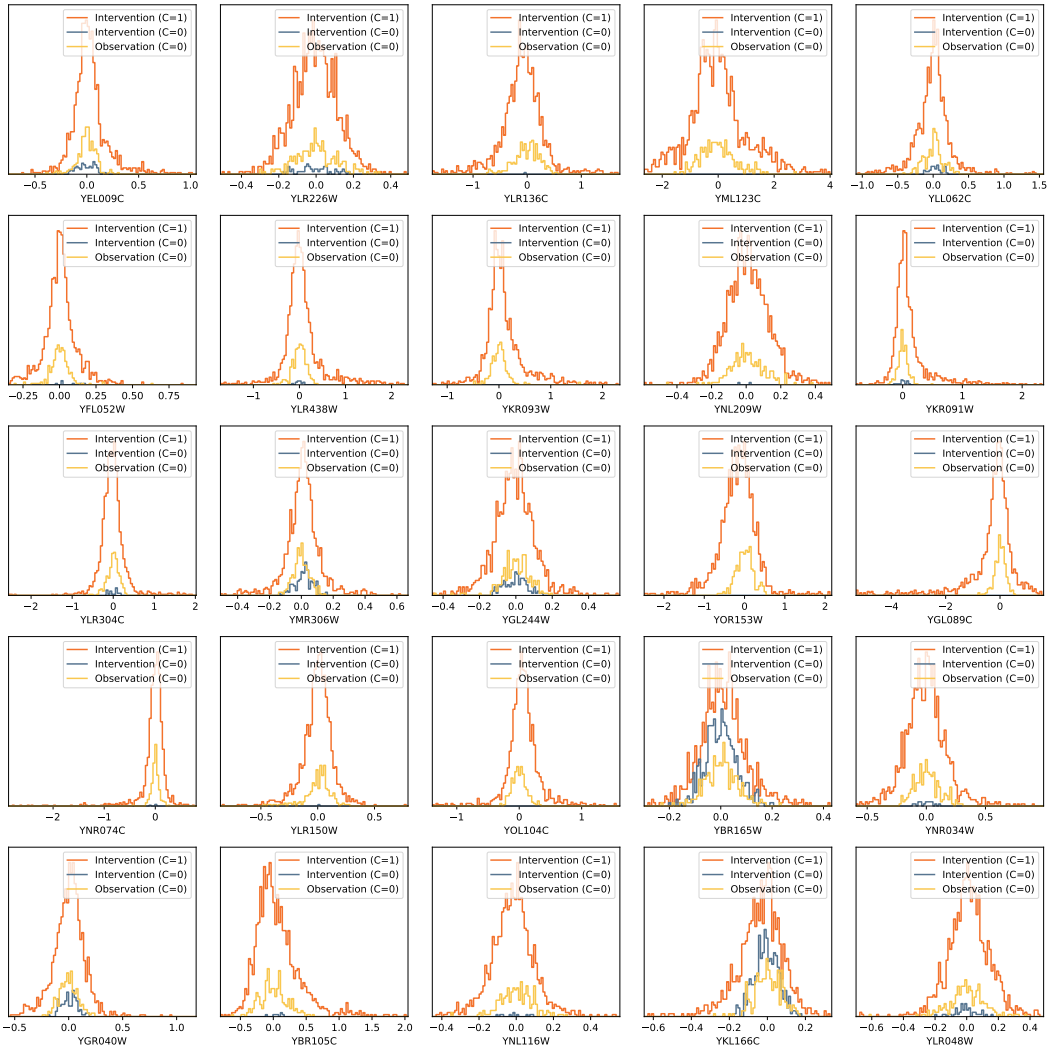

Figure 41: Distribution of the expression level of the cause genes in the strongest discrete predictions of the **L_2-boosting baseline**. Clusters are made based on whether the data points are observation or intervention data, and based on the context value if the data point is interventional.

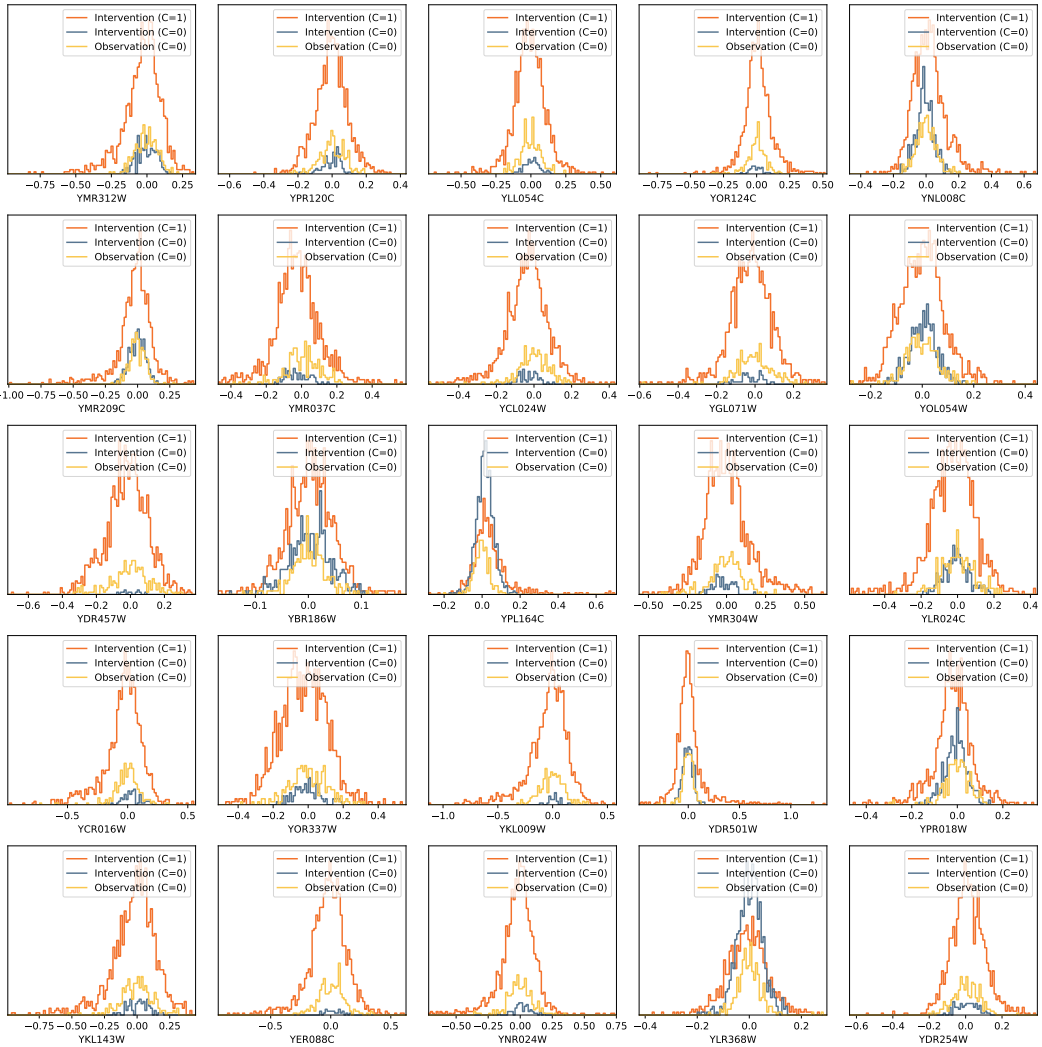

Figure 42: Distribution of the expression level of 25 **random** cause genes. Clusters are made based on whether the data points are observation or intervention data, and based on the context value if the data point is interventional.

References

D. M. Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554, 2002.

T. Claassen, J. Mooij, and T. Heskes. Learning sparse causal models is not NP-hard. *arXiv preprint arXiv:1309.6824*, 2013.

G. F. Cooper. A simple constraint-based algorithm for efficiently mining observational databases for causal relationships. *Data Mining and Knowledge Discovery*, 1(2):203–224, 1997.

J. Edmonds. Optimum branchings. *Journal of Research of the national Bureau of Standards B*, 71(4):233–240, 1967.

A. E. Eiben, J. E. Smith, et al. *Introduction to evolutionary computing*. Springer, 2003.

R. A. Fisher. The distribution of the partial correlation coefficient. *Metron*, 3: 329–332, 1924.

P. Forré and J. M. Mooij. Markov properties for graphical models with cycles and latent variables. *arXiv preprint arXiv:1710.08775*, 2017.

K. Grace, J. Salvatier, A. Dafoe, B. Zhang, and O. Evans. When will ai exceed human performance? evidence from ai experts. *Journal of Artificial Intelligence Research*, 62:729–754, 2018.

M. Gupte, P. Shankar, J. Li, S. Muthukrishnan, and L. Iftode. Finding hierarchy in directed online social networks. In *Proceedings of the 20th international conference on World wide web*, pages 557–566, 2011.

V. Guruswami, R. Manokaran, and P. Raghavendra. Beating the random ordering is hard: Inapproximability of maximum acyclic subgraph. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 573–582. IEEE, 2008.

A. Hauser and P. Bühlmann. Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research*, 13(Aug):2409–2464, 2012.

R. Herbrich, T. Minka, and T. Graepel. Trueskill™: a bayesian skill rating system. In *Advances in neural information processing systems*, pages 569–576, 2007.

A. Hyttinen, F. Eberhardt, and M. Järvisalo. Constraint-based Causal Discovery: Conflict Resolution with Answer Set Programming. In *UAI*, pages 340–349, 2014.

A. B. Kahn. Topological sorting of large networks. *Communications of the ACM*, 5(11):558–562, 1962.

P. Kemmeren, K. Sameith, L. A. L. van de Pasch, J. J. Benschop, T. L. Lenstra, T. Margaritis, E. O'Duibhir, E. Apweiler, S. van Wageningen, C. W. Ko, and Others. Large-scale genetic perturbations reveal regulatory networks and an abundance of gene-specific repressors. *Cell*, 157(3):740–752, 2014.

F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on information theory*, 47(2): 498–519, 2001.

T. L. Lenstra, J. J. Benschop, T. Kim, J. M. Schulze, N. A. C. H. Brabers, T. Margaritis, L. A. L. van de Pasch, S. A. A. C. van Heesch, M. O. Brok, M. J. A. G. Koerkamp, and Others. The specificity and topology of chromatin interaction pathways in yeast. *Molecular cell*, 42(4):536–549, 2011.

S. Magliacane, T. Claassen, and J. M. Mooij. Ancestral causal inference. In *Advances in Neural Information Processing Systems*, pages 4466–4474, 2016.

S. Mani. *A bayesian local causal discovery framework*. PhD thesis, University of Pittsburgh, 2006.

C. Meek. *Graphical Models: Selecting causal and statistical models*. PhD thesis, PhD thesis, Carnegie Mellon University, 1997.

N. Meinshausen and P. Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

N. Meinshausen, A. Hauser, J. M. Mooij, J. Peters, P. Versteeg, and P. Bühlmann. Methods for causal inference from gene perturbation experiments and validation. *Proceedings of the National Academy of Sciences*, 113 (27):7361–7368, 2016.

T. P. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

J. M. Mooij, J. Cremers, and Others. An empirical study of one of the simplest causal prediction algorithms. In *UAI 2015 Workshop on Advances in Causal Inference*, number 1504, pages 30–39, 2015.

J. M. Mooij, S. Magliacane, and T. Claassen. Joint causal inference from multiple contexts. *arXiv preprint arXiv:1611.10351*, 2016.

I. Oliver, D. Smith, and J. R. Holland. Study of permutation crossover operators on the traveling salesman problem. In *Genetic algorithms and their applications: proceedings of the second International Conference on Genetic Algorithms: July 28-31, 1987 at the Massachusetts Institute of Technology, Cambridge, MA*. Hillsdale, NJ: L. Erlhaum Associates, 1987., 1987.

L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

J. Pearl. *Causality*. Cambridge university press, 2009.

J. Pearl. The seven tools of causal inference, with reflections on machine learning. *Communications of the ACM*, 62(3):54–60, 2019.

J. Peters, P. Bühlmann, and N. Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016.

J. Peters, D. Janzing, and B. Schölkopf. *Elements of causal inference: foundations and learning algorithms*. MIT press, 2017.

G. Raskutti and C. Uhler. Learning directed acyclic graph models based on sparsest permutations. *Stat*, 7(1):e183, 2018.

H. Reichenbach. *The direction of time*. University of California Press, Berkeley, 1956.

R. E. Schapire, Y. Freund, P. Bartlett, W. S. Lee, et al. Boosting the margin: A new explanation for the effectiveness of voting methods. *The annals of statistics*, 26(5):1651–1686, 1998.

L. Solus, Y. Wang, L. Matejovicova, and C. Uhler. Consistency guarantees for permutation-based causal inference algorithms. *arXiv preprint arXiv:1702.03530*, 2017.

P. Spirtes and C. Glymour. An algorithm for fast recovery of sparse causal graphs. *Social science computer review*, 9(1):62–72, 1991.

P. Spirtes, C. Meek, and T. Richardson. *An algorithm for causal inference in the presence of latent variables and selection bias*, volume 1. MIT Press, 1999.

P. Spirtes, C. N. Glymour, R. Scheines, D. Heckerman, C. Meek, G. Cooper, and T. Richardson. *Causation, prediction, and search*. MIT press, 2000.

J. Sun, D. Ajwani, P. K. Nicholson, A. Sala, and S. Parthasarathy. Breaking cycles in noisy hierarchies. In *Proceedings of the 2017 ACM on Web Science Conference*, pages 151–160, 2017.

N. Tatti. Hierarchies in directed networks. In *2015 IEEE international conference on data mining*, pages 991–996. IEEE, 2015.

S. Triantafillou, V. Lagani, C. Heinze-Deml, A. Schmidt, J. Tegner, and I. Tsamardinos. Predicting causal relationships from biological data: Applying automated causal discovery on mass cytometry data of human immune cells. *Scientific reports*, 7(1):1–11, 2017.

N. Tyagi and S. Sharma. Weighted page rank algorithm based on number of visits of links of web page. *International Journal of Soft Computing and Engineering (IJSCE) ISSN*, pages 2231–2307, 2012.

C. Uhler, G. Raskutti, P. Bühlmann, and B. Yu. Geometry of the faithfulness assumption in causal inference. *The Annals of Statistics*, pages 436–463, 2013.

O. D. Van der Wal and C. K. L. Man. On the motivational benefits of friendship and stand-ups: A case study. *Vansil Journal of Psychology*, 62(3):54–60, 2020.

F. Van Harmelen and A. t. Teije. A boxology of design patterns for hybrid learning and reasoning systems. *arXiv preprint arXiv:1905.12389*, 2019.

S. Van Wageningen, P. Kemmeren, P. Lijnzaad, T. Margaritis, J. J. Benschop, I. J. de Castro, D. Van Leenen, M. J. A. G. Koerkamp, C. W. Ko, A. J. Miles, and Others. Functional overlap and regulatory links shape genetic interactions between signaling pathways. *Cell*, 143(6):991–1004, 2010.

T. Verma and J. Pearl. *Equivalence and synthesis of causal models*. UCLA, Computer Science Department, 1991.

P. Versteeg and J. M. Mooij. Boosting Local Causal Discovery in High-Dimensional Expression Data. *arXiv preprint arXiv:1910.02505*, 2019.

Y. Wang, L. Solus, K. Yang, and C. Uhler. Permutation-based causal inference algorithms with interventions. In *Advances in Neural Information Processing Systems*, pages 5822–5831, 2017.