DCI: Learning Causal Differences between Gene Regulatory Networks

Anastasiya Belyaeva[1], Chandler Squires[1] and Caroline Uhler[1, 2,*]

[1]Laboratory for Information & Decision Systems and Institute for Data, Systems and Society, Massachusetts Institute of Technology, Cambridge, 02139, USA
[2]Department of Biosystems Science and Engineering, ETH Zurich, Basel, 4058, Switzerland

[*]To whom correspondence should be addressed.

Abstract

Summary: Designing interventions to control gene regulation necessitates modeling a gene regulatory network by a causal graph. Currently, large-scale expression datasets from different conditions, cell types, disease states and developmental time points are being collected. However, application of classical causal inference algorithms to infer gene regulatory networks based on such data is still challenging, requiring high sample sizes and computational resources. Here, we propose an algorithm that efficiently learns the differences in gene regulatory mechanisms between different conditions. Our difference causal inference (DCI) algorithm infers changes (i.e., edges that appeared, disappeared or changed weight) between two causal graphs given gene expression data from the two conditions. This algorithm is efficient in its use of samples and computation since it infers the differences between causal graphs directly without estimating each possibly large causal graph separately. We provide a user-friendly Python implementation of DCI and also enable the user to learn the most robust difference causal graph across different tuning parameters via stability selection. Finally, we show how to apply DCI to bulk and single-cell RNA-seq data from different conditions and cell states, and we also validate our algorithm by predicting the effects of interventions.

Availability and implementation: All algorithms are freely available as a Python package at http://uhlerlab.github.io/causaldag/dci

Contact: cuhler@mit.edu

1 Introduction

Biological processes from differentiation to disease progression are governed by gene regulatory networks. Over the past few decades, various methods have been developed for inferring gene regulatory networks from gene expression data (Wang and Huang, 2014). The majority of methods learn undirected graphs of interactions between genes such as correlation-based coexpression networks (Langfelder and Horvath, 2008), Gaussian graphical models that capture partial correlations (Friedman *et al.*, 2008), or networks that measure dependencies between genes using mutual information (Reshef *et al.*, 2011) . However, the ultimate goal is often to use gene regulatory networks to predict the effect of an intervention (small molecule, overexpression of a transcription factor, knock-out of a gene, etc.). This cannot be done using an undirected graph and necessitates modeling a gene regulatory network by a causal (directed) graph.

One of the most common frameworks for representing causal relationships are directed acyclic graphs (DAGs). A variety of methods including the prominent PC and GES algorithms have been developed for learning causal graphs from observational data (Glymour *et al.*, 2019). These methods have been successfully applied to learning (directed) gene regulatory networks on a small number of genes, starting with a pioneering study by Friedman *et al.*, 2000. However, applying these methods at the whole genome-level is still challenging due to high sample size and computational requirements of the algorithms.

We address this problem by noting that it is often of interest to learn *changes* in causal (regulatory) relationships between two related gene regulatory networks corresponding to different conditions, disease states, cell types or developmental time points, as opposed to learning the full gene regulatory network for each condition. This can reduce the high sample and computational requirements of current causal inference algorithms, since while the full regulatory network is often large and dense, the difference between two related regulatory networks is often small and sparse. As of now, this problem has only been addressed in the undirected setting, namely by KLIEP (Liu *et al.*, 2017), DPM (Zhao *et al.*, 2014) and others (Fukushima, 2013; Lichtblau *et al.*, 2017) that estimate differences between undirected graphs; for a recent review see Shojaie (2020).

With the recent advances in high-throughput single-cell RNA-sequencing under different contexts and cell types (Zheng *et al.*, 2017), there is a growing need for methods that learn changes between gene regulatory networks. Causal inference algorithms that infer changes between gene regulatory networks could for example reveal that a particular gene controls different sets of target genes in different conditions. While for small number of genes, it is feasible to apply current causal structure discovery algorithms to learn the causal graphs for each condition seperately and take the difference of the graphs, this approach is highly inefficient in its use of samples and computation. In this paper, we describe the *difference causal inference* (*DCI*) algorithm for direct estimation of the difference causal graph based on observational data from two conditions. For theoretical properties of this algorithm, in particular the proof that DCI provides consistent estimates of the difference causal graph, and simulations showing that it outperforms the naive approach of separate estimation of each causal graph and subsequent computation of the difference, see Wang *et al.* (2018). In this paper, we present an easy to use Python package to apply DCI to gene expression data from different conditions and demonstrate the algorithm's performance on predicting the effects of interventions on single-cell RNA-seq data. Importantly, our DCI Python implementation also allows selecting the most robust difference gene regulatory network based on a collection of tuning parameters via stability selection (Meinshausen and Bühlmann, 2010). We also include a tutorial and an example use case of DCI on a bulk RNA-seq dataset from two ovarian cancer patient cohorts with different survival rates from Tothill *et al.* (2008) at http://uhlerlab.github.io/causaldag/dci_tutorial To seamlessly integrate DCI with other causal inference methods, we incorporated our DCI code into the larger `causaldag` package.

2 Difference Causal Inference (DCI) package

DCI takes as input two gene expression matrices $\hat{X}^{(1)}$ and $\hat{X}^{(2)}$ of size $n_1 \times p$ and $n_2 \times p$, where n_1 and n_2 denote the number of samples in each dataset and p denotes the number of genes. These matrices contain the RNA-seq values corresponding to two different conditions, such as healthy and diseased, different cell types, or different time points. DCI outputs the difference causal graph between the two conditions, i.e. the edges in the gene regulatory networks that appeared, disappeared or changed weight between the two conditions (Fig. 1).

The data for each condition is assumed to be generated by a linear structural equation model with Gaussian noise. More precisely, let $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ denote two DAGs on p nodes with weighted adjacency matrices $B^{(1)}$ and $B^{(2)}$. Each node $j \in \{1, \ldots, p\}$ in the two graphs $\mathcal{G}^{(k)}$, $k \in \{1, 2\}$, is associated with a random variable $X_j^{(k)}$, which is given by a weighted sum of its parents and independent Gaussian noise $\epsilon^{(k)}$, i.e.,

$$X_j^{(k)} = \sum_{i=1}^{p} B_{ij}^{(k)} X_i + \epsilon_j^{(k)}.$$

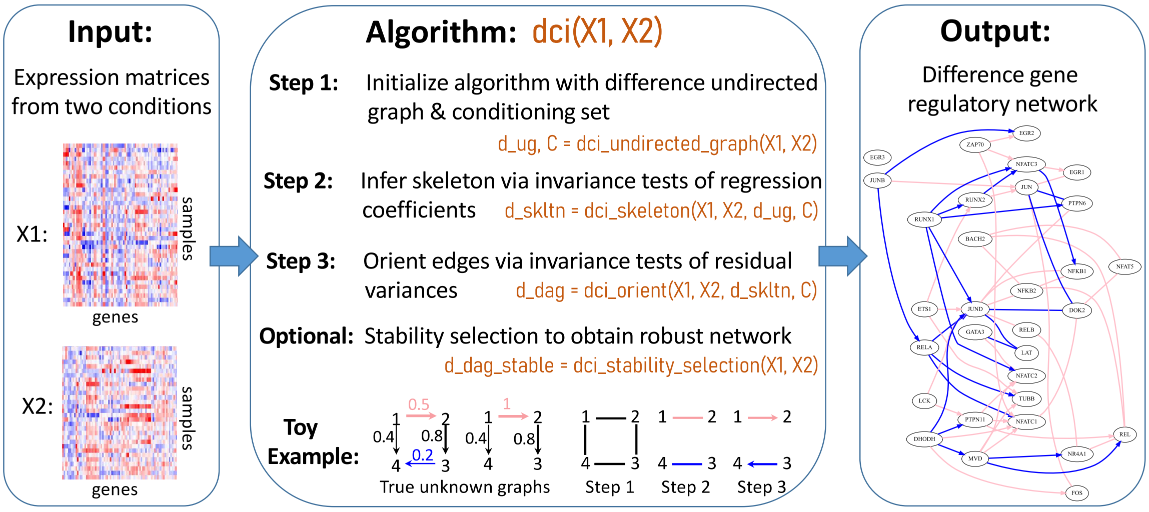

Input:
Expression matrices from two conditions
X1:
samples
genes
X2:
samples
genes
Algorithm: dci(X1, X2)
Step 1: Initialize algorithm with difference undirected graph & conditioning set
d_ug, C = dci_undirected_graph(X1, X2)
Step 2: Infer skeleton via invariance tests of regression coefficients
d_skltn = dci_skeleton(X1, X2, d_ug, C)
Step 3: Orient edges via invariance tests of residual variances
d_dag = dci_orient(X1, X2, d_skltn, C)
Optional: Stability selection to obtain robust network
d_dag_stable = dci_stability_selection(X1, X2)
Toy Example:
0.5
1
2
0.4
0.8
4
0.2
3
1
1
2
0.4
0.8
4
3
True unknown graphs
1 2 4 3
Step 1
1 2 4 3
Step 2
1 2 4 3
Step 3
Output:
Difference gene regulatory network
EGR2
ZAP70
EGR3
JUNB
NFATC3
EGR1
JUN
RUNX2
PTPN6
RUNX1
BACH2
NFAT5
NFKB1
NFKB2
ETS1
JUND
DOK2
GATA3
RELB
RELA
LAT
NFATC2
TUBB
LCK
PTPN11
NFATC1
DHODH
REL
NR4A1
MVD
FOS

Figure 1: Overview of DCI algorithm: DCI takes as input two gene expression matrices $X1$ and $X2$, representing two different conditions of interest. The function `dci(X1,X2)` outputs the difference gene regulatory network consisting of the causal relationships that appeared, disappeared or changed weight between the two conditions.

Given data $\hat{X}^{(1)}$ and $\hat{X}^{(2)}$ from two unknown causal graphs $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$, DCI determines their difference, i.e., edges $i \to j$ for which $B_{ij}^{(1)} \neq B_{ij}^{(2)}$. DCI consists of three steps described below (for further details see Supplementary Materials) and in Fig. 1. These steps are implemented in the `dci` function of the `causaldag` package.

Step 1: Initialization with a difference undirected graph. To start with a reduced set of nodes and edges, DCI is initialized with a difference undirected graph, which represents changes of conditional dependencies among genes between the two conditions, as well as with a node set $\mathcal{C}$, a superset of the nodes in the difference undirected graph consisting of nodes to be considered as conditioning sets in the downstream hypothesis tests. The undirected difference graph and the node set $\mathcal{C}$ can be estimated using previous methods such as KLIEP (Liu *et al.*, 2017), which we implemented in the function `dci_undirected_graph`. Alternatively, DCI can be initialized based on a user's prior biological knowledge or, if the number of genes to be considered is small, with the complete graph.

Step 2: Estimation of the skeleton of the difference causal graph. The function `dci_skeleton` removes edges from the difference undirected graph to obtain the skeleton of the difference causal graph by testing for invariance of regression coefficients. Note that each entry $B_{ij}^{(k)}$ corresponds to a particular regression coefficient $\beta_{ij|S}$, namely obtained when regressing $X_j^{(k)}$ on $X_i^{(k)}$ given the parents of node j in $\mathcal{G}^{(k)}$. Thus, testing whether $B_{ij}^{(1)} = B_{ij}^{(2)}$, is equivalent to testing whether there exists a set of nodes $S \subset \mathcal{C}$ such that $\beta_{i,j|S}^{(1)} = \beta_{i,j|S}^{(2)}$. These regression coefficients are estimated from the data $\hat{X}^{(1)}$ and $\hat{X}^{(2)}$ and invariance of the regression coefficients across $k \in \{1, 2\}$ is tested via an F-test.

Step 3: Orienting edges in the difference causal graph. While not all edge directions in the difference causal graph are identifiable from observational data, the function `dci_orient` orients all identifiable edges by testing for invariance of residual variances giving rise to a partially oriented difference gene regulatory network. For any edge $i - j$ in the undirected graph obtained in Step 2, if there exists a set of nodes $S \subseteq \mathcal{C} \setminus \{j\}$ such that the residual variances satisfy $\sigma_{j|S}^{(1)} = \sigma_{j|S}^{(2)}$, then the edge is directed as $i \to j$ if $i \in S$ and $j \to i$ otherwise (see Supplementary Materials). The residual variances are again estimated from the data $\hat{X}^{(1)}$ and $\hat{X}^{(2)}$ and their invariance across $k \in \{1, 2\}$ is tested via an F-test.

Stability selection to obtain robust difference gene regulatory network. DCI requires choosing hyperparameters for each step, namely the ℓ_1 regularization parameter for KLIEP in step (1) and the significance levels for the hypothesis tests of invariance of regression coefficients in step (2) and residual variances in step (3). To overcome the difficulty of selecting hyperparameters for model selection, Meinshausen and Bühlmann (2010) proposed stability selection, which achieves family-wise error rate control and has been successfully applied for learning causal graphs in genomics (Meinshausen *et al.*, 2016). The function `dci_stability_selection` implements DCI with stability selection by running the DCI algorithm across a grid of tuning parameter combinations and bootstrap samples of the datasets $\hat{X}^{(1)}$ and $\hat{X}^{(2)}$. The results are aggregated and only edges with a stability score above a predefined threshold are output in the difference causal graph.

3 Applications and Conclusions

We applied DCI to two single-cell gene expression datasets, namely CROP-seq (Datlinger *et al.*, 2017) and Perturb-seq (Dixit *et al.*, 2016). Both datasets also contain interventional gene expression data from knockouts, thereby allowing us to assess the performance of DCI. In particular, we applied DCI to the observational single-cell data and evaluated it using an ROC curve based on the interventional data (see Supplementary Materials). By applying DCI to the CROP-seq and Perturb-seq data respectively, we learned the difference gene regulatory network between naive and activated T-cells (SI Fig. S1-S3) as well as between pre- and post-stimulation of dendritic cells with LPS (SI Fig. S4-S6). In both cases DCI outperforms the naive approach of estimating two causal graphs separately and taking their difference, and can provide valuable mechanistic insights into the biological processes of interest.

We developed the DCI package for learning differences between gene regulatory networks based on gene expression data from two different conditions of interest, such as healthy and diseased, different cell types or developmental time points. Our package is implemented in Python for ease-of-use and also includes functionality to ensure that the output difference gene regulatory network is stable and robust across different hyperparameters and data subsampling.

Funding

Anastasiya Belyaeva was supported by an NSF Graduate Research Fellowship (1122374) and the Abdul Latif Jameel World Water and Food Security Lab (J-WAFS) at MIT as well as the MIT J-Clinic for Machine Learning and Health. Chandler Squires was supported by an NSF Graduate Research Fellowship, an MIT Presidential Fellowship and an MIT UCEM Sloan scholarship. Caroline Uhler was partially supported by NSF (DMS-1651995), ONR (N00014-17-1-2147 and N00014-18-1-2765), IBM and a Simons Investigator Award.

References

Datlinger, P., Rendeiro, A. F., Schmidl, C., Krausgruber, T., Traxler, P., Klughammer, J., Schuster, L. C., Kuchler, A., Alpar, D., and Bock, C. (2017). Pooled CRISPR screening with single-cell transcriptome readout. *Nature Methods*, **14**(3), 297.

Dixit, A., Parnas, O., Li, B., Chen, J., Fulco, C. P., Jerby-Arnon, L., Marjanovic, N. D., Dionne, D., Burks, T., Raychowdhury, R., Adamson, B., Norman, T. M., Lander, E. S., Weissman, J. S., Friedman, N., and Regev, A. (2016). Perturb-seq: dissecting molecular circuits with scalable single-cell RNA profiling of pooled genetic screens. *Cell*, **167**(7), 1853–1866.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, **9**(3), 432–441.

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, **7**(3-4), 601–620.

Fukushima, A. (2013). DiffCorr: an R package to analyze and visualize differential correlations in biological networks. *Gene*, **518**(1), 209–214.

Glymour, C., Zhang, K., and Spirtes, P. (2019). Review of causal discovery methods based on graphical models. *Frontiers in Genetics*, **10**, 524.

Langfelder, P. and Horvath, S. (2008). WGCNA: an R package for weighted correlation network analysis. *BMC Bioinformatics*, **9**(1), 559.

Lichtblau, Y., Zimmermann, K., Haldemann, B., Lenze, D., Hummel, M., and Leser, U. (2017). Comparative assessment of differential network analysis methods. *Briefings in Bioinformatics*, **18**(5), 837–850.

Liu, S., Fukumizu, K., and Suzuki, T. (2017). Learning sparse structural changes in high-dimensional Markov networks. *Behaviormetrika*, **44**(1), 265–286.

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

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

Reshef, D. N., Reshef, Y. A., Finucane, H. K., Grossman, S. R., McVean, G., Turnbaugh, P. J., Lander, E. S., Mitzenmacher, M., and Sabeti, P. C. (2011). Detecting novel associations in large data sets. *Science*, **334**(6062), 1518–1524.

Shojaie, A. (2020). Differential network analysis: A statistical perspective. *Wiley Interdisciplinary Reviews: Computational Statistics*, page e1508.

Tothill, R. W. *et al.* (2008). Novel molecular subtypes of serous and endometrioid ovarian cancer linked to clinical outcome. *Clinical Cancer Research*, **14**(16), 5198–5208.

Wang, Y., Squires, C., Belyaeva, A., and Uhler, C. (2018). Direct estimation of differences in causal graphs. In *Advances in Neural Information Processing Systems*, pages 3770–3781.

Wang, Y. R. and Huang, H. (2014). Review on statistical methods for gene network reconstruction using expression data. *Journal of Theoretical Biology*, **362**, 53–61.

Zhao, S. D., Cai, T. T., and Li, H. (2014). Direct estimation of differential networks. *Biometrika*, **101**(2), 253–268.

Zheng, G. X. Y. *et al.* (2017). Massively parallel digital transcriptional profiling of single cells. *Nature Communications*, **8**, 14049.

Supplementary Materials for DCI: Learning Causal Differences between Gene Regulatory Networks

Anastasiya Belyaeva[1], Chandler Squires[1] and Caroline Uhler[1, 2,*]

[1]Laboratory for Information & Decision Systems and Institute for Data, Systems and Society, Massachusetts Institute of Technology, Cambridge, 02139, USA
[2]Department of Biosystems Science and Engineering, ETH Zurich, Basel, 4058, Switzerland

[*]To whom correspondence should be addressed.

This PDF file includes:

Supplementary Note
Figures S1 to S6
References

Supplementary Note

Difference Causal Inference (DCI) algorithm

In the following, we provide more details regarding the DCI algorithm; for a theoretical analysis of the algorithm see also [1]. Let $\mathcal{G}^{(k)} = ([p], E^{(k)})$ for $k \in \{1, 2\}$ be a directed acyclic graph (DAG) with nodes $[p] := \{1, \dots, p\}$ and directed edges $E^{(k)}$. The DAGs $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ model the gene regulatory networks in the two conditions of interest. We assume that the two DAGs are consistent with the same ordering, meaning that there cannot be an edge $i \to j$ in $\mathcal{G}^{(1)}$ if there is a directed path $j \to \cdots \to i$ in $\mathcal{G}^{(2)}$ and vice-versa. This assumption is reasonable in gene regulatory networks, since genetic interactions may appear or disappear or change edge weights, but generally do not change directions. For each graph we associate a random variable $X_i^{(k)}$ to each node $i \in [p]$. Recall that we consider the setting where we have data from two conditions and this data is generated by a linear structural equation model

$$X^{(k)} = B^{(k)T} X^{(k)} + \epsilon^{(k)} \qquad \text{for} \;\; k \in \{1, 2\}, \tag{1}$$

where $X = (X_1, \cdots, X_p)^T$ is a random vector, $B^{(k)}$ denotes the weighted adjacency matrix of the DAG $\mathcal{G}^{(k)}$ and $\epsilon^{(k)} \sim \mathcal{N}(0, \Omega^{(k)})$ denotes Gaussian noise with covariance matrix $\Omega^{(k)} := \mathrm{diag}(\sigma_1^{2(k)}, \cdots, \sigma_p^{2(k)})$. Given samples $\hat{X}^{(1)} \in \mathbb{R}^{n_1 \times p}$ and $\hat{X}^{(2)} \in \mathbb{R}^{n_2 \times p}$ from the two models (where n_1 and n_2 denote the sample size under each condition), our goal is to estimate the difference-DAG across the two conditions. The difference-DAG is denoted by $\Delta = ([p], E)$ and contains an edge $i \to j \in E$ if and only if $B_{ij}^{(1)} \neq B_{ij}^{(2)}$.

Algorithm 1 describes the three steps of our DCI algorithm for computing the difference-DAG. In the first step, the algorithm is initialized with a difference undirected graph, which we denote by $\bar{\Delta}$, with edge $i - j$ if and only if $\Theta_{ij}^{(1)} \neq \Theta_{ij}^{(2)}$ for $i \neq j$, where $\Theta^{(1)}$ and $\Theta^{(2)}$ are the precision matrices corresponding to the DAGs $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$. This is done to remove some edges to reduce the downstream computational burden. The difference undirected graph can be determined either using previous methods such as KLIEP [2–6], based on prior biological knowledge, or simply with the complete graph when the number of considered genes is small. In addition, to reduce the number of downstream hypothesis tests, the nodes to be considered as conditioning sets can be reduced to the nodes in the difference undirected graph as well as nodes whose conditional distribution changes between the two conditions, namely $\mathcal{C} = \left\{ i \mid \exists j \in [p] \text{ such that } \Theta_{i,j}^{(1)} \neq \Theta_{i,j}^{(2)} \right\}$. The reduced node set can be determined from the output of methods such as KLIEP [2–6], prior biological knowledge, or taken as the set of all nodes when the number of genes to be considered is small.

In the second step, the skeleton of the difference-DAG, denoted by $\tilde{\Delta}$, is estimated via Algorithm 2. This is done by calculating regression coefficients $\beta_{i,j|S}^{(k)}$ and testing whether they are invariant, i.e. whether $\beta_{i,j|S}^{(1)} = \beta_{i,j|S}^{(2)}$, using an F-test. Given $i, j \in [p]$ and $S \subseteq [p] \setminus \{i, j\}$, the regression coefficient $\beta_{i,j|S}^{(k)}$ is defined as the entry in $\beta_M^{(k)}$ corresponding to i, where $\beta_M^{(k)}$ is the best linear predictor of $X_j^{(k)}$ given $X_M^{(k)}$, i.e., the minimizer of $\mathbb{E}[(X_j^{(k)} - (\beta_M^{(k)})^T X_M^{(k)})^2]$ and $M := \{i\} \cup S$. Hence, $\beta_{i,j|S}^{(k)}$ can be computed in closed form. Note that $B_{ij}^{(k)}$ corresponds to a particular regression coefficient, namely when $S = \mathrm{Pa}^{(k)}(j) \setminus \{i\}$, where $\mathrm{Pa}^{(k)}(j)$ denotes the parents of node j in $\mathcal{G}^{(k)}$. This means that we can determine whether $B_{ij}^{(1)} = B_{ij}^{(2)}$ without learning each graph $\mathcal{G}^{(k)}$, namely by testing subsets S: if there exists a subset S such that $\beta_{i,j|S}^{(1)} = \beta_{i,j|S}^{(2)}$, then $B_{ij}^{(1)} = B_{ij}^{(2)}$ and hence the edge $(i, j) \notin \tilde{\Delta}$. In fact, it turns out that it is sufficient to consider conditioning sets $S \subseteq \mathcal{C}$ [1].

Finally, in the third step we direct edges in the skeleton of the difference-DAG $\tilde{\Delta}$ using Algorithm 3. Similar to many prominent causal inference algorithms such as the PC [7] and GES [8] algorithms, we may not be able to determine the directions of all edges in $\tilde{\Delta}$, since in general, the difference-DAG Δ is not completely identifiable. In fact, we are able to identify the direction of all edges adjacent to nodes whose internal node variances are unchanged across the two conditions, i.e. for which $\sigma_i^{(1)} = \sigma_i^{(2)}$ [1]. Hence the output of the DCI algorithm is a partially directed acyclic graph, which contains both directed and undirected edges. Edge directions in the difference-DAG are determined by calculating residual variances $(\sigma_{j|S}^{(k)})^2$ and testing whether they are invariant, i.e. whether $(\sigma_{j|S}^{(1)})^2 = (\sigma_{j|S}^{(2)})^2$, again using an F-test. Given $j \in [p]$ and $S \subseteq [p] \setminus \{j\}$, the residual variance $(\sigma_{j|S}^{(k)})^2$ is defined as the variance of the regression residual when regressing $X_j^{(k)}$ onto the random vector $X_S^{(k)}$. In fact it holds that $\sigma_i^{(1)} = \sigma_i^{(2)}$ if and only if there exists a subset $S \subseteq \mathcal{C} \setminus \{i\}$ such that $\sigma_{i|S}^{(1)} = \sigma_{i|S}^{(2)}$ and if $i \to j$ in Δ then $j \notin S$, whereas

Algorithm 1 Difference Causal Inference (DCI) algorithm (`dci` function)

```
Input: Sample data $\hat{X}^{(1)}$, $\hat{X}^{(2)}$.
Output: Estimated difference-DAG $\hat{\Delta}$.
Initialize with difference undirected graph $\bar{\Delta}$ and conditioning set $\mathcal{C}$.
Estimate the skeleton of the difference-DAG $\tilde{\Delta}$ using Algorithm 2.
Direct edges in $\tilde{\Delta}$ using Algorithm 3 to obtain $\hat{\Delta}$.
```

Algorithm 2 Estimating skeleton of the difference-DAG (`dci_skeleton` function)

```
Input: Sample data $\hat{X}^{(1)}$, $\hat{X}^{(2)}$, estimated difference undirected graph $\bar{\Delta}$ and conditioning set $\mathcal{C}$.
Output: Estimated skeleton $\tilde{\Delta}$.
Set $\tilde{\Delta} := \bar{\Delta}$;
for each edge $i - j$ in $\tilde{\Delta}$ do
    If $\exists S \subseteq \mathcal{C} \setminus \{i, j\}$ such that $\beta^{(k)}_{i,j|S}$ is invariant across $k = \{1, 2\}$, delete $i - j$ in $\tilde{\Delta}$ and continue to the next edge. Otherwise, continue.
end for
```

Algorithm 3 Directing edges in the difference-DAG (`dci_orient` function)

```
Input: Sample data $\hat{X}^{(1)}$, $\hat{X}^{(2)}$, estimated skeleton $\tilde{\Delta}$ and conditioning set $\mathcal{C}$.
Output: Estimated difference-DAG $\hat{\Delta}$.
Set $\hat{\Delta} := \emptyset$;
for each node $j$ incident to at least one undirected edge in $\tilde{\Delta}$ do
    If $\exists S \subseteq \mathcal{C} \setminus \{j\}$ such that $\sigma^{(k)}_{j|S}$ is invariant across $k = \{1, 2\}$, add $i \to j$ to $\hat{\Delta}$ for all $i \in S$, and add $j \to i$ to $\hat{\Delta}$ for all $i \notin S$ and continue to the next node. Otherwise, add $i - j$ to $\hat{\Delta}$.
end for
Orient as many undirected edges as possible via graph traversal using the following rule:
    Orient $i - j$ as $i \to j$ whenever there is a chain $i \to \ell_1 \to \cdots \to \ell_t \to j$.
```

if $j \to i$ in Δ then $j \in S$ [1]. Hence determining conditioning sets that lead to the invariance of residual variances can be used to orient some of the edges in the difference-DAG Δ.

DCI with stability selection

Running DCI requires choosing several hyperparameters, namely the ℓ_1-regularizer for estimating the difference undirected graph via KLIEP [3] as well as the significance levels for hypothesis testing of invariance of regression coefficients as well as residual variances. We implemented DCI with stability selection to address the issue of choosing the correct hyperparameters. Stability selection was introduced by [9] and has been successfully applied in tandem with other causal inference methods [10]. The idea behind stability selection is to choose the most stable estimate across different hyperparameters as opposed to focusing on choosing the right value for the hyperparameters.

Algorithm 4 outlines the methodology for running DCI with stability selection. Let Λ denote the set of considered hyperparameter values consisting of ℓ_1 regularizers for KLIEP, significance levels for hypothesis testing of invariance of regression coefficients and significance levels for hypothesis testing of invariance of residual variances. Given a particular $\lambda \in \Lambda$, we can run DCI (Algorithm 1) and obtain the corresponding estimated difference causal graph $\hat{\Delta}^{\lambda}$. Stability selection relies on estimating the probability of selection of each edge $\hat{\Pi}^{\lambda}_{k}$ by running the DCI algorithm on subsamples of the data. Aggregating selection probabilities across different tuning parameters $\lambda \in \Lambda$, we keep edges with high selection probability as the stable set of estimated edges in the difference-DAG $\hat{\Delta}^{\text{stable}}$.

Evaluation on real data

We evaluate DCI for learning the causal difference gene regulatory network on single-cell gene expression data and quantify its performance in predicting the effects of gene perturbations. Note that a major advantage of our work is the ability to learn a causal as opposed to an undirected graph, which enables us to predict the effects of interventions on genes and evaluate them against true effects of interventions,

Algorithm 4 DCI with stability selection (dci_stability_selection function)

```
Input: Sample data $\hat{X}^{(1)}$, $\hat{X}^{(2)}$, set of tuning parameters $\Lambda$, number of subsamples $N$ of size given by the fraction $f$ of all samples, and threshold $\pi_{\text{thr}}$ for choosing stable variables.
Output: Stable estimate of difference-DAG $\hat{\Delta}^{\text{stable}}$.
for each $\lambda$ in $\Lambda$ do
  for each $i$ in $1, \ldots, N$ do
    Generate subsamples of the two datasets, $\hat{X}^{(1)}_{(i)}$ and $\hat{X}^{(2)}_{(i)}$ (without replacement) of size defined by the fraction $f$ of the full samples size.
    Run Algorithm 1 on $\hat{X}^{(1)}_{(i)}$ and $\hat{X}^{(2)}_{(i)}$ with hyperparameters $\lambda$ to obtain $\hat{\Delta}^{\lambda}_{(i)}$.
  end for
  Calculate selection probability for each edge $k$ by $\hat{\Pi}^{\lambda}_{k} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\{k \in \hat{\Delta}^{\lambda}_{(i)}\}$.
end for
Construct stable estimate of difference-DAG $\hat{\Delta}^{\text{stable}} = \{k : \max_{\lambda \in \Lambda} \hat{\Pi}^{\lambda}_{k} \geq \pi_{\text{thr}}\}$.
```

measured experimentally. In the following, we assess the performance of DCI on two datasets collected via CROP-seq [11] and Perturb-seq [12]. Both of these experimental techniques collect, in a pooled fashion, single-cell gene expression data with no interventions (observational data) as well as single-cell gene expression data where some genes were knocked out via CRISPR/Cas9 (interventional data). We use the observational data to learn a causal difference gene regulatory network via DCI and evaluate this graph against the held-out CRISPR/Cas9 gene knockouts, similar in spirit to prior evaluations of causal inference methods [13].

CROP-seq: Naive versus activated T cells

We test our method on gene expression data collected via CROP-seq for naive and activated Jurkat T cells. In particular, we use DCI to learn the differences in the gene regulatory networks as a result of T-cell activation. The CROP-seq data includes 615 observational naive Jurkat T cells and 1320 observational activated Jurkat T cells. As in the original CROP-seq study [11], we normalize the gene expression of each cell by the total number of reads corresponding to the cell, scale expression by 10^4 and apply a $\log_2(x+1)$ transformation to the data. The data is mean-centered prior to applying our algorithm. We follow [11] in focusing on genes most relevant to T-cell activation and keep genes that have non-zero variance, resulting in 31 genes.

We apply DCI on the observational naive and activated gene expression data to directly obtain the causal difference gene regulatory network (difference-DAG), which contains edges that appeared, disappeared or changed weight between the two cell states. We report the performance of DCI when initialized in the complete graph as well as when initialized with the difference undirected graph estimated via KLIEP (ℓ_1 regularization set to 0.005). Additionally, we compared the performance of DCI to the naive approach of running classical causal inference algorithms such as PC [7] or GES [8] on each dataset (naive and activated) separately, obtaining two causal graphs and then taking the difference. We consider an edge to be in the difference-DAG if the edge was directed in one causal graph and absent in the other causal graph.

As previously mentioned, we can use gene knockouts, collected as part of the CROP-seq study for evaluation of the causal difference gene regulatory network. Note that if perturbing a gene affected the gene expression distribution of another gene, this means that the perturbed gene is upstream of the affected gene in the gene regulatory network. In the following we describe how we estimate the differences in the effects of CRISPR/Cas9 perturbations on genes between the two states (naive and activated) to construct an ROC curve for evaluating the DCI algorithm versus naive applications of PC and GES.

First, for each condition (naive and activated), we separately obtain a matrix that describes which gene knockouts had an effect on which genes (Figures S1a and S1b). Then, we take the difference between these matrices to determine the differences in the effects of perturbations (Figure S1c). In order to construct the matrices in Figures S1a and S1b, for each condition, we estimate the impact of each gene deletion $j \in \{1, \ldots, d\}$ on each of the measured genes $i \in \{1, \ldots, p\}$ by testing whether the observational distribution (no intervention) of the measured gene i is significantly different from the interventional distribution of the measured gene i when gene j was deleted using a Wilcoxon rank-sum test. We form a $p \times d$ matrix of p-values, Q, from the Wilcoxon rank-sum tests. Next, each column j in Q is thresholded using the entry q_{jj}, which is the p-value obtained by comparing the distribution of the gene expression level of a deleted gene versus its distribution without intervention. The rationale is that knocking out a particular gene should result in a change in its own gene expression distribution and can be used as a baseline to threshold the other entries in the column. In particular, we conclude that q_{ij} is significant if and only if $q_{ij} \leq q_{jj}$. After thresholding the matrix Q in this manner, we obtain the binary matrices in Figures S1a and S1b, which summarize the effects of the interventions. By forming the difference of these binary matrices we obtain the binary matrix Q^{Δ} in Figure S1c. Since not all CRISPR/Cas9 knockouts were effective, here we focused our analysis on the top most effective interventions, which were prioritized based on the maximum q_{jj} p-value (taken over two conditions), using the mean p-value as the cutoff to filter interventions.

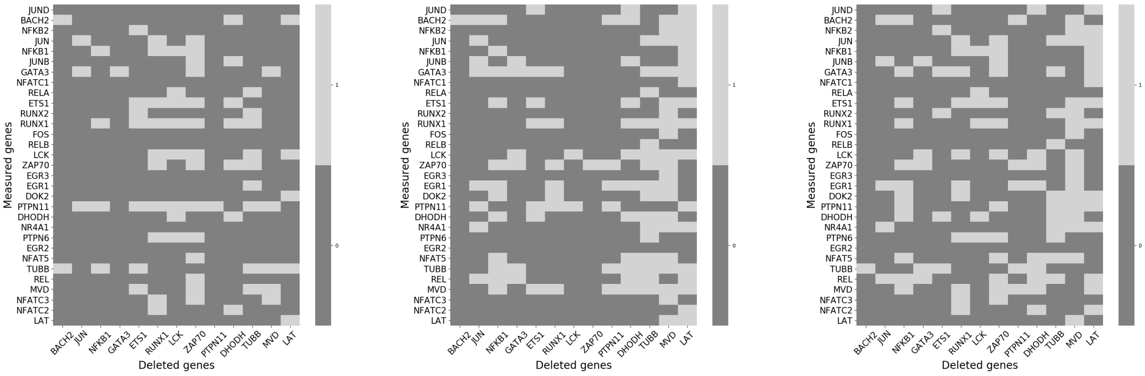
We use the matrix of differences in the effects of interventions to evaluate DCI, PC and GES by constructing an ROC curve. If the predicted difference-DAG has a directed edge from j to i, we count this edge as a true positive if $Q^{\Delta}_{ij} = 1$, i.e. there was a difference in the effect of knocking out gene j on gene i between the two conditions. If the predicted difference-DAG has a directed edge from $j \rightarrow i$ but $Q^{\Delta}_{ij} = 0$, the edge is counted as a false positive. Note that this definition of a false positive is overly conservative, since we may have $Q^{\Delta}_{ij} = 0$ if q_{ij} is significant in both matrices, but the magnitude of the effect changes. In other words, $Q^{\Delta}_{ij} = 1$ only captures additions/deletions of edges, but does not capture changes in edge weights. We construct an ROC curve by varying the parameters of DCI, PC and GES. The ROC curve in Figure S2 shows that DCI outperforms PC and GES in predicting the effects of interventions on this single-cell gene expression dataset. In Figure S3, we include examples of the estimated difference gene regulatory networks inferred via DCI (our algorithm) and GES (the best performing baseline).

Perturb-seq: Dendritic cells at 0 versus 3 hours post-stimulation

We perform a similar evaluation of DCI on gene expression data collected as part of the Perturb-seq dataset [12]. Gene expression data was collected from bone-marrow derived dendritic cells (BMDCs) pre-stimulation (0 hours) and after stimulation with LPS (3 hours). We applied DCI to learn the difference gene regulatory network between these two time points. We used the same procedure for pre-processing Perturb-seq data as we used for CROP-seq. Additionally, we filtered cells for quality, only keeping cells with at least two nonzero counts (CROP-seq dataset already satisfied this filtering constraint). The filtered Perturb-seq data includes 940 observational cells collected at 0 hours and 990 observational cells collected at 3 hours. We followed [12] in focusing on 24 transcription factors that are important for dendritic cell regulation.

Using the same procedure as performed on the CROP-seq dataset, we constructed the binary matrices describing the effects of gene deletions on measured genes for the two time points (0 and 3 hours) separately, shown in Figures S4a and S4b, and then determined the difference in the effects of the interventions between the two time points in Figure S4c. As above, we constructed an ROC curve, taking the differences in the effects of interventions as the ground truth. The ROC curve (Figure S5) shows that in the majority of settings, DCI outperforms the naive approach of estimating two causal graphs separately via PC or GES and taking the difference of the output graph. Figure S6 shows examples of the estimated difference gene regulatory networks inferred via DCI (our algorithm) and GES (best performing baseline).

SI Figures



(a)

(b)

(c)

Fig. S1: Effects of gene deletions estimated from CROP-seq data; (a) naive T cells, (b) activated T cells, and (c) the difference between the binary matrices in (a) and (b), i.e., the difference in the effects of each gene deletion on the measured genes between naive and activated T cells; this binary matrix is taken to be the ground truth for constructing ROC curves.

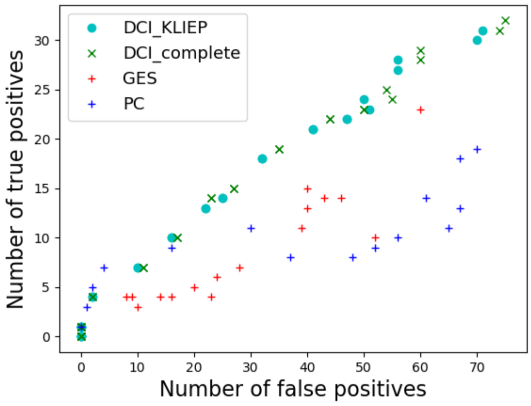
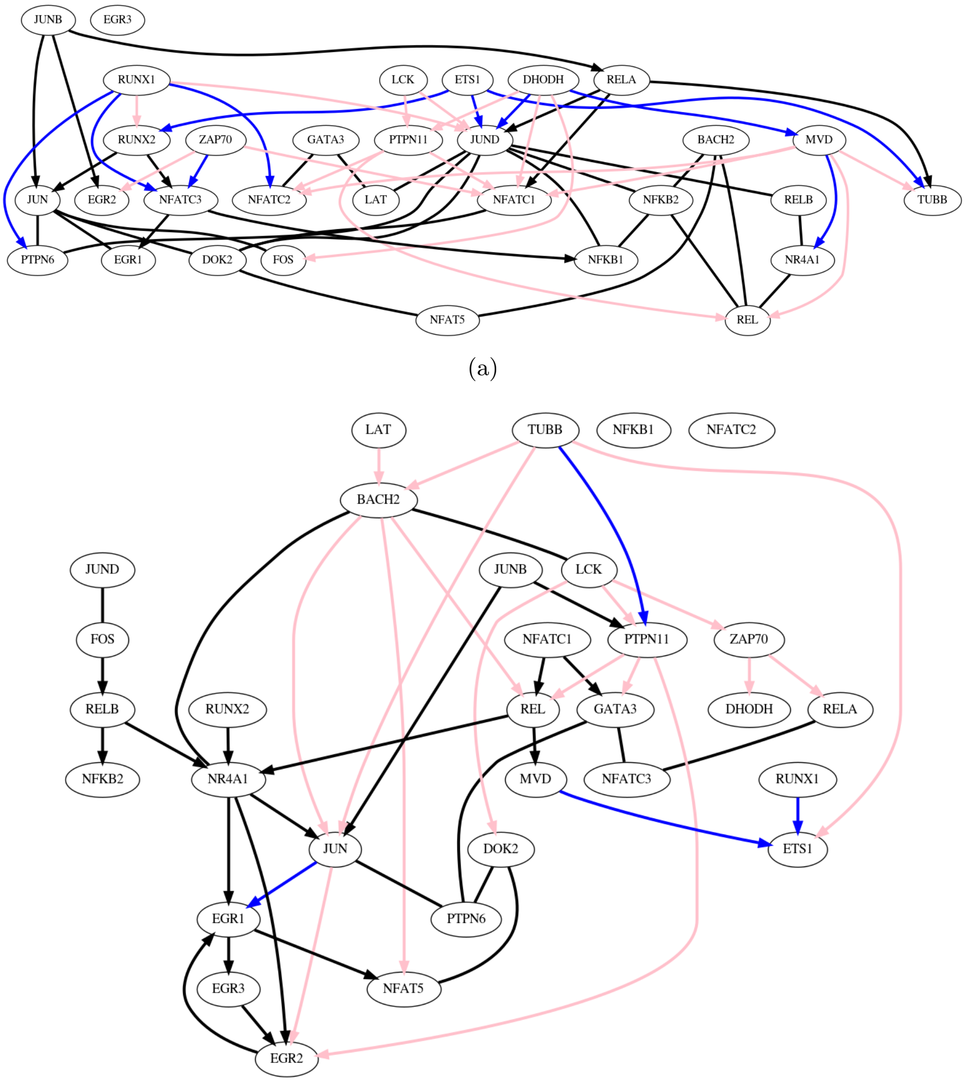


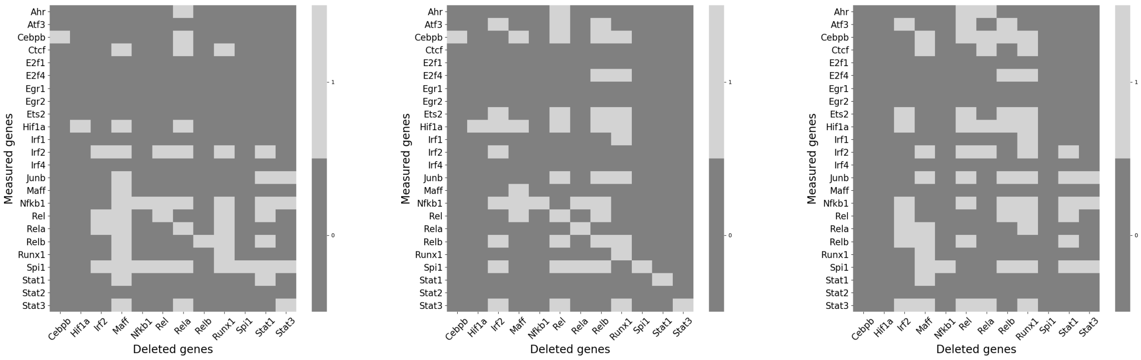
Fig. S2: ROC plot evaluating DCI (initialized in the undirected difference graph estimated via KLIEP as well as in the complete graph), GES and PC on the CROP-seq data for predicting the differences in the effects of gene knockouts.



(a)

(b)

Fig. S3: Examples of difference gene regulatory networks between naive and activated Jurkat T cells, estimated from the CROP-seq data. Difference gene regulatory network inferred via (a) our algorithm, DCI, initialized with KLIEP, which directly learns the difference causal graph from two datasets and (b) baseline causal structure discovery algorithm, GES, which estimates two gene regulatory networks separately and then takes the difference. Blue edges indicate true positives and pink edges indicate false positives. Black edges are the edges inferred to be in the difference gene regulatory network for which ground truth is not available. Graphs were chosen such that the number of false positives is the same across the two methods (16 false positives).



(a)

(b)

(c)

Fig. S4: Effects of gene deletions estimated from Perturb-seq data; (a) before stimulation with LPS, (b) after stimulation with LPS, and (c) the difference between the binary matrices in (a) and (b), i.e., the difference in the effects of each gene deletion on the measured genes before and after stimulation with LPS; this binary matrix is taken to be the ground truth for constructing ROC curves.

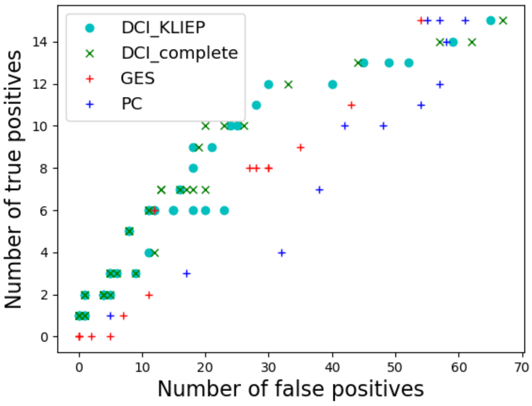
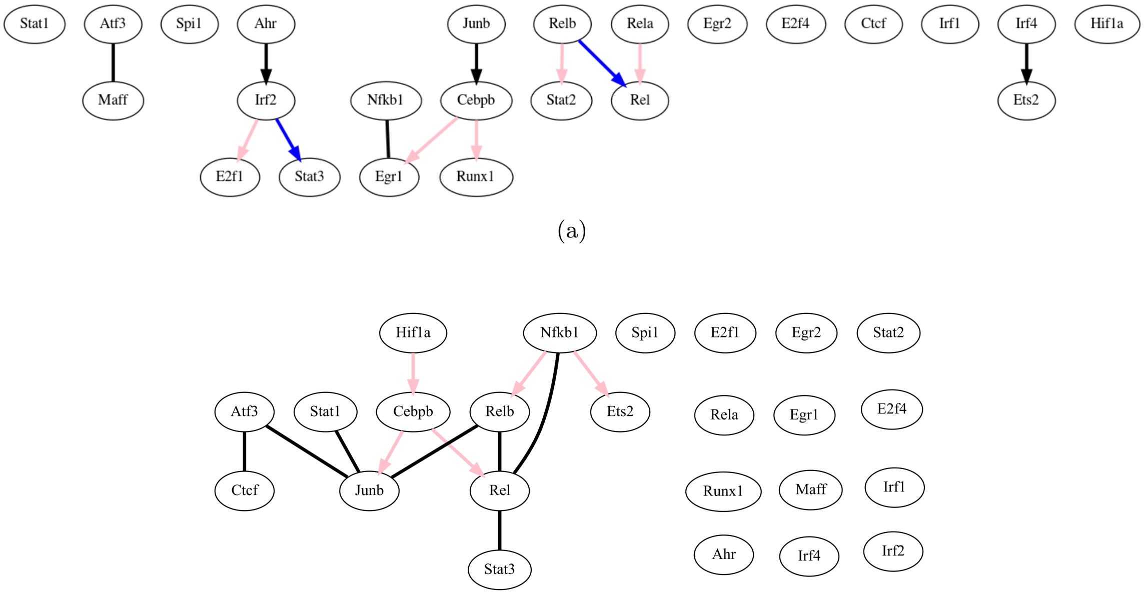


Fig. S5: ROC plot evaluating DCI (initialized in the undirected difference graph estimated via KLIEP as well as in the complete graph), GES and PC on the Perturb-seq data for predicting the differences in the effects of gene knockouts.



(a)

(b)

Fig. S6: Examples of difference gene regulatory networks of dendritic cells before and after stimulation with LPS, estimated from the Perturb-seq data. Difference gene regulatory network inferred via (a) our algorithm, DCI, initialized with KLIEP, which directly learns the difference causal graph from two datasets and (b) baseline causal structure discovery algorithm, GES, which estimates two gene regulatory networks separately and then takes the difference. Blue edges indicate true positives and pink edges indicate false positives. Black edges are the edges inferred to be in the difference gene regulatory network for which ground truth is not available. Graphs were chosen such that the number of false positives is the same across the two methods (5 false positives).

References

1. Wang, Y., Squires, C., Belyaeva, A. & Uhler, C. *Direct estimation of differences in causal graphs* in *Advances in Neural Information Processing Systems* (2018), 3770–3781.
2. Liu, S., Quinn, J. A., Gutmann, M. U., Suzuki, T. & Sugiyama, M. Direct learning of sparse changes in Markov networks by density ratio estimation. *Neural Computation* **26,** 1169–1197 (2014).
3. Liu, S., Fukumizu, K. & Suzuki, T. Learning sparse structural changes in high-dimensional Markov networks. *Behaviormetrika* **44,** 265–286 (2017).
4. Zhao, S. D., Cai, T. T. & Li, H. Direct estimation of differential networks. *Biometrika* **101,** 253–268 (2014).
5. Fukushima, A. DiffCorr: an R package to analyze and visualize differential correlations in biological networks. *Gene* **518,** 209–214 (2013).
6. Lichtblau, Y. *et al.* Comparative assessment of differential network analysis methods. *Briefings in Bioinformatics* **18,** 837–850 (2017).
7. Spirtes, P., Glymour, C. N. & Scheines, R. *Causation, Prediction, and Search* (MIT press, 2000).
8. Meek, C. *Graphical Models: Selecting Causal and Statistical Models* PhD thesis (Carnegie Mellon University, 1997).
9. Meinshausen, N. & Bühlmann, P. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **72,** 417–473 (2010).
10. Meinshausen, N. *et al.* Methods for causal inference from gene perturbation experiments and validation. *Proceedings of the National Academy of Sciences* **113,** 7361–7368 (2016).
11. Datlinger, P. *et al.* Pooled CRISPR screening with single-cell transcriptome readout. *Nature Methods* **14,** 297 (2017).
12. Dixit, A. *et al.* Perturb-Seq: dissecting molecular circuits with scalable single-cell RNA profiling of pooled genetic screens. *Cell* **167,** 1853–1866 (2016).
13. Wang, Y., Solus, L., Yang, K. & Uhler, C. *Permutation-based causal inference algorithms with interventions* in *Advances in Neural Information Processing Systems* (2017), 5822–5831.