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

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

[1]Laboratory for Information & Decision Systems and Institute for Data, Systems and Society,
Massachusetts Institute of Technology, Cambridge, 02139, USA

*To whom correspondence should be addressed.

This PDF file includes:

Supplementary Note
Figures S1 to S9
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 $[p] := \{1, \ldots, p\}$ 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 our constraint-based method outlined below, 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 $\mathcal{C} = \{i \mid \exists j \in [p] \text{ such that } \Theta_{i,j}^{(1)} \neq \Theta_{i,j}^{(2)}\}$. The reduced node set can be determined from the output of methods such as our constraint-based method, KLIEP [2–6], prior biological knowledge, or taken as the set of all nodes when the number of genes to be considered is small.

Our constraint-based method for determining the difference undirected graph estimates the precision matrices corresponding to each dataset, $\hat{\Theta}^{(1)}$ and $\hat{\Theta}^{(2)}$, from data and computes the following test statistic for each entry (i, j) to quantify the difference:

$$\hat{Q_{ij}} := \left(\hat{\Theta}_{ij}^{(1)} - \hat{\Theta}_{ij}^{(2)}\right)^2 \cdot \left(\frac{\hat{\Theta}_{ii}^{(1)}\hat{\Theta}_{jj}^{(1)} + (\hat{\Theta}_{ij}^{(1)})^2}{n_1} + \frac{\hat{\Theta}_{ii}^{(2)}\hat{\Theta}_{jj}^{(2)} + (\hat{\Theta}_{ij}^{(2)})^2}{n_2}\right)^{-1}.$$

In order to determine whether a particular edge should remain as part of the undirected difference graph, $\hat{Q_{ij}}$ is tested for fit to the F-distribution with parameters $F(1, n_1 + n_2 - 2p + 2)$ and the edge remains if the null hypothesis is rejected. As described in [1], this hypothesis testing framework comes from the facts that (1) the entry $\hat{\Theta}_{ij}$ converges asymptotically to a multivariate normal centered at the true parameter, with variance $\Theta_{ii}\Theta_{jj} + \Theta_{ij}^2$, (2) the difference between two independent standard normal random variables follows a χ^2 distribution, and (3) the F-distribution with the suggested parameters converges asymptotically to a χ^2 distribution, but the fatter tails of the F-distribution better match the finite sample distribution of the test statistic [7].

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

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 $\mathcal{C}$, maximum conditioning set size $r$.
Output: Estimated skeleton $\tilde{\Delta}$.
Set $\tilde{\Delta} := \bar{\Delta}$;
for each edge $i - j$ in $\tilde{\Delta}$ do
  If $\exists S \subseteq \mathcal{C} \setminus \{i, j\}$, with $|S| \leq r$, 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
```

such that $\beta^{(1)}_{i,j|S} = \beta^{(2)}_{i,j|S}$, then $B^{(1)}_{ij} = B^{(2)}_{ij}$ 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 [8] and GES [9] 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 directions of all edges adjacent to nodes whose internal node variances are unchanged across the two conditions, i.e. for which $\sigma^{(1)}_i = \sigma^{(2)}_i$ [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 $(\sigma^{(k)}_{j|S})^2$ and testing whether they are invariant, i.e. whether $(\sigma^{(1)}_{j|S})^2 = (\sigma^{(2)}_{j|S})^2$, again using an F-test. Given $j \in [p]$ and $S \subseteq [p] \setminus \{j\}$, the residual variance $(\sigma^{(k)}_{j|S})^2$ is defined as the variance of the regression residual when regressing $X^{(k)}_j$ onto the random vector $X^{(k)}_S$. In fact it holds that $\sigma^{(1)}_i = \sigma^{(2)}_i$ if and only if there exists a subset $S \subseteq \mathcal{C} \setminus \{i\}$ such that $\sigma^{(1)}_{i|S} = \sigma^{(2)}_{i|S}$ and if $i \to j$ in Δ then $j \notin S$, whereas 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. Algorithm 3 is a modification of the original `dci_orient` algorithm introduced in [1], which ensures that the output is *order-independent*. In particular, the output of the original `dci_orient` algorithm introduced in [1] depends on the order in which one iterates over nodes, which can lead to biased or inconsistent results. The modified version in Algorithm 3 eliminates this issue by simultaneously considering all nodes at each level of conditioning set size. In Algorithm 3 we note that $\mathrm{pval}(\sigma^{(1)}_{j|S} = \sigma^{(2)}_{j|S})$ refers to the p-value obtained from the F-test to determine the invariance of residual variances. We provide both versions of the `dci_orient` algorithm in the `causaldag` package.

An issue that can arise in practice when applying the original DCI method to gene expression data is due to the need to compute a test statistic $\hat{T}$ that depends on the inverse of the sample covariance matrix [1]. This inverse may not exist, since gene expression data is often high-dimensional with more genes than samples, in particular when the data is subsampled for stability selection, and the matrix can have many zeros (possibly leading to variance zero for a node) due to dropout. In this case, we use the pseudoinverse instead of the inverse to compute the test statistics.

DCI with stability selection

Running DCI requires choosing several hyperparameters, namely the ℓ_1-regularizer for estimating the difference undirected graph via KLIEP [3] (or significance levels for the constraint-based algorithm) 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 [10] and has been successfully applied in

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 $\mathcal{C}$, maximum conditioning set size $r$.
Output: Estimated difference-DAG $\hat{\Delta}$.
Set $\hat{\Delta} := \emptyset$;
for conditioning set size $k = 1, \dots, r$ do
  Set $\mathcal{V}$ to all nodes $j$ incident to at least one undirected edge in $\bar{\Delta}$
  For each $j \in \mathcal{V}$, let $p_j = \max_{S \subseteq \mathcal{C} \setminus \{j\} : |S| = k} \operatorname{pval}(\sigma^{(1)}_{j|S} = \sigma^{(2)}_{j|S})$
  while $\mathcal{V} \neq \emptyset$ do
    Let $j = \arg\max_{j' \in \mathcal{V}} p_j$
    If $p_j > \alpha$, set the corresponding $S$ as the parent set for $j$ in $\hat{\Delta}$, and the remaining adjacent nodes to $j$ as its children in $\hat{\Delta}$, as long as this does not create any cycles or contradict any existing edges.
    Let $\mathcal{V} = \mathcal{V} \setminus \{j\}$
  end while
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 $i \to \ell_1 \to \cdots \to \ell_t \to j$.
```

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 $\hat{\Delta}^{\text{stable}}$ and selection probabilities $\hat{\Pi}^{\lambda}_k$.
for each $\lambda$ in $\Lambda$ do
  for each $i$ in $1, \dots, N$ 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 $\hat{\Delta}^{\text{stable}} = \{k : \max_{\lambda \in \Lambda} \hat{\Pi}^{\lambda}_k \geq \pi_{\text{thr}}\}$.
```

tandem with other causal inference methods [11]. 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 (or significance levels for the constraint-based algorithm), 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 $\hat{\Delta}^{\text{stable}}$. We note that DCI with stability selection outputs a single estimate of the causal graph along with selection probabilities of each edge (see Figure S4c and Figure S7c). These probabilities can be used to infer the trustworthiness of each edge in the graph.

Stability selection alleviates the need for the user to rely on the results from a single hyperparameter in DCI. In addition, the user can see how the selection probability of edges vary in comparison to each other across a range of hyperparameters. Such a plot is shown in Figure S1, where the significance level for hypothesis tests in the difference skeleton discovery phase is varied from $\alpha = 10^{-5}$ to $\alpha = 0.1$, and the probability that each edge is included in the difference skeleton is given on the vertical axis. Additionally, for each edge, we may use a heatmap to indicate the probability that it appears in either one or the other orientation in the difference-DAG. This provides additional information on how much the user should trust the presence of an edge: a true edge should consistently be oriented in its correct orientation, and thus inconsistent orientation of an edge across different subsamples of data indicates that it is a false positive. In Figure S1, true positive edges are colored according to the probability that they appear in their correct orientation, while false positive edges are colored according to the proportion of times that they occur in an arbitrary fixed direction. It is apparent from Figure S1 that the three edges (a-c) are significantly more likely than the others to belong to the difference skeleton, and that all of these edges have a consistent orientation, which suggests that these are true positives, and indeed they are. Meanwhile, the fourth and fifth most likely edges (d, e) have inconsistent orientations, and they indeed do not correspond to edges in the true difference-DAG, while the sixth most likely edge (f) has a more consistent orientation and is indeed a true positive. This information allows the user to select a trade-off between false positives and true positives that is suitable for their application, e.g. a conservative rule which excludes any edges which themselves have inconsistent orientations or lay below an edge with inconsistent orientations would still get three of the four true positives correct, while more liberal rules would pick up all four true positives while only returning a small number of false positives.

DCI with stability selection is optimized to run in parallel on multiple cores across the different bootstrap subsamples. In addition, since the DCI skeleton learning phase has a monotonicity property, i.e. if an edge is absent in the difference skeleton for some α, then it is absent in the difference skeleton for all $\alpha' < \alpha$, the DCI skeleton discovery phase is run simultaneously for all significance levels to speed up computation.

Limitations and Extensions

One limitation of our method is the assumption of a linear-Gaussian model for gene expression data, which may exhibit complicated nonlinear relationships. Indeed, prior work [12] has demonstrated the utility of removing the linear Gaussian assumption when working with gene expression data. While the present work does not investigate nonlinear models, it would be straightforward to extend the current algorithm to the nonlinear setting. For instance, by allowing each function in the structural causal model to be a generalized additive model (GAM), we can associate a vector to each edge, representing the coefficients of each basis function in the model. Then, we can define the difference-DAG by including an edge whenever at least one of these coefficients changes between two settings. Finally, we could modify the algorithm to use GAM regression and hypothesis tests for the invariance of this whole vector. For more complicated models with interaction terms, the difference-DAG would need to be defined slightly differently, but the same ideas still carry through. As for the assumption of Gaussian noise, our test statistic remains valid even for non-Gaussian noise, but we may no longer be able to compute confidence intervals, in which case stability selection may be even more important to obtain robust results.

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, measured experimentally. In the following, we assess the performance of DCI on two datasets collected via CROP-seq [13] and Perturb-seq [14]. 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 [12, 15, 16].

Evaluating causal graphs against interventions

Interventional data such as CRISPR/Cas9 gene knockouts provide information on ancestral relationships between genes, e.g. whether a particular gene is upstream or downstream of another gene, since knocking out a particular gene should have downstream effects (i.e., on a gene's descendants) but no effect on any upstream genes (i.e., a gene's ancestors). Interventional data has been previously used for evaluating causal graphs through the construction of ROC curves that measure the number of true and false positives in the estimated graph across different hyperparameters [12, 15, 16]. A typical approach is to denote an edge $j \to i$ in the estimated causal graph as a potential true positive if (based on interventional data) knocking out gene j has an effect on gene i, and a false positive otherwise. While we use such an analysis also in this work, we note that it only provides indirect support for an edge since interventional data cannot provide information on which genes are the direct parents of other genes. Thus only a subset of the edges that we consider as potential true positives are real true positives and hence the resulting ROC curves should only be viewed as an indication of an algorithm's performance across hyperparameters. In addition to this analysis, we also provide estimated networks (obtained via stability selection across different hyperparameters) and show that in each estimated network the edge with the highest confidence corresponds to a pair of nodes where the ancestral relationship does indeed change.

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 [13], 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 [13] 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) and the constraint-based method (significance level set to 0.8). Additionally, we compared the performance of DCI to the naive approach of running classical causal inference algorithms such as PC [8] or GES [9] 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 S2a and S2b). Then, we take the difference between these matrices to determine the differences in the effects of perturbations (Figure S2c). In order to construct the matrices in Figures S2a and S2b, for each condition, we estimate the impact of each gene deletion $j \in \{1, \dots, d\}$ on each of the measured genes $i \in \{1, \dots, p\}$ 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 S2a and S2b, which summarize the effects of the interventions. By forming the difference of these binary matrices we obtain the binary matrix Q^{Δ} in Figure S2c. 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 S3 shows that DCI outperforms PC and GES in predicting the effects of interventions on this single-cell gene expression dataset. In order to quantify the improvement of the DCI algorithm over the naive approaches, we report a p-value quantifying the difference from random guessing. On the CROP-seq dataset, the PC algorithm achieved a p-value of 0.46, GES a p-value of 0.21, DCI_complete a p-value of 1.75×10^{-18}, DCI_KLIEP a p-value of 7.32×10^{-20}, and DCI_constraint a p-value of 1.45×10^{-20}. The p-value is calculated by sampling causal graphs from a Barabasi–Albert preferential attachment model and quantifying the number of true and false positives. For each false positive level, we created a distribution over true positives based on the sampled random causal graphs and calculated the p-value for the number of true positives obtained from the PC, GES and DCI algorithms. The p-values were combined using Fisher's method and this combined p-value was used for evaluating the causal algorithms. In Figures S4a,b, we include examples of the estimated difference gene regulatory networks inferred via DCI (our algorithm) and GES (the best performing baseline). In Figure S4c we additionally include the output of DCI with stability selection, which was applied over the following combination of hyperparameters: KLIEP ℓ_1 regularization $\in \{0.001, 0.005, 0.01\}$, α-level for skeleton discovery $\in \{0.01, 0.1, 0.2\}$, and α-level for orientation discovery $\in \{0.01, 0.1, 0.2\}$. For each combination of the three hyperparameters, the data was subsampled $N = 50$ times, keeping $f = 0.7$ fraction of samples and the DCI algorithm was applied. The selection probability of each edge was estimated across the subsamples and aggregated across the different combinations of hyperparameters. To validate the resulting network, we investigate the directed edge with the highest probability of selection using knockout data. For the CROP-seq dataset, the highest probability edge points from ETS1 to TUBB. The CRISPR/Cas9 knockout data (shown in Figure S2) indicates that while in naive T-cells, knocking out ETS1 affects the expression of TUBB, which means that ETS1 is upstream of TUBB, in activated T-cells knocking out ETS1 does not affect the expression of TUBB, which means that ETS1 is no longer upstream of TUBB. This change in the effect of ETS1 gene deletion indicates that there is indeed a change in the gene regulatory network between the two cell states, thereby providing some evidence for the edge between ETS1 and TUBB in the difference causal graph. While based on the causal graph shown in Figure S4c, we inferred that ETS1 is directly upstream (i.e., a parent) of TUBB, we note that since knockout experiments only give evidence for a gene being upstream or downstream of another gene, there are other possibilities for causal paths between ETS1 and TUBB (e.g. not just a direct parent) that would be consistent with the knockout data.

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 [14]. 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 [14] 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 S5a and S5b, and then determined the difference in the effects of the interventions between the two time points in Figure S5c. As above, we constructed an ROC curve, taking the differences in the effects of interventions as the ground truth. The ROC curve (Figure S6) 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 graphs. On the Perturb-seq dataset, the PC algorithm achieved a p-value of 0.66, GES a p-value of 0.6×10^{-2}, DCI_complete a p-value of 3.61×10^{-8}, DCI_KLIEP a p-value of 1.18×10^{-9}, and DCI_constraint a p-value of 3.87×10^{-10}. Figure S7a,b shows examples of the estimated difference gene regulatory networks inferred via DCI (our algorithm) and GES (best performing baseline). Figure S7c additionally shows the causal graph obtained from running DCI with stability selection. To validate the resulting network, we investigate the directed edge with the highest probability of selection using knockout data. For the Perturb-seq dataset, the highest probability edge points from Relb to Rel. The CRISPR/Cas9 knockout data (shown in Figure

S5) indicates that while in cells before LPS stimulation, knocking out Relb does not affect the expression of Rel, which means that Relb is not upstream of Rel, in cells after LPS stimulation, knocking out Relb affects the expression of Rel, which means that Rel is upstream of Relb. This change in the effect of Relb gene deletion indicates that there is indeed a change in the gene regulatory network between the two cell states, thereby providing some evidence for the edge between Relb and Rel in the difference causal graph. In addition, based on the causal graph shown in Figure S7c, we inferred that Relb is directly upstream (i.e., a parent) of Rel, indicating that the change in regulation between Relb and Rel indeed likely occurred due to a change in the directed edge between these two genes and not due to a change in a longer path between the two genes.

Time complexity comparison

We assess the run time of DCI as compared to the naive approach of estimating each graph separately via PC or GES and then taking the difference on simulated data. For this, we generate 10 different pairs of ground truth causal graphs and sample 10 pairs of datasets from these graphs. For the generation of causal graphs, we sample a weighted adjacency matrix $B^{(1)}$ using an Erdös-Renyi model with expected neighbourhood size of 10, on p nodes. The weights are uniformly drawn from $[-1, -0.25] \cup [0.25, 1]$ to ensure that they are bounded away from zero. The second weighted adjacency matrix $B^{(2)}$ is constructed from $B^{(1)}$ by adding and removing 5 edges (10 changes in total). We sample datasets $\hat{X}^{(1)}$ and $\hat{X}^{(2)}$ from the distribution induced by the Gaussian DAG models with $n = 100,000$ samples. Next, we run DCI ($\alpha_{\text{undirected}} = 0, \alpha_{\text{skeleton}} = 0.1, \alpha_{\text{orient}} = 0.1$), PC ($\alpha = 1 \times 10^{-6}$) and GES ($\lambda = 1000$) on $\hat{X}^{(1)}$ and $\hat{X}^{(2)}$ and evaluate the CPU time (in seconds) as well as the number of true and false positives averaged over the 10 simulations. As shown in Figure S8, DCI is much faster than PC and GES (in terms of mean CPU time) and significantly more accurate as indicated by the average number of true and false positives for each setting. For example, with 500 nodes, on average, DCI runs in 45 *seconds* and results in 9 true positives (out of 10 possible) and 1 false positive, while GES runs in 0.67 *hours* and results in 9.67 true positives and 1529.56 false positives, and PC runs in 4 *hours* and results in 2.33 true positives and 187.56 false positives. On 10,000 nodes, DCI ran for 6.84 *hours* and resulted in 10 true positives and 2 false positives. To analyze the time complexity of PC, GES and DCI, Figure S8 is shown on a log-log scale. On such a scale, a linear curve corresponds to a polynomial time algorithm with $O(p^q)$, where the slope of the line indicates the exponent q of polynomial growth. PC, GES and DCI appear to have approximately linear curves with slopes (q) of 2.8, 1.8, and 1.1, respectively, as estimated by fitting a linear function and obtaining its slope. Figure S9 shows the CPU time of DCI for varied parameter settings, which control the sparsity of the output given by the different steps in the DCI algorithm. We performed these benchmark experiments on an Intel(R) Core(TM) i7-6950X CPU @ 3.00GHz machine with 125GiB of memory.

SI Figures

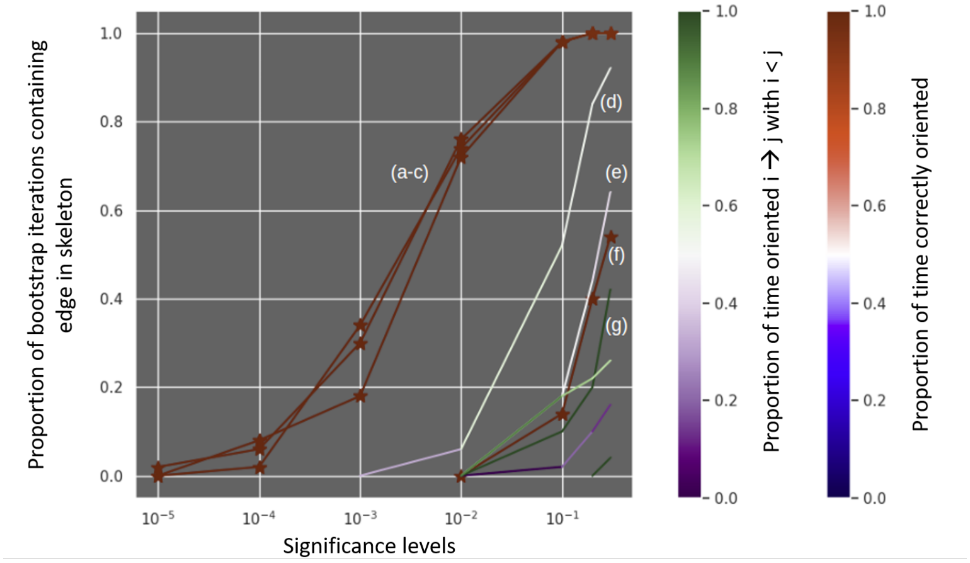
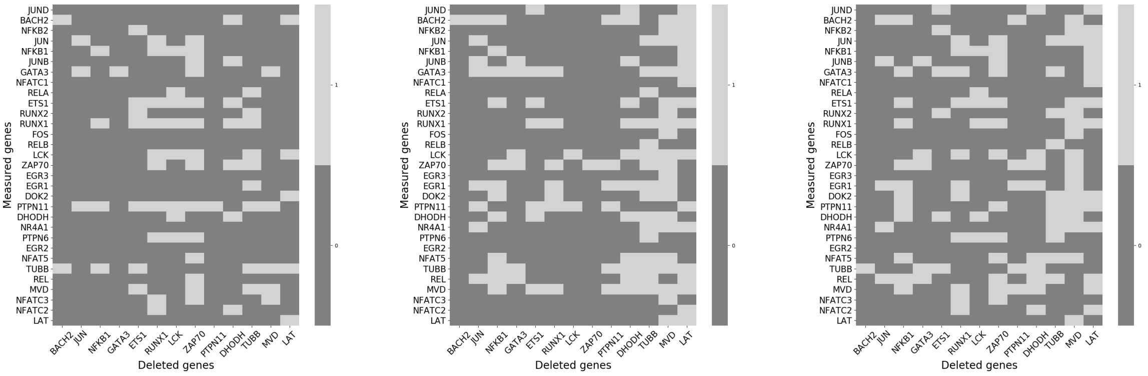


Figure S1: Stability selection paths for the difference between two simulated DAGs, with $p = 40$ nodes, $n = 2,000$ samples, and 2 additions/deletions between either DAG, where the skeleton of one of the DAGs was generated from an Erdös-Renyi model with 2 expected neighbors per node. Each curve represents an edge in the difference skeleton. The horizontal axis displays the significance level used for hypothesis tests in the difference skeleton discovery phase of the method, and the vertical axis displays the proportion of bootstrap iterations (out of 50) for which an edge was picked to be present. For the 4 edges which belong to the true difference skeleton (curves a-c and f), their color (varying from blue to red) indicates the probability that the edge was oriented in the correct direction, given that it was included in the graph. For the edges which do not belong to the true difference skeleton (such as curves d, e and g), their color (varying from purple to green) indicates the probability that the edge was oriented such that the node with the smaller index pointed to the larger one.



(a) (b) (c)

Figure S2: 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.

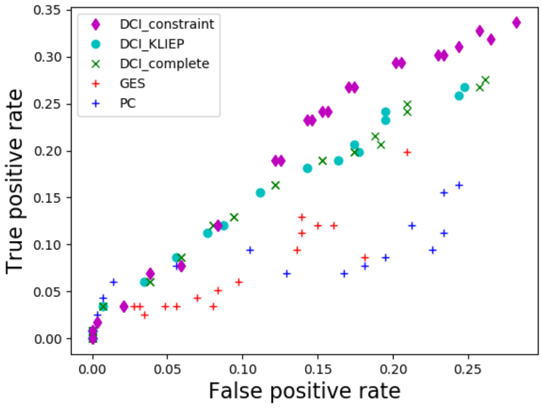
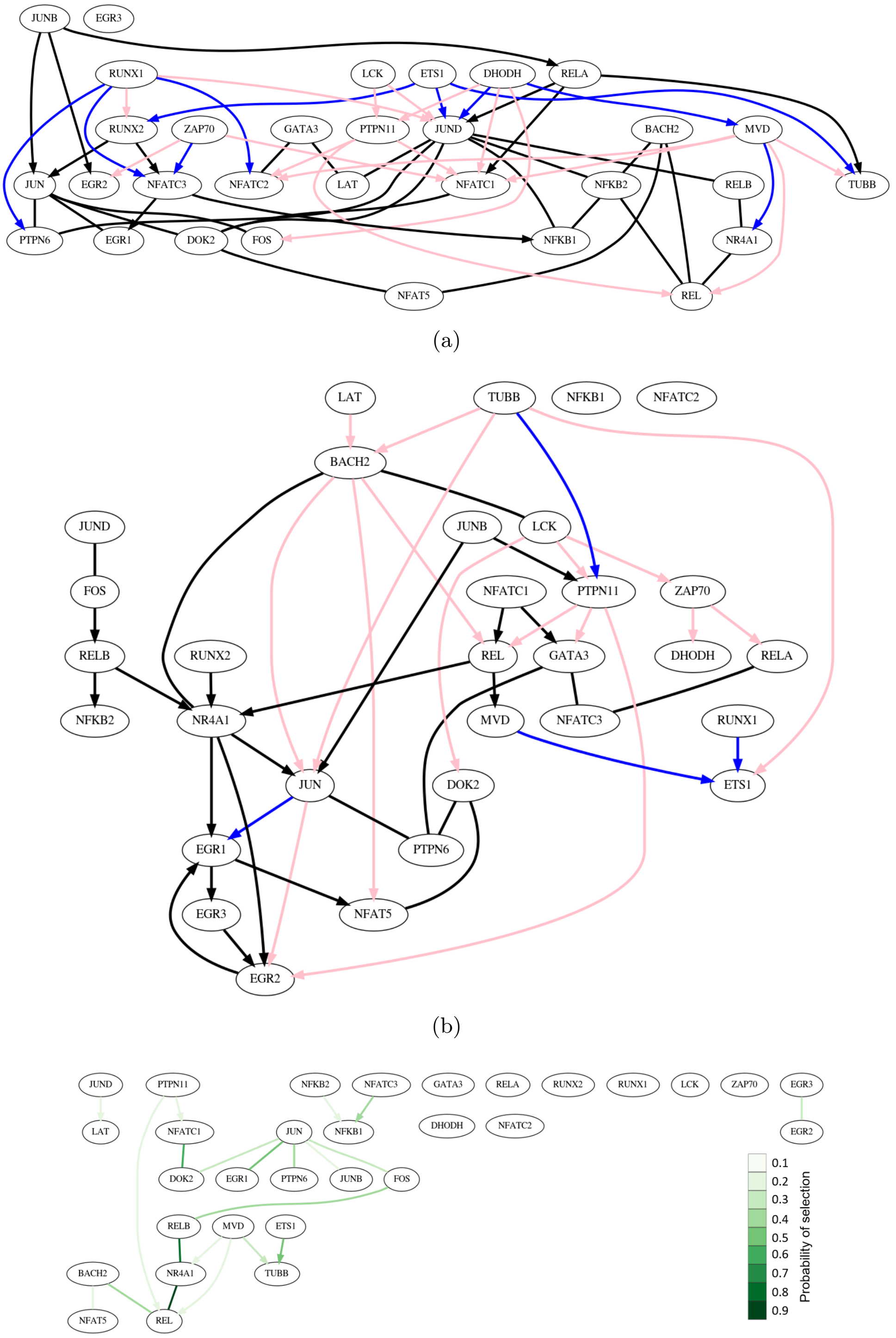


Figure S3: ROC plot evaluating DCI (initialized in the undirected difference graph estimated via the constraint-based method, KLIEP as well as in the complete graph), GES and PC on the CROP-seq data for predicting the differences in the effects of gene knockouts. Each point in the ROC curve represents a run with different tuning parameters for DCI, PC and GES.

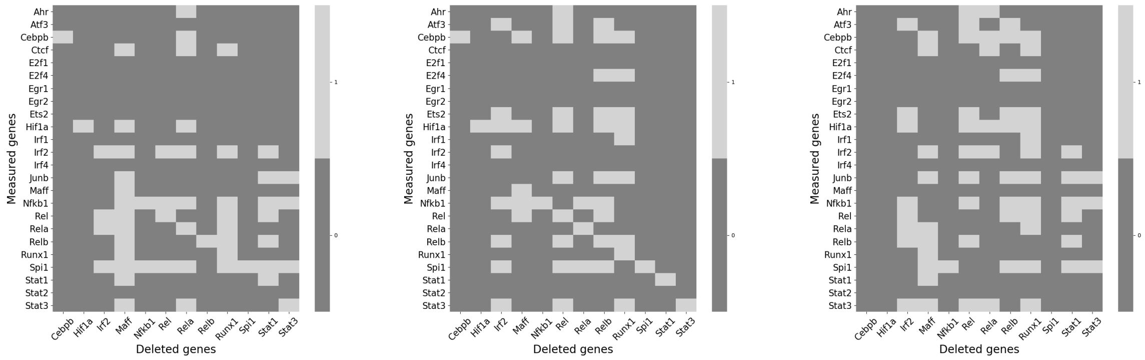


(a)

(b)

(c)

Figure S4: 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). (c) Difference gene regulatory network estimated using DCI with stability selection ($\pi_{\text{thr}} = 0.3$).



(a) (b) (c)

Figure S5: 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.

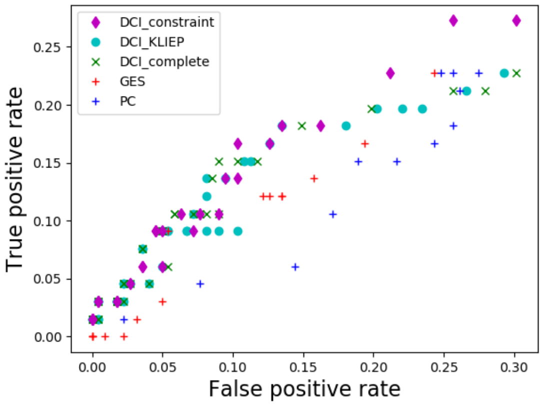


Figure S6: ROC plot evaluating DCI (initialized in the undirected difference graph estimated via the constraint-based method, KLIEP as well as in the complete graph), GES and PC on the Perturb-seq data for predicting the differences in the effects of gene knockouts. Each point in the ROC curve represents a run with different tuning parameters for DCI, PC and GES.

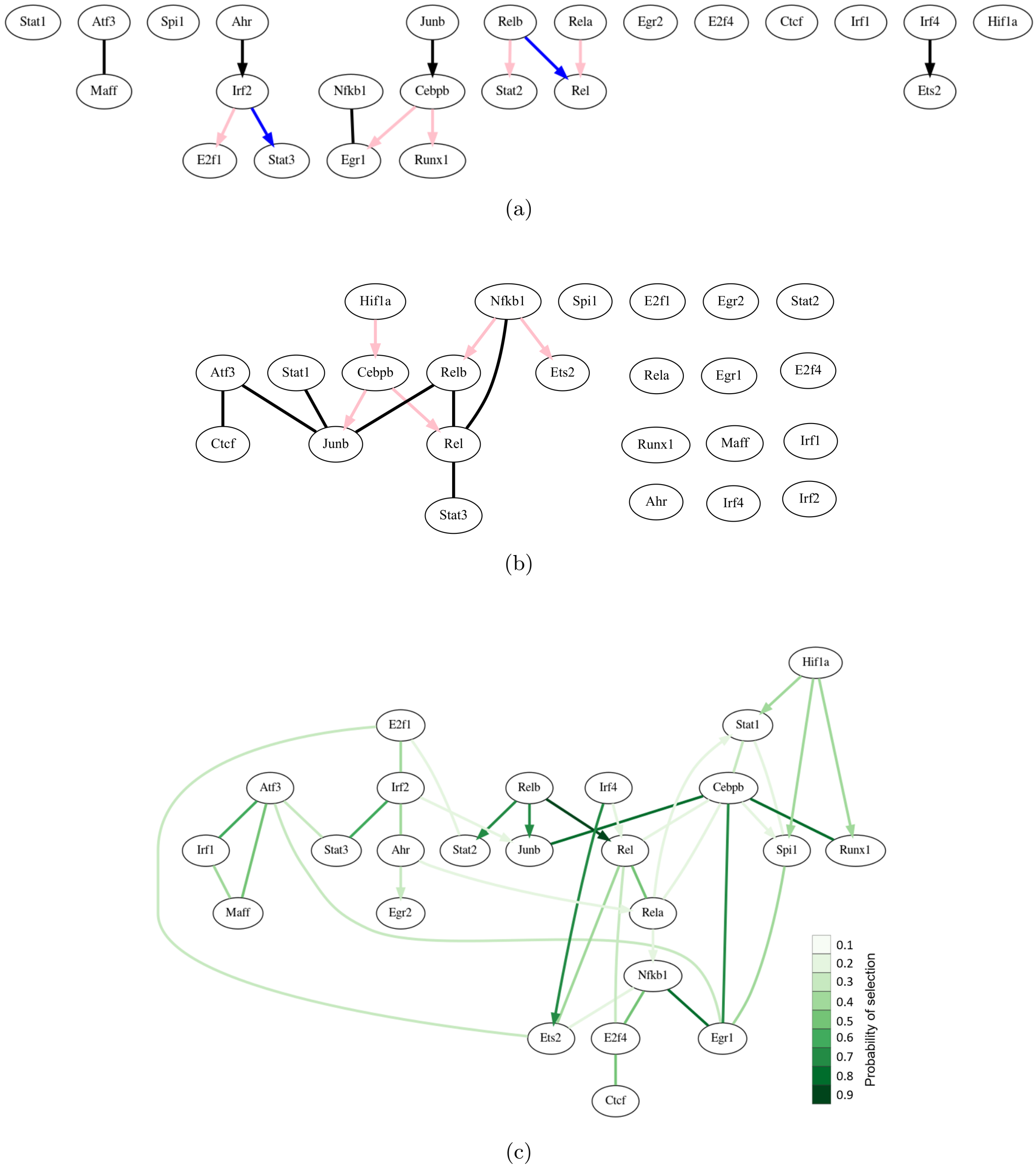


Figure S7: Examples of difference gene regulatory networks of dendritic cells before and after stimulation with LPS, estimated from the Perturb-seq data. Difference gene regulatory network inferred via (a) our algorithm, DCI, initialized with KLIEP, which directly learns the difference causal graph from 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). (c) Difference gene regulatory network estimated using DCI with stability selection ($\pi_{\text{thr}} = 0.3$).

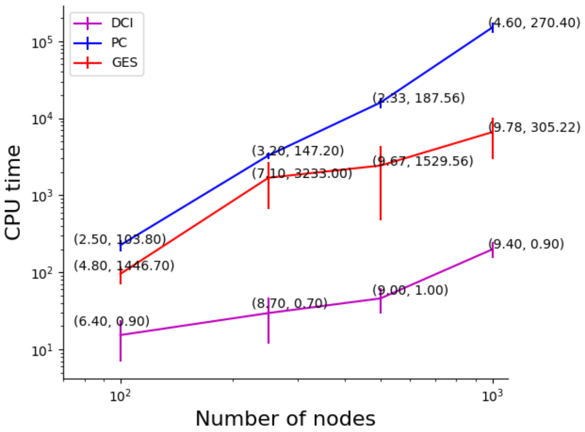
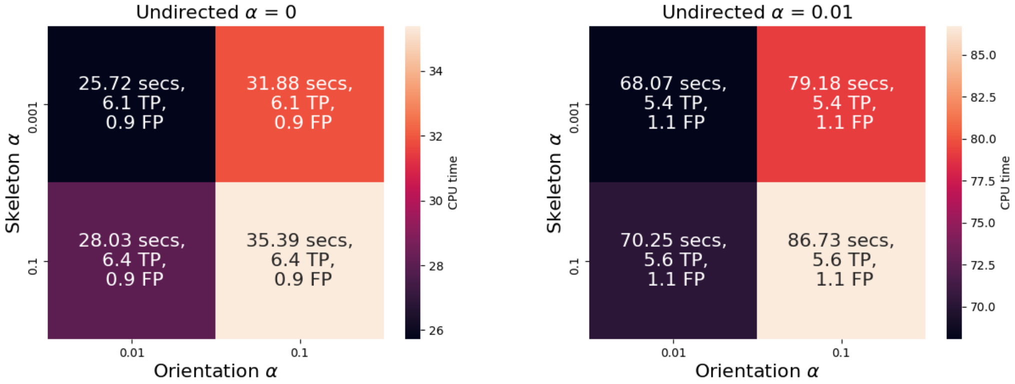


Figure S8: CPU time, in seconds, averaged over 10 simulations for variable input size. Each simulation consisted of $p \in \{100, 250, 500, 1000\}$ nodes, 10 expected neighbors, 10 changed edges between the graphs and 100,000 samples. DCI was run with $\alpha = 0$ for undirected graph estimation via the constraint-based method, $\alpha = 0.1$ for skeleton estimation, and $\alpha = 0.1$ for inferring the edge orientations. PC was run with $\alpha = 1 \times 10^{-6}$ and GES was run with $\lambda = 1000$. For PC algorithm with $p = 1000$, the simulations were averaged over only 5 separate runs due to long run times. Each point is annotated with a tuple consisting of the average number of true and false positives.



(a) (b)

Figure S9: CPU time, in seconds, averaged over 10 simulations for varying tuning parameters (sparsity). Each simulation consisted of $p = 100$ nodes, 10 expected neighbors, 10 changed edges between the graphs and 100,000 samples. DCI was run with (a) $\alpha = 0$ and (b) $\alpha = 0.01$ for undirected graph estimation via the constraint-based method, $\alpha \in \{0.001, 0.1\}$ for skeleton estimation, and $\alpha \in \{0.01, 0.1\}$ for inferring the edge orientations. Each square is annotated with the average CPU time as well as the number of true and 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. Lütkepohl, H. *New Introduction to Multiple Time Series Analysis* (Springer Science & Business Media, 2005).
8. Spirtes, P., Glymour, C. N. & Scheines, R. *Causation, Prediction, and Search* (MIT press, 2000).
9. Meek, C. *Graphical Models: Selecting Causal and Statistical Models* PhD thesis (Carnegie Mellon University, 1997).
10. Meinshausen, N. & Bühlmann, P. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **72,** 417–473 (2010).
11. 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).
12. 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.
13. Datlinger, P. *et al.* Pooled CRISPR screening with single-cell transcriptome readout. *Nature Methods* **14,** 297–301 (2017).
14. Dixit, A. *et al.* Perturb-Seq: dissecting molecular circuits with scalable single-cell RNA profiling of pooled genetic screens. *Cell* **167,** 1853–1866 (2016).
15. Yang, K., Katcoff, A. & Uhler, C. *Characterizing and Learning Equivalence Classes of Causal DAGs under Interventions* in *International Conference on Machine Learning* (2018), 5541–5550.
16. Saeed, B., Belyaeva, A., Wang, Y. & Uhler, C. *Anchored Causal Inference in the Presence of Measurement Error* in *Conference on Uncertainty in Artificial Intelligence* (2020), 619–628.