# Ensemble Inference and Inferability of Gene Regulatory Networks

**S. M. Minhaz Ud-Dean, Rudiyanto Gunawan***

Institute for Chemical and Bioengineering, ETH Zurich, Zurich, Switzerland

## Abstract

The inference of gene regulatory network (GRN) from gene expression data is an unsolved problem of great importance. This inference has been stated, though not proven, to be underdetermined implying that there could be many equivalent (indistinguishable) solutions. Motivated by this fundamental limitation, we have developed new framework and algorithm, called TRaCE, for the ensemble inference of GRNs. The ensemble corresponds to the inherent uncertainty associated with discriminating direct and indirect gene regulations from steady-state data of gene knock-out (KO) experiments. We applied TRaCE to analyze the inferability of random GRNs and the GRNs of *E. coli* and yeast from single- and double-gene KO experiments. The results showed that, with the exception of networks with very few edges, GRNs are typically not inferable even when the data are ideal (unbiased and noise-free). Finally, we compared the performance of TRaCE with top performing methods of DREAM4 *in silico* network inference challenge.

**Citation:** Ud-Dean SMM, Gunawan R (2014) Ensemble Inference and Inferability of Gene Regulatory Networks. PLoS ONE 9(8): e103812. doi:10.1371/journal.pone.0103812

**Editor:** Panayiotis V. Benos, University of Pittsburgh, United States of America

**Received** February 12, 2014; **Accepted** July 5, 2014; **Published** August 5, 2014

**Funding:** The work was supported by funding from the Swiss National Science Foundation (grant 137614, http://www.snf.ch). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Competing Interests:** The authors have declared that no competing interests exist.

* Email: rudi.gunawan@chem.ethz.ch

## Introduction

The discovery and analysis of biological networks have important applications, from finding treatment of diseases to engineering of microbes for the production of drugs and biofuels [1–4]. With continued advances in high throughput and omics technology, the inference of biological networks from omics data has received a great deal of interest. In particular, the inference of gene regulatory networks from gene expression data constitutes a major research topic in systems biology. In the last decade, the number of methodologies that are dedicated for the GRN inference has increased tremendously [5–7].

The Dialogue for Reverse Engineering Assessments and Methods (DREAM) project is a community-wide effort initiated to fulfill the need for a rigorous and fair comparison of the strengths and weaknesses of methods for the reverse engineering of biological networks from data. To this end, challenges involving the inference of cellular networks are organized on a yearly basis (http://www.the-dream-project.org/challenges). Specifically, the inference of GRN has become a major focus of several DREAM challenges. The outcomes of these challenges indicate that the state-of-the-art algorithms for GRN inference are unable to provide accurate and reliable network predictions, even when large expression datasets are available and the number of genes is small (10–100 genes) [8–10]. Nonetheless, a crowd-sourcing strategy that combines the predictions of different inference methods has been shown to be more reliable than any individual method [10].

Whether or not a direct regulation of one gene by another can be correctly inferred depends not only on the ability of an inference method to extract the relevant information from data, but also on the availability of such information in the data. In general, the information content of data is determined first and foremost by the conditions of the experiments. If the required information is unavailable or incompletely available, then the inference problem is underdetermined. In such a case, the network is not inferable from the data regardless of the method used.

The underdetermined nature is not exclusive to the inference of GRNs. Much of the difficulty in the inverse modeling of signaling and metabolic networks can also be attributed to the lack of inferability or identifiability of model structure and parameters [11–14]. As the inference problem is underdetermined, there exist multiple solutions which are indistinguishable. The lack of model identifiability has motivated a paradigm shift toward ensemble modeling [15–18]. While such a strategy has begun to gain traction in the modeling of signaling and metabolic networks, the ensemble paradigm has not been widely used in the inference of GRNs. Also, since the network representation and data for GRNs differ markedly from those for signaling and metabolic networks, existing algorithms for ensemble modeling cannot be directly applied for the inference of GRN.

In this work, we introduce new framework and algorithms, called **T**ransitive **R**eduction **a**nd **C**losure **E**nsemble (TRaCE), for the ensemble inference of GRNs. Specifically, TRaCE produces the lower and upper bounds of the ensemble, i.e. the smallest network and the largest network that limit the complexity of networks in the ensemble. As the size of the ensemble reflects the uncertainty about the GRN inference, the bounds can also be used to analyze the inferability of GRNs. In this study, we have used TRaCE in two applications. First, we investigated the inferability of random GRNs and the GRNs of *E. coli* and *S. cereviseae* given steady-state gene expression data of single- and double-gene KO experiments. Then, we applied TRaCE to simulated gene expression data, generated in the same manner as the DREAM 4 *in silico* network inference challenge, and compared the performance of TRaCE with existing methods.

## Methods

### Theoretical Foundation

**Definitions.** Here, we provide a short synopsis of basic concepts in graph theory that are necessary for the development of our algorithm. A *graph* $G$ is an ordered pair ($V(G)$, $E(G)$), where $V(G)$ is the set of vertices (or nodes) and $E(G)$ is the set of edges. The number of vertices $n$ and the number of edges $m$ are called the order and the size of the graph, respectively. An edge $e$ is defined by the pair $(i,j) \in E(G) \subset V(G) \times V(G)$, describing the existence of a relationship between two vertices $i$ and $j$. In this case, the edge $e=(i,j)$ is said to be *incident* to the vertices $i$ and $j$. The set of edges of a graph $G$ that are incident to a vertex $i$ is denoted by $E_G(i)$, while the cardinality of $E_G(i)$ is called the degree of the vertex $i$. Similarly, the set of edges that are incident to a set of vertices $Z \subset V(G)$ is denoted by $E_G(Z) = \bigcup_{i \in Z} E_G(i)$. A graph $A$ is a *subgraph* of $B$, denoted by $A \subset B$, if $V(A) \subset V(B)$ and $E(A) \subset E(B)$. In this case, $B$ is called the *supergraph* of $A$, and is also said to contain $A$. Furthermore, the union of two graphs $A$ and $B$ is denoted by $A \bigcup B$, where $E(A \bigcup B) = E(A) \bigcup E(B)$ and $V(A \bigcup B) = V(A) \bigcup V(B)$. The intersection of two graphs $A$ and $B$ is denoted by $A \bigcap B$, where $E(A \bigcap B) = E(A) \bigcap E(B)$ and $V(A \bigcap B) = V(A) \bigcap V(B)$. Finally, the difference between two graphs $A$ and $B$ with $V(A) = V(B)$ is denoted by $A - B$ and defined as the set of edges in the graph $A$ that do not belong to the graph $B$, i.e. edges in the set difference $E(A) - E(B)$.

A *directed edge* is an ordered pair $(i,j)$, representing an edge from the vertex $i$ pointing to the vertex $j$. A *directed graph* or *digraph* $G$ is a graph in which all of its edges are directed. A *directed path* is a sequence of vertices such that there exists a directed edge from one vertex to the next vertex in the graph. The first vertex in a directed path is called the *start vertex*, and the last is called the *end vertex*. A *directed cycle* is a directed path where the start and the end vertices are the same. A *directed acyclic graph* (DAG) is a digraph which does not contain any cycle. The *adjacency matrix* of a digraph $G$ of order $n$, denoted by $Adj(G)$, is an $n \times n$ matrix with $Adj_{i,j} = 1$ when $(i,j) \in E(G)$, and $Adj_{i,j} = 0$ otherwise. In other words, the non-zero elements of the adjacency matrix represent all directed edges from any node $i$ to another node $j$ in the graph $G$. Meanwhile, the *accessibility matrix* of $G$, denoted by $Acc(G)$, is an $n \times n$ matrix with $Acc_{i,j} = 1$ when there exists a directed path from node $i$ to node $j$, and $Acc_{i,j} = 0$ otherwise. When $Acc_{i,j} = 1$, vertex $j$ is said to be *accessible* from the vertex $i$.

A strongly connected component or strong component of a digraph $G$ is a maximal subset of nodes in $G$ where any two nodes in the subset are mutually accessible. Every pair of nodes that are part of a directed cycle belong to the same strong component, while any node that is not part of a cycle is a strong component of its own. The *condensation* of a digraph $G$ is the DAG of the strong components of $G$, which is generated by lumping the nodes belonging to a cycle into a single node and replicating the edges that are incident to any of these nodes onto the lumped node [19].

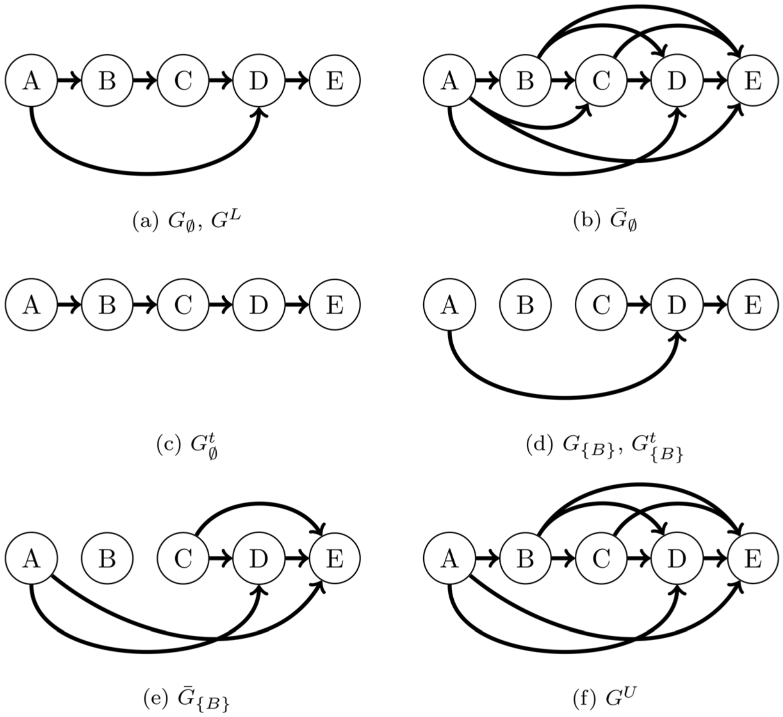
A digraph is *transitive* if for every pair of vertices $i$ and $j$, there exists an edge $(i,j)$ when there is a directed path from $i$ to $j$. The *transitive closure* of a digraph $G$, denoted by $\bar{G}$, is the smallest transitive supergraph of $G$ (with the fewest edges) [20]. When $G$ is a DAG, we denote the transitive closure of $G$ as $G^T$. As shown in Fig. 1(a)-(b), the transitive closure of a digraph can be generated by adding a directed edge $(i,j)$, whenever a directed path exists from vertex $i$ to vertex $j$. Note that the accessibility matrix of a digraph is the adjacency matrix of its transitive closure, i.e. $Acc(G) = Adj(\bar{G})$. For a digraph $G$, the set of digraphs that have the same transitive closure $\bar{G}$ is denoted by $S(G) = \{G_1 : \bar{G}_1 = \bar{G}\}$. The *transitive reduction* of $G$, denoted by $G^t$, is defined as the smallest member of $S(G)$ in size (i.e. the graph with the fewest edges). The transitive reduction of a DAG is unique, given by $G^t = \bigcap_{G_i \in S(G)} G_i$ [20]. An algorithm for obtaining transitive reduction has been previously developed [19], in which any directed edge $(i,j)$ is pruned whenever there exists a directed path from $i$ to $j$ that does not include $(i,j)$ (for example, see Fig. 1(c)). Note that the transitive reduction of a digraph with cycles is not unique.

**Inference of Network Ensemble Bounds.** In this work, we consider the inference of GRNs as digraphs, where the nodes correspond to the genes and the directed edges represent the gene regulatory interactions. An edge $i \rightarrow j$ implies that the expression of gene $i$ influences the expression of gene $j$. In the following, the GRN of interest is denoted by the digraph $G_\emptyset$. For any set of genes $V_{KO} \subset V(G_\emptyset)$, we use $G_{V_{KO}}$ to denote a subgraph of $G_\emptyset$ that results from removing all edges incident to the genes in the set $V_{KO}$, $E_{G_\emptyset}(V_{KO})$. In other words, $G_{V_{KO}}$ is the digraph with $V(G_{V_{KO}}) = V(G_\emptyset)$ and $E(G_{V_{KO}}) = E(G_\emptyset) - E_{G_\emptyset}(V_{KO})$. For example, $G_{\{B\}}$ associated with $G_\emptyset$ in Fig. 1(a) is the graph with all edges incident to gene $B$ removed, as shown in Fig. 1(d). Here, we interchangeably use the notations for a graph $G$ and its adjacency matrix $Adj(G)$.

Gene KO experiments are commonly performed for the purpose of GRN inference. In these experiments, the resulting data typically consist of gene expression profiles taken after the effects of the gene perturbation have reached steady state. While temporal gene expressions are increasingly measured, here we focus on using more commonly available steady-state expression data. The treatments of time-series measurements and observational data are left to future publications. Many network inference algorithms have been developed for using data of gene KOs [7,9], and most of these algorithms produce a single network prediction. In contrast, an ensemble inference strategy is adopted in this work.

In order to illustrate the limitation of using steady-state gene expression data for GRN inference, we consider a GRN $G_\emptyset$ described by the graph in Fig. 1(a). Here, KO of gene $A$ is expected to cause changes in the expression of genes $B$, $C$, $D$ and $E$ at steady state, even though $A$ directly regulates only $B$ and $D$. This simple illustration demonstrates that we cannot in principle discriminate direct and indirect gene regulations from steady-state gene KO expression data [19]. In general, genes that are differentially expressed upon knocking out gene $i$ in the GRN correspond to those that are directly and indirectly regulated by gene $i$, i.e. vertices in $G_\emptyset$ that are accessible from the vertex $i$. Motivated by such a limitation, in TRaCE we first convert gene KO data into gene accessibility lists or matrices. As the minimum input, TRaCE requires the complete dataset of single-gene KO experiments, from which one can construct the accessibility matrix $\bar{G}_\emptyset$. More specifically, the $j$-th element in the $i$-th row of $\bar{G}_\emptyset$ (i.e. $(\bar{G}_\emptyset)_{i,j}$) is set to 1 when gene $j$ is differentially expressed in the KO experiment of gene $i$. The other elements of $\bar{G}_\emptyset$ are set to 0. The detailed procedure of differential expression analysis adopted in this work is described in the Numerical Implementation section.

For data of multi-gene KO experiments, we consider the accessibility matrix of $G_{V_{KO}}$ for an appropriately chosen set of genes $V_{KO}$. In principle, we can determine $\bar{G}_{V_{KO}}$ from the complete set of experiments involving KOs of the genes in the set



(a) $G_\emptyset$, $G^L$

(b) $\bar{G}_\emptyset$

(c) $G^t_\emptyset$

(d) $G_{\{B\}}$, $G^t_{\{B\}}$

(e) $\bar{G}_{\{B\}}$

(f) $G^U$

**Figure 1. (a) An example of a directed graph $G_\emptyset$.** (b) The transitive closure $\bar{G}_\emptyset$ (in this case, $\bar{G}_\emptyset = G_\emptyset^T$ since $G_\emptyset$ is a DAG). (c) The transitive reduction $G_\emptyset^t$ of $\bar{G}_\emptyset$. (d) The directed graph $G_{\{B\}}$ associated with $G_\emptyset$. (e) The transitive closure $\bar{G}_{\{B\}}$. In this case, the transitive reduction $G^t_{\{B\}}$ happens to be the graph $G_{\{B\}}$. (f) The ensemble upper bound $G^U$ obtained from $\bar{G}_\emptyset$ and $\bar{G}_{\{B\}}$. The ensemble lower bound $G^L$ obtained from $G^t_\emptyset$ and $G^t_{\{B\}}$ happens to be the graph $G_\emptyset$.
doi:10.1371/journal.pone.0103812.g001

$V_{KO}$ and an additional gene $i$ for all $i \in \{V(G_\emptyset) - V_{KO}\}$. These experiments are equivalent to performing single-gene KOs of the GRN $G_{V_{KO}}$, and therefore $\bar{G}_{V_{KO}}$ can be obtained by following the same procedure as that for $\bar{G}_\emptyset$ above. As an illustration, consider the GRN in Fig. 1(a) with $V_{KO} = \{B\}$. The graph $G_{\{B\}}$ is given in Fig. 1(d). In this case, we can construct the accessibility matrix $\bar{G}_{\{B\}}$ from the data of two-gene KO experiments, namely $\{A,B\}$, $\{C,B\}$, $\{D,B\}$ and $\{E,B\}$ KOs. As these experiments differ from each other in only one gene while sharing the KO of gene $B$, the differential expression analysis of the data thus correspond to changes in the expression of the GRN $G_{\{B\}}$ caused by a single-gene KO. Consequently, in this analysis, genes that are found to be differentially expressed in the KO of $\{i,B\}$ are those that are accessible from gene $i$ ($i \in \{A,C,D,E\}$) in the graph $G_{\{B\}}$. For example, the KO of $\{A,B\}$ is expected to cause differential expression in genes $D$ and $E$. The full accessibility matrix of $G_{\{B\}}$ is illustrated by the digraph in Fig. 1(e).

We can generalize the simple example above to any set of genes $V_{KO}$ that could be derived from the available multi-gene KO experiments. More specifically, we set $(\bar{G}_{V_{KO}})_{i,j}$ to 1 when knocking-out $\{i, V_{KO}\}$ leads to a differential expression of gene $j$ with respect to its expression level in $G_{V_{KO}}$. The remaining elements of $\bar{G}_{V_{KO}}$ are set to 0. Unfortunately, the construction of $\bar{G}_{V_{KO}}$ of a large GRN $G_\emptyset$ would proportionally require a high number of KO experiments (the number of KO experiments is $n_{exp} = n - n_{V_{KO}}$, where $n$ and $n_{V_{KO}}$ is the number of genes in $G_\emptyset$ and $V_{KO}$, respectively). However, when $G_\emptyset$ is sparse, $\bar{G}_{V_{KO}}$ differs from $\bar{G}_\emptyset$ for only a few elements and importantly, these elements can be determined from $\bar{G}_\emptyset$ (see the next section).

In the theoretical development below, we assume that the accessibility matrices $\bar{G}_\emptyset$ and $\bar{G}_{V^k_{KO}}$ for $k = 1,2,\ldots,K$, have already been obtained from the expression data. Here, $K$ denotes the total number of accessibility matrices involving subgraphs of the GRN $G_\emptyset$ that can be constructed from data. For example, from the dataset of the complete double-gene KO experiments, we can obtain the accessibility matrix $\bar{G}_{V^k_{KO}} = \bar{G}_{\{k\}}$ for $k = 1,2,\ldots,n$ (here, $K = n$). In TRaCE, we consider the ensemble containing all digraphs that are consistent with the accessibility matrices $\bar{G}_\emptyset$ and $\bar{G}_{V^k_{KO}}$'s, which is the set:

$$\mathcal{A} = \left\{ G^* : \bar{G}^* = \bar{G}_\emptyset \text{ and } \overline{G}^*_{V^k_{KO}} = \bar{G}_{V^k_{KO}}, k = 1,2,\ldots,K \right\} \quad (1)$$

where $G^*_{V^k_{KO}}$ is the digraph with $V(G^*_{V^k_{KO}}) = V(G^*)$ and $E(G^*_{V^k_{KO}}) = E(G^*) - E_{G^*}(V^k_{KO})$. Note that the GRN $G_\emptyset$ is a member of the ensemble $\mathcal{A}$. The size of the ensemble is a direct measure of uncertainty in the network inference problem. A GRN is therefore deemed inferable when the ensemble only contains a single (unique) network.

As the number of digraphs in the ensemble is often very large, in TRaCE we generate only the lower and upper bounds of the ensemble, denoted by $G^L$ and $G^U$, respectively. The bounds are defined such that each digraph in the ensemble is a supergraph of $G^L$ and a subgraph of $G^U$. For GRNs without any cycle (i.e.

DAGs), the lower and upper bound GRNs can be obtained from the accessibility matrices of $G_\emptyset$ and $G_{V_{KO}^k}$'s (i.e. $G_\emptyset^T$ and $G_{V_{KO}^k}^T$'s) and their transitive reductions (i.e. $G_\emptyset^t$ and $G_{V_{KO}^k}^t$'s), using the following equations (for details see the Numerical Implementation section):

$$G^L = G_\emptyset^t \cup \left( \bigcup_{k=1}^{K} G_{V_{KO}^k}^t \right) \tag{2}$$

$$G^U = G_\emptyset^T \cap \left( \bigcap_{k=1}^{K} \left( G_{V_{KO}^k}^T \cup E_{G_\emptyset^T}(V_{KO}^k) \right) \right) \tag{3}$$

where $G_{V_{KO}^k}^T \cup E_{G_\emptyset^T}(V_{KO}^k)$ denotes the digraph with vertices $V(G_{V_{KO}^k})$ and edges $E(G_{V_{KO}^k}) \cup E_{G_\emptyset^T}(V_{KO}^k)$. Without any $G_{V_{KO}^k}^T$, the upper bound of the ensemble is simply given by the accessibility matrix $G_\emptyset^T$ and the lower bound is the transitive reduction $G_\emptyset^t$. As $G_{V_{KO}^k}$ is a subgraph of $G_\emptyset$, the transitive closure $G_{V_{KO}^k}^T$ is also a subgraph of $G_\emptyset^T$. In Eq. (3), the upper bound is constructed starting from $G_\emptyset^T$ in which edges are removed based on $G_{V_{KO}^k}^T$. Here, edges incident to $V_{KO}^k$ are not altered during the intersection of the accessibility matrix $G_{V_{KO}^k}^T$. For example, consider again the GRN in Fig. 1(a) with the accessibility matrices $G_\emptyset^T$ and $G_{\{B\}}^T$ in Figs. 1(b) and 1(e). The resulting upper bound $G^U$ from the combination of these accessibility matrices will have one fewer edge than $G_\emptyset^T$, which is the edge $(A,C)$ (see Fig. 1(f)). Thus, the size of the upper bound is generally reduced with the incorporation of $G_{V_{KO}^k}^T$'s. On the other hand, according to Eq. (2), the lower bound becomes larger with the inclusion of every available $G_{V_{KO}^k}^t$. In the same example above, the transitive reduction of $G_{\{B\}}^T$ happens to be the graph $G_{\{B\}}$ (i.e. in this case $G_{\{B\}}^t = G_{\{B\}}$). Here, the combination of $G_\emptyset^t$ and $G_{\{B\}}^t$ in Figs. 1(c) and 1(d), respectively, gives the lower bound $G^L$ that is equal to $G_\emptyset$. However, in general, $G^L$ is a subgraph of $G_\emptyset$.

Theorem 1 establishes $G^L$ and $G^U$ in Eqs. (2)-(3) as valid lower and upper bounds of the set $\mathcal{A}$ for DAGs.

**Theorem 1:** For $G^L$ and $G^U$ described in Eqs. (2)-(3), the following relationship applies:

$$G^L \subseteq G \subseteq G^U \ \forall \ G \in \mathcal{A}$$

*Proof of* $G^L \subseteq G \ \forall \ G \in \mathcal{A}$: For any edge $e \in E(G^L)$, Eq. (2) implies that either $e \in E\left(G_\emptyset^t\right)$ or $e \in E\left(G_{V_{KO}^k}^t\right)$. Therefore, we have either:

- $e \in E\left(G_\emptyset^t\right) = E(G^t) \subseteq E(G) \ \forall \ G \in \mathcal{A}$, or
- $e \in E\left(G_{V_{KO}^k}^t\right) = E((G - E_G(V_{KO}^k))^t) \subseteq E(G) \ \forall \ G \in \mathcal{A}$. □

*Proof of* $G \subseteq G^U \ \forall \ G \in \mathcal{A}$: If $e \in E(G)$ for some $G \in \mathcal{A}$, then $e \in E(G^T) = E(G_\emptyset^T)$. In addition, this edge satisfies either:

$$e \in E(G - E_G(V_{KO}^k)) \subseteq E\left(\left(G - E_G(V_{KO}^k)\right)^T\right) = E\left(G_{V_{KO}^k}^T\right), \text{ or}$$

$$e \in E_G(V_{KO}^k) \subseteq E_{G^T}(V_{KO}^k) = E_{G_\emptyset^T}(V_{KO}^k).$$

Therefore, $e \in E(G^U)$. □

**Remark**: Since $G_\emptyset$ is a member of $\mathcal{A}$, $G^L$ and $G^U$ can also be thought as the lower and upper bounds of $G_\emptyset$, i.e. $G^L \subseteq G_\emptyset \subseteq G^U$. For DAGs, the members of the set $\mathcal{A}$ can be obtained by combinatorially adding edges in the set $(G^U - G^L)$ to $G^L$. Therefore, the dimension of $\mathcal{A}$ is equal to $2^N$ where $N$ is the difference between the number of edges in $G^U$ and $G^L$. Finally, Theorem 1 guarantees that when $G^L = G^U$, $G_\emptyset$ is fully identifiable, i.e. $G^L = G^U = G_\emptyset$.

For digraphs with cycles, the upper bound can still be constructed using Eq. 3 with $\bar{G}$ replacing $G^T$. In this more general case, the relationship $G_\emptyset \subset G^U$ in Theorem 1 is still valid. However, as mentioned earlier, the transitive reduction of digraphs with cycles is not unique. In a previous publication [19], Wagner proposed a procedure in which digraphs are first condensed into DAGs before constructing the transitive reduction [19]. Similarly, in TRaCE, each input accessibility matrix is first condensed and the transitive reduction algorithm is subsequently applied to the DAG of the strong components. Here, edges incident to the condensations of cycles are also removed. Afterwards, the transitive reduction graph is expanded, reversing the condensation step. Except for cycles involving two nodes, edges of any directed cycle cannot be uniquely prescribed and are therefore pruned. The above procedure for reducing digraphs with cycles is referred to as **Con**densation, **T**ransitive **R**eduction and **Ex**pansion (ConTREx). The ConTREx of an accessibility matrix $\bar{G}$, denoted by $\underline{G}$, may no longer be a valid transitive reduction (i.e. $\underline{G}^T$ may not necessarily be equal to $\bar{G}$). Nonetheless, the lower bound constructed using Eq. (2) with $\underline{G}$'s replacing $G^t$'s, satisfies $G^L \subset G_\emptyset$. The proof of this relationship is analogous to the one presented for Theorem 1. However, $G^L$ may not be a member of $\mathcal{A}$ (see Fig. S1). Finally, the enumeration of digraphs with cycles from $G^L$ and $G^U$ is more complicated than that for DAGs. The main difference is in the generation of all possible cycles among nodes belonging to a particular strong component, constrained by $G^L$ and $G^U$ (see an example in Fig. S2).

**Error Correction and Filter.** In practice, the accessibility matrices constructed from data contain errors. Some elements of the accessibility matrices maybe identified as 1 when they should be 0 (i.e. false positive, FP), and some maybe identified as 0 when they should be 1 (i.e. false negative, FN). These errors can affect the lower and upper bound constructed by Eqs. (2)-(3). We denote the erroneous lower and upper bound by $\tilde{G}^L$ and $\tilde{G}^U$, respectively. In this case, neither $\tilde{G}^L$ is guaranteed to be a subgraph of $G_\emptyset$, nor $\tilde{G}^U$ a supergraph of $G_\emptyset$ and $\tilde{G}^L$.

There are several types of errors affecting $\tilde{G}^L$ and $\tilde{G}^U$. In the first case (Type A error), an edge which is not present in $G_\emptyset$ ($e \notin E(G_\emptyset)$) erroneously appears in $\tilde{G}^L$ and $\tilde{G}^U$ ($e \in E(\tilde{G}^L \cap \tilde{G}^U)$). Or, an edge in $G_\emptyset$ ($e \in E(G_\emptyset)$) is missing from both $\tilde{G}^L$ and $\tilde{G}^U$ ($e \notin E(\tilde{G}^L \cup \tilde{G}^U)$). As such error affects both $\tilde{G}^L$ and $\tilde{G}^U$ in the same manner, this error is not detectable from $\tilde{G}^L$ and $\tilde{G}^U$. The second case (Type B error) involves either a FP in the accessibility

**Table 1.** Types of Errors in $\tilde{G}^L$ and $\tilde{G}^U$.

| | Error | | | |
|---|---|---|---|---|
| | Type A | | Type B | Type C |
| $Adj(G_\emptyset)_{i,j}$ | 0 | 1 | 0 or 1 | 0 or 1 |
| $Adj(\tilde{G}^U)_{i,j}$ | 1 | 0 | 1 | 0 |
| $Adj(\tilde{G}^L)_{i,j}$ | 1 | 0 | 0 | 1 |

doi:10.1371/journal.pone.0103812.t001

matrix or a FN in the ConTREx matrix. In this case, the resulting bounds are still consistent with each other and are still valid for $G_\emptyset$. However, the ensemble size and the network uncertainty increase due to this error. In the third case (Type C error), an edge erroneously appears only in $\tilde{G}^L$, or vice versa, an edge is errorneously missing only from $\tilde{G}^U$ ($e \in E(\tilde{G}^L) \cap E(\tilde{G}^U)^c$), where $c$ denote the complement of a set). Here, the bounds become inconsistent with each other (i.e., $\tilde{G}^L \not\subseteq \tilde{G}^U$). Thus, we refer such errors as *inconsistent edges*, which can be identified by searching for edges belonging to $\tilde{G}^L$ that are not in $\tilde{G}^U$ (i.e., from $\tilde{G}^L - \tilde{G}^U$). Table 1 illustrates the three types of errors mentioned above.

A closer scrutiny of Eqs. (2)-(3) reveals that errors from the input accessibility matrices are passed on and compounded in the bounds. For example, FN errors in $\bar{G}_\emptyset$ or any $\bar{G}_{V_{KO}^k}$ will end up in $\tilde{G}^U$, while FPs in $\underline{G}_\emptyset$ and any $\underline{G}_{V_{KO}^k}$ will also appear in $\tilde{G}^L$. In order to reduce the transmission of errors, we have developed a filter such that only a subset of edges of $\bar{G}_{V_{KO}^k}$ are used for the construction of $\tilde{G}^L$ and $\tilde{G}^U$. The filter is based on the concept of *testable edges*. Specifically, the testable edges of $\bar{G}_{V_{KO}^k}$ are any edge $(i,j) \in E(\bar{G}_\emptyset)$ with $i,j \notin V_{KO}^k$, such that there exists a directed path from $i$ to $j$ involving one or more genes in $V_{KO}^k$. As directed paths involving genes in $V_{KO}^k$ are disconnected in $G_{V_{KO}}$, we can potentially verify the existence of the testable edges from $\bar{G}_{V_{KO}^k}$ and $\underline{G}_{V_{KO}^k}$. For example, the existence of the edge $(A,D)$ in Fig. 1(a) can be verified using the transitive reduction of $\bar{G}_{\{B\}}$, which is the graph shown in Fig. 1(d). Meanwhile, we can establish the absence of the edge $(A,C)$ in $G_\emptyset$ using the accessibility matrix $\bar{G}_{\{B\}}$ (see Fig. 1(e)). The number of testable edges can also be used to estimate the information content of a $\bar{G}_{V_{KO}}$, where a higher number of testable edges indicates more informative $\bar{G}_{V_{KO}}$.

For a given $V_{KO}$, it is straightforward to show that the testable edges are the non-zero entries of the testability matrix:

$$T_{V_{KO}} = \sum_{i \in V_{KO}} (\bar{G}_\emptyset)^i \otimes (\bar{G}_\emptyset)_i, \tag{4}$$

where $(\bar{G}_\emptyset)^i$ and $(\bar{G}_\emptyset)_i$ are the $i$-th column and $i$-th row of $\bar{G}_\emptyset$, respectively, and $\otimes$ denotes the outer product. During the construction of the lower and upper bound GRNs, the incorporation of each $\bar{G}_{V_{KO}^k}$ will only need to be performed for the associated testable edges, i.e. non-zero elements of $T_{V_{KO}^k}$. Moreover, as the number of testable edges corresponding to $\bar{G}_{V_{KO}^k}$ is typically small and as such edges can be determined from $\bar{G}_\emptyset$, only a few rows of $\bar{G}_{V_{KO}^k}$ need to be determined from data, i.e. rows of $T_{V_{KO}^k}$ with non-zero entries. Thus, the number of KO experiments for constructing $\bar{G}_{V_{KO}^k}$ could be relaxed by considering only testable edges.

## Numerical Implementation

The pseudo-codes and the MATLAB implementations of TRaCE are provided in the supporting material and the following website (http://www.cabsel.ethz.ch/tools/trace). Given steady-state gene expression data, we first group the data into datasets according to the KO experiments required for the construction of the accessibility matrices. We perform differential expression analysis for each dataset using Z-score transformation and obtain the corresponding accessibility matrix. We provide two implementations of TRaCE, one with and another without error correction. TRaCE without error correction should only be applied when the input accessibility matrices are error free (e.g. for inferability analysis). In any other scenario, TRaCE with error correction should be used. If desired, a ranked list of gene regulatory predictions can also be generated using the lower and upper bounds of the ensemble and the differential expression analysis.

**Constructing Accessibility Matrices from Expression Data.** In the case studies, we have employed the Z-score transformation for differential gene expression analysis [21]. Without loss of generality, we describe below the procedure for constructing $\bar{G}_{V_{KO}}$ from the complete set of single-gene KOs of $G_{V_{KO}}$, i.e. all possible combinations of $\{i, V_{KO}\}$ genes KOs. The gene expression dataset is organized into a matrix in which the rows correspond to the experiments and the columns correspond to the genes. Technical replicates are arranged into separate data matrices. For microarray data, the gene expression is typically represented by log-10 transformed fluorescence intensity data. The following procedure is also illustrated in Fig. 2.

1. We first obtain the sample mean $\mu_j$ and standard deviation $s_j$ of the expression of each gene $j$ in the dataset. More specifically, for each technical replicate, we calculate the sample mean and standard deviation of the $j$-th column in the data matrix. Then, we identify expressions that differ from the mean by more than a specified multiple $z_{cutoff}$ of the standard deviation. We subsequently recompute the sample mean and standard deviation $\mu_j$ and $s_j$ by excluding the data beyond $z_{cutoff}$. When available, we also use the expression data from the KO experiment of genes $V_{KO}$ in calculating $\mu_j$ and $s_j$.
2. For each replicate, we compute a z-score matrix $z_{i,j}$ for $i,j \in V(G_\emptyset)$ according to [21]

$$z_{i,j} = \begin{cases} \frac{g_{i,j} - \mu_j}{s_j}, & \forall i,j \notin V_{KO} \\ 0 & \forall i,j \in V_{KO} \end{cases} \tag{5}$$

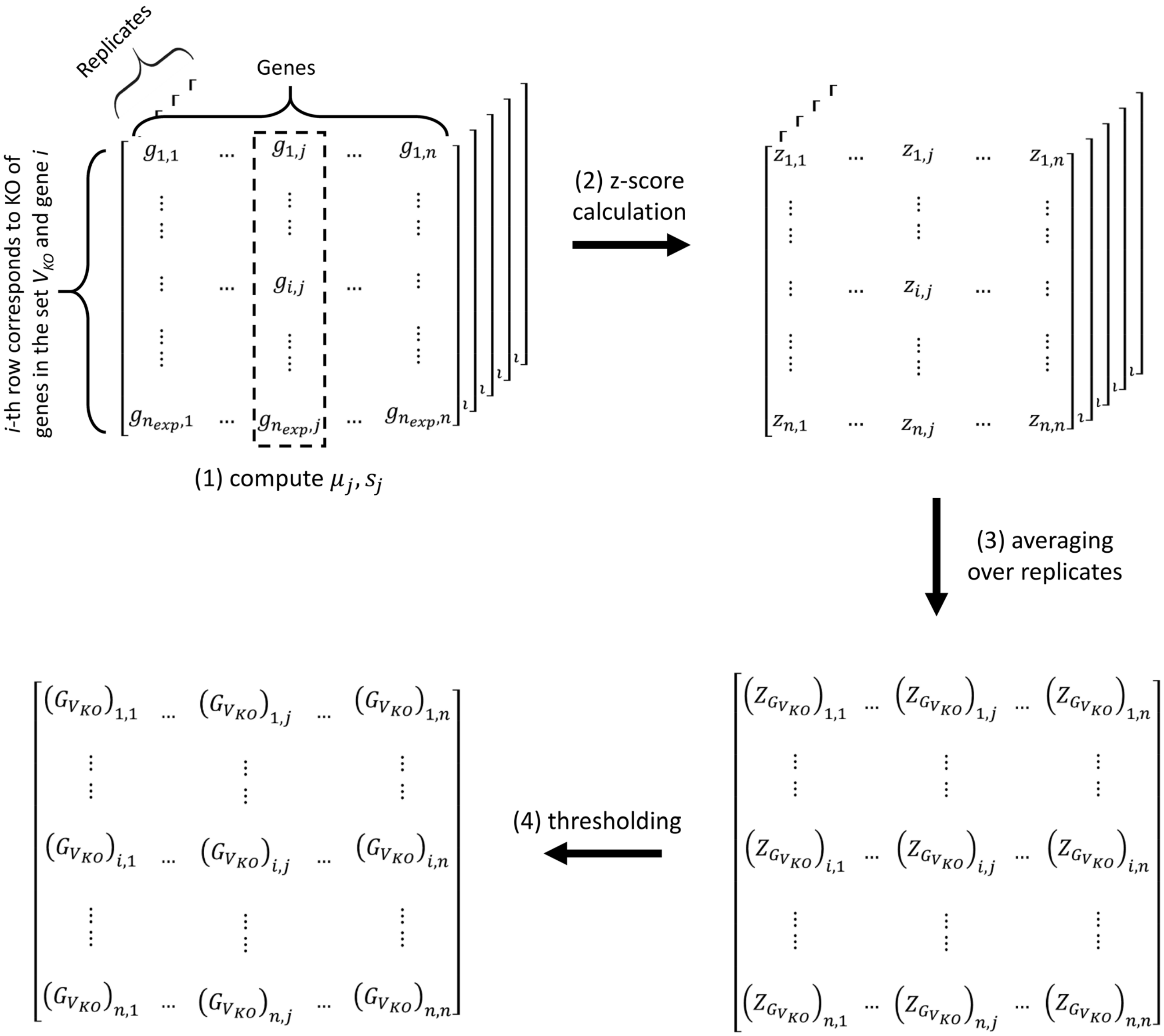

Replicates
Genes
$i$-th row corresponds to KO of genes in the set $V_{KO}$ and gene $i$
$g_{1,1}$ … $g_{1,j}$ … $g_{1,n}$
$g_{i,j}$
$g_{n_{exp},1}$ … $g_{n_{exp},j}$ … $g_{n_{exp},n}$
(1) compute $\mu_j, s_j$
(2) z-score calculation
$z_{1,1}$ … $z_{1,j}$ … $z_{1,n}$
$z_{i,j}$
$z_{n,1}$ … $z_{n,j}$ … $z_{n,n}$
(3) averaging over replicates
$(Z_{G_{V_{KO}}})_{1,1}$ … $(Z_{G_{V_{KO}}})_{1,j}$ … $(Z_{G_{V_{KO}}})_{1,n}$
$(Z_{G_{V_{KO}}})_{i,1}$ … $(Z_{G_{V_{KO}}})_{i,j}$ … $(Z_{G_{V_{KO}}})_{i,n}$
$(Z_{G_{V_{KO}}})_{n,1}$ … $(Z_{G_{V_{KO}}})_{n,j}$ … $(Z_{G_{V_{KO}}})_{n,n}$
(4) thresholding
$(G_{V_{KO}})_{1,1}$ … $(G_{V_{KO}})_{1,j}$ … $(G_{V_{KO}})_{1,n}$
$(G_{V_{KO}})_{i,1}$ … $(G_{V_{KO}})_{i,j}$ … $(G_{V_{KO}})_{i,n}$
$(G_{V_{KO}})_{n,1}$ … $(G_{V_{KO}})_{n,j}$ … $(G_{V_{KO}})_{n,n}$

**Figure 2. Construction of accessibility matrix $\bar{G}_{V_{KO}}$ from expression data.** The data come from KOs of genes in the set $V_{KO}$ and an additional gene $i$, $\forall i \in V(G_\emptyset) - V_{KO}$. For each replicate, the expression data are arranged into a matrix where the rows correspond to the experiments and the columns correspond to the genes. (1) The sample mean and standard deviation of the expression of gene $j$, denoted by $\mu_j$ and $s_j$, respectively, are obtained using the expression data in the $j$-th column of the data matrix. (2) For each replicate, a z-score matrix is computed according to Eq. (5). (3) Subsequently, the z-score matrices are averaged over the technical replicates to give $Z_{G_{V_{KO}}}$. (4) The accessibility matrix $\bar{G}_{V_{KO}}$ is determined from $Z_{G_{V_{KO}}}$ based on a threshold criterion in Eq. (6).

doi:10.1371/journal.pone.0103812.g002

where $\boldsymbol{g}_{i,j}$ is the expression level of gene $j$ associated with knocking out gene $i$ and genes in $V_{KO}$. These z-scores reflect the significance of changes in the gene expression with respect to the GRN $G_{V_{KO}}$.

3. Subsequently, we average the z-score matrices over the technical replicates, producing the overall z-score matrix $Z_{G_{V_{KO}}}$.
4. We determine the accessibility matrix $\bar{\boldsymbol{G}}_{V_{KO}}$ from $Z_{G_{V_{KO}}}$ using a threshold, as follows:

$$(\bar{G}_{V_{KO}})_{i,j} = \begin{cases} 1 & \text{if } (Z_{G_{V_{KO}}})_{i,j} > z_{threshold} \\ 0 & \text{if } (Z_{G_{V_{KO}}})_{i,j} \leq z_{threshold} \end{cases} \quad (6)$$

In our experience, $z_{cutoff} = 3$ and $z_{threshold} = 2$ provide reliable network ensembles. In general, choosing higher $z_{cutoff}$ and $z_{threshold}$ will lead to fewer FPs but more FNs in the accessibility matrix. For the GRN examples considered in this work, the performance of TRaCE does not vary considerably within the selected ranges of $z_{threshold}$ between 1.5 and 2.5 and $z_{cutoff}$ between 2 and 3 (see Results).

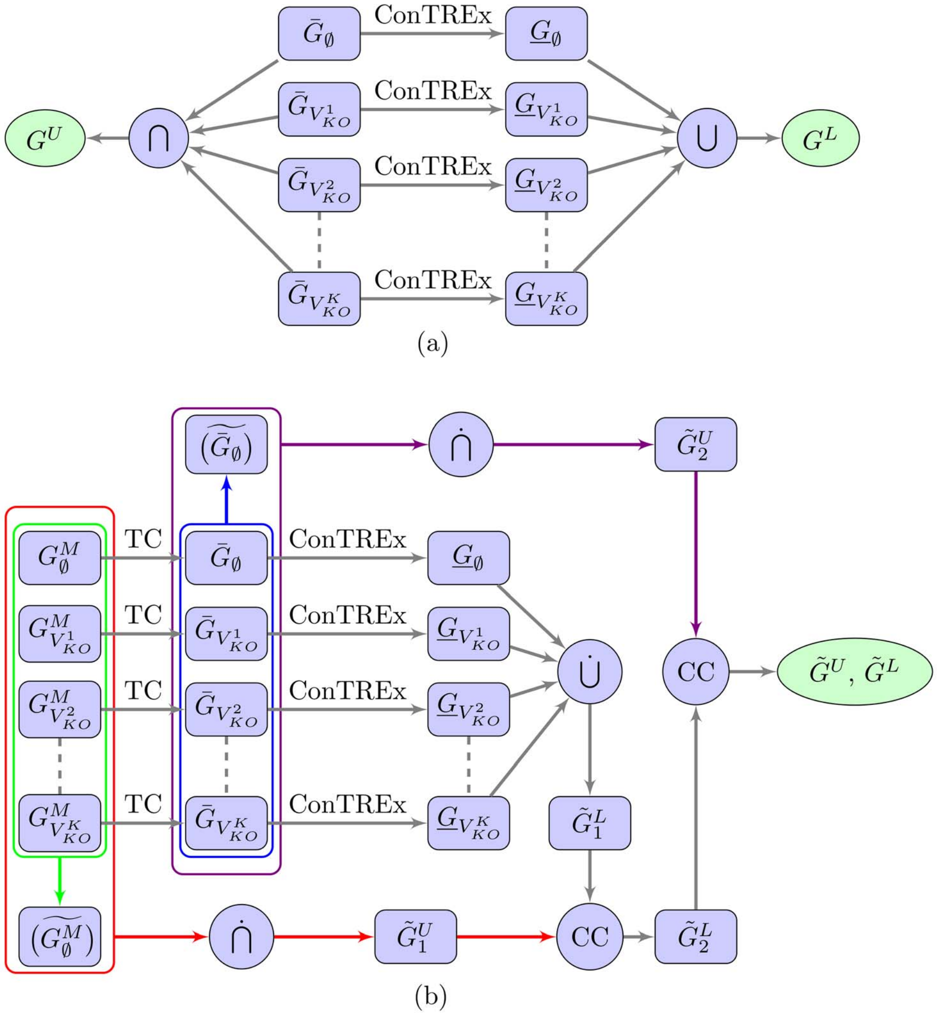


**Figure 3. Schematic diagrams of TRaCE with and without error correction.** (a) Construction of the lower bound $G^L$ and upper bound $G^U$ from $\bar{G}_\emptyset$ and $\bar{G}_{V_{KO}^k}$'s using TRaCE without error correction. Expression data from gene KO experiments are first converted into accessibility matrices. ConTREx is then applied to each accessibility matrix, removing feed-forward edges and edges incident to vertices belonging cycles with more than 2 nodes. The upper bound is constructed by taking the intersection of the accessibility matrices, while the lower bound is constructed by taking the union of the ConTREx outputs. (b) Construction of the lower bound $G^L$ and upper bound $G^U$ from $\bar{G}_\emptyset$ and $\bar{G}_{V_{KO}^k}$'s using TRaCE with error correction. Expression data from gene KO experiments are converted into accessibility matrices $G_\emptyset^M$ and $G_{V_{KO}^k}^M$'s, where the superscript $M$ indicates that these matrices may not be transitive due to noise in the measured gene expression levels. Subsequently, the transitive closures of $G_\emptyset^M$ and $G_{V_{KO}^k}^M$'s are created, denoted respectively by $\bar{G}_\emptyset$ and $\bar{G}_{V_{KO}^k}$'s, and the ConTREx of these closures are evaluated. TRaCE with error correction begins with the preprocessing of $G^M$'s and $\bar{G}$'s to produce the corrected matrices $\left(\widetilde{G_\emptyset^M}\right)$ and $(\widetilde{\bar{G}_\emptyset})$, which are required to determine testable edges. For the construction of the lower and upper bounds, the union and intersection of matrices are performed with filtering, denoted by $\dot{\bigcup}$ and $\dot{\bigcap}$, respectively, where only the relevant testable edges are updated. Two candidate upper bounds are obtained, the first from the matrices $G^M$'s, denoted by $\tilde{G}_1^U$, and the second from the matrices $\bar{G}$'s, denoted by $\tilde{G}_2^U$. Meanwhile, the initial lower bound estimate, denoted by $\tilde{G}_1^L$, is obtained from the ConTREx matrices. The consistency check (CC) is first applied to the pair $\tilde{G}_1^L$ and $\tilde{G}_1^U$ to produce the corrected lower bound $\tilde{G}_2^L$, and then to the pair $\tilde{G}_2^L$ and $\tilde{G}_2^U$ to produce the final estimates of the bounds $\tilde{G}^L$ and $\tilde{G}^U$. More detailed descriptions of the filtering and consistency check can be found in supporting material (text S1 and text S2).

doi:10.1371/journal.pone.0103812.g003

**TRaCE without error correction.** TRaCE without error correction is implemented as matrix-operations of Eqs. (2) and (3). Briefly, the upper bound $G^U$ is constructed by performing Hadamard (element wise) multiplications of the accessibility matrices, excluding the rows and columns corresponding to genes in $V_{KO}^k$. On the other hand, the transitive reduction is based on the algorithm by Wagner [19], which has been re-implemented using matrix operations. When there is no cycle in $\bar{G}_\emptyset$ and $\bar{G}_{V_{KO}^k}$'s, the transitive reduction algorithm is applied to each accessibility matrix and the construction of $G^L$ is done by binary additions of the transitive reductions, following Eq. (2). Cycles and genes involved in cycles can be detected from entries of $Acc(G_\emptyset) \times Acc(G_\emptyset)^T$ [22]. For GRNs with cycles, the ConTREx procedure is applied to each available accessibility matrix, and the resulting $\underline{G}$ matrices are again combined using binary

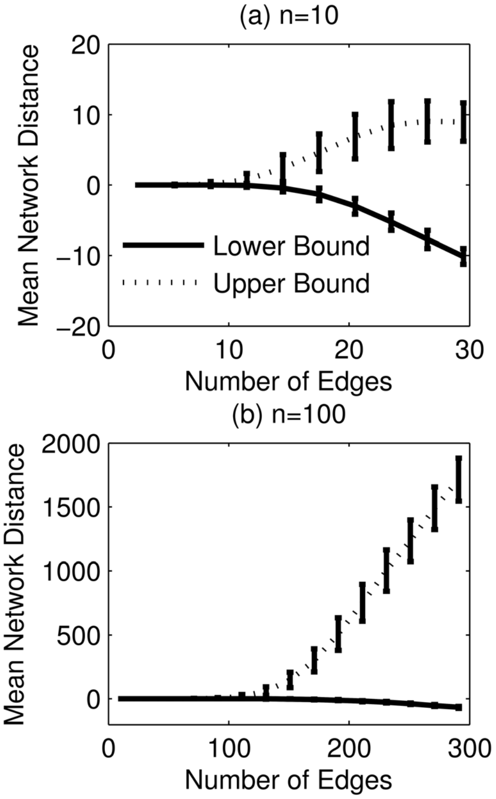


**Figure 4. Ensemble inference and inferability of 10,000 random networks of order (a) $n=10$ and (b) $n=100$ genes.** The mean network distances of the lower and upper bounds from $G_\emptyset$ are shown as a function of network size (i.e. number of edges). The error bars indicate the standard deviations.
doi:10.1371/journal.pone.0103812.g004

additions to produce $G^L$. The schematic diagram of the error-free implementation is shown in Fig. 3(a).

**TRaCE with error correction.** The procedure for TRaCE with error correction is illustrated in Fig. 3(b). There are two main steps in this procedure: (1) the construction of lower and upper bounds with filtering and (2) the correction of inconsistent edges. The first main step refers to an implementation of Eqs. (2) and (3) in which the intersection and union operations involving $G_{V_{KO}^k}$ are performed only for testable edges associated with non-zero entries of $T_{V_{KO}^k}$. As testable edges are determined from $\bar{G}_\emptyset$, a pre-processing step is performed to reduce errors in $\bar{G}_\emptyset$. The premise behind the pre-processing step is that an error unlikely affects the same edge, and that testable edges of any $\bar{G}_{V_{KO}^k}$ constitute only a small subset of edges in $\bar{G}_\emptyset$ (i.e. the network is sparse). Following this premise, edges that appear in a majority of the accessibility matrices (above a certain threshold) are kept, but are otherwise removed. In our experience, a threshold of 65% gives a good and reliable performance, but any value between 50% to 80% works quite well in the case studies (see Results). A more detailed description of the pre-processing method can be found in text S1, while the filtering algorithm is provided in text S2.

The schematic diagram of TRaCE with error correction is given in Fig. 3(b). We consider two sets of accessibility matrices; the first set comes from differential expression analysis (based on $Z_G$'s) and the second set comes from the transitive closure of the first set. We create the second set of matrices since the accessibility matrices identified from differential expressions may not satisfy the transitivity condition due to errors. The pre-processing step above is applied to both sets of matrices. Subsequently, two candidate upper bounds are generated using TRaCE with filtering. The upper bound obtained from the first set of accessibility matrices, denoted by $\tilde{G}_1^U$, is expectedly smaller (in size) than the bound from the second set, denoted by $\tilde{G}_2^U$. Note that ConTREx is only applicable to transitive digraphs, and therefore is applied only to the transitive closures (i.e. the second set). Using TRaCE with filtering, a candidate lower bound, denoted by $\tilde{G}_1^L$, is generated from the results of ConTREx.

The last step in the procedure is to correct inconsistent edges, which is done by voting. For each inconsistent edge, we compared the number of times that the edge is present in the accessibility matrices and the ConTREx results (supporting the presence of the edge), with the number of times that the edge is absent from the accessibility and ConTREx matrices (supporting the absence of the edge). The upper bound is corrected (by addition of this edge) when the presence of the edge receives a (simple) majority vote. Vice versa, the lower bound is corrected (by removal of this edge) when the absence of the edge receives a majority vote. In the case of no majority vote, the edge is added to the upper bound and removed from the lower bound. The detail of the consistency check is described in text S3. As shown in Fig. 3(b), the consistency check and correction are first performed for the pair $\tilde{G}_1^U$ and $\tilde{G}_1^L$, and subsequently the corrected lower bound, denoted by $\tilde{G}_2^L$, is compared with $\tilde{G}_2^U$ to obtain the final corrected $\tilde{G}^L$ and $\tilde{G}^U$.

**Ranking of Edges from Ensemble Bounds.** If desired, a ranked list of edges can be generated using the lower and upper bounds of TRaCE in conjunction with the average z-scores for $G_\emptyset$, i.e. $Z_{G_\emptyset}$. Here, we carry out the ranking of regulatory edges in two phases. In the first phase, we rank subsets of edges according to the lower and upper bounds in the following order: edges in $\Sigma_1 = \underline{G}^L$, edges in $\Sigma_2 = G^L - \ \underline{G}^L$ , edges in $\Sigma_3 = G^U - \ G^L$, edges in $\Sigma_4 = \bar{G}^U - \ G^U$, and finally edges in $\Sigma_5 = (\bar{G}^U)^c$. In the second phase, we rank the edges within individual subsets according to the average z-scores. We implement the second phase by first computing the overall scores $\bar{z}_{i,j}$ according to

$$\bar{z}_{i,j} = \begin{cases} \left(Z_{G_\emptyset}\right)_{i,j} + \max\limits_{(k,l)\in\Sigma_2} \bar{z}_{k,l} & \text{if } (i,j)\in\Sigma_1 \\ \left(Z_{G_\emptyset}\right)_{i,j} + \max\limits_{(k,l)\in\Sigma_3} \bar{z}_{k,l} & \text{if } (i,j)\in\Sigma_2 \\ \left(Z_{G_\emptyset}\right)_{i,j} + \max\limits_{(k,l)\in\Sigma_4} \bar{z}_{k,l} & \text{if } (i,j)\in\Sigma_3 \\ \left(Z_{G_\emptyset}\right)_{i,j} + \max\limits_{(k,l)\in\Sigma_5} \bar{z}_{k,l} & \text{if } (i,j)\in\Sigma_4 \\ \left(Z_{G_\emptyset}\right)_{i,j} & \text{if } (i,j)\in\Sigma_5 \end{cases} \tag{7}$$

Following the submission requirement of DREAM 4 network inference challenge, we then assign a confidence score $x_{i,j}$ to the edge $(i,j)$ according to:

$$x_{i,j} = \frac{\bar{z}_{i,j}}{\max_{i,j} \bar{z}_{i,j}} \tag{8}$$

A score $x_{i,j}$ of 1 reflects the highest confidence of the existence of an edge $(i,j)$, and vice versa a zero confidence score indicates certainty in the inexistence of $(i,j)$. Finally, the ranked list of edges is generated by sorting the edges in decreasing order of confidence

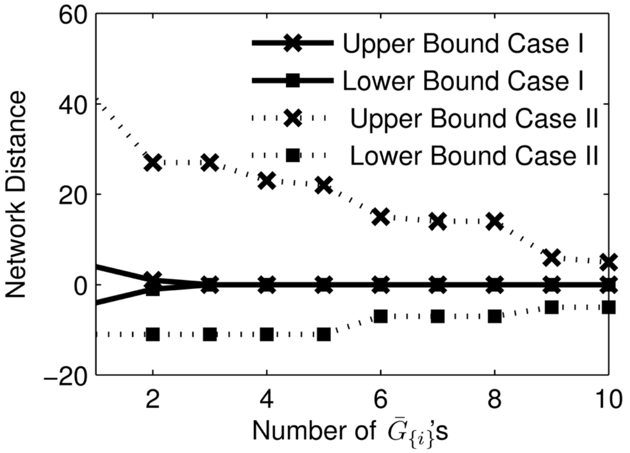


**Figure 5. Examples of the ensemble inference of random networks with 10 genes.** In case I, the GRN has 8 edges and is inferable from the accessibility matrices $\bar{G}_{\emptyset}$ and as few as three $\bar{G}_{\{i\}}$'s. In case II, the GRN has 13 edges and is not inferable.
doi:10.1371/journal.pone.0103812.g005

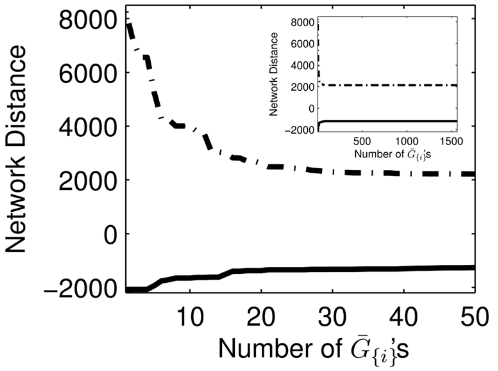


**Figure 7. Ensemble inference of *E. coli* GRN from error-free $\bar{G}_{\emptyset}$ and the complete set of $\bar{G}_{\{i\}}$'s.** The plot shows the network distances of the lower and upper bounds from $G_{\emptyset}$ as a function of the number of $\bar{G}_{\{i\}}$'s for the 50 most informative $\bar{G}_{\{i\}}$'s, i.e. the top 50 highest number of testable edges. The incorporation of $\bar{G}_{\{i\}}$'s and $\underline{G}_{\{i\}}$'s was performed sequentially in decreasing number of testable edges. The inset shows the result for the complete set of $\bar{G}_{\{i\}}$'s.
doi:10.1371/journal.pone.0103812.g007

scores. A similar procedure, called down-ranking, has been presented in Pinna *et al.* [23], where feed-forward edges are ranked lower than edges in the transitive reduction of the accessibility matrix. However, the down-ranking algorithm is described only for data of single-gene KO experiments.

## Results

### Inferability Analysis

We first applied TRaCE to error-free accessibility matrices of $G_{\emptyset}$ and $G_{V_{KO}^k}$ by assuming ideal data (unbiased and error free) for the purpose of inferability analysis. Such an analysis is analogous to *a priori* identifiability analysis in the kinetic modeling of biological networks [12]. Here, we evaluated the network distances between the lower and upper bounds and the GRN, i.e. the numbers of edges in the set $G_{\emptyset} - G^L$ and $G^U - G_{\emptyset}$, respectively.

**Random GRNs.** We investigated the inferability of random GRNs of orders $n = 10$ and $n = 100$ genes. We set the network size

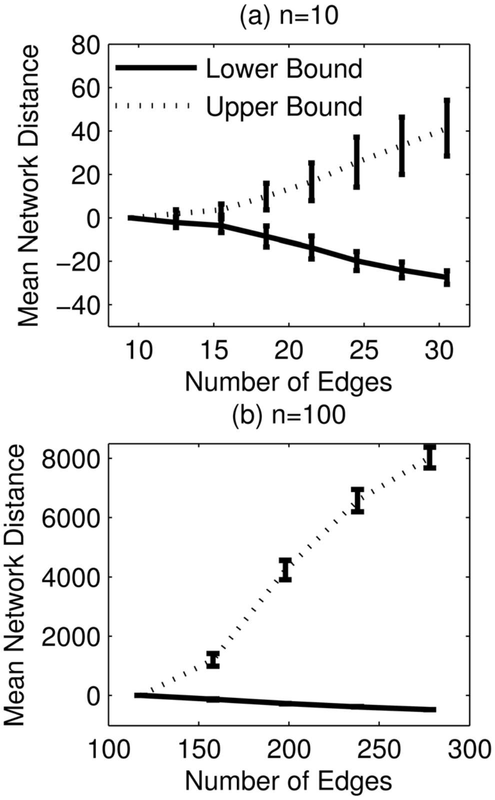


**Figure 6. Ensemble inference and inferability of 5,000 random scale-free networks of order (a) $n = 10$ and (b) $n = 100$ genes.** The mean network distances of the lower and upper bounds from $G_{\emptyset}$ are shown as a function of network size (i.e. number of edges). The error bars indicate the standard deviations.
doi:10.1371/journal.pone.0103812.g006

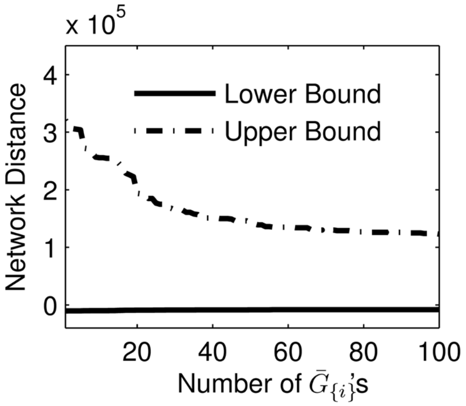


**Figure 8. Ensemble inference of *S. cerevisieae* GRN from error-free $\bar{G}_{\emptyset}$ and the 100 most informative $\bar{G}_{\{i\}}$'s based on the number of testable edges.** The plot shows the network distances of the lower and upper bounds from $G_{\emptyset}$ as a function of the number of $\bar{G}_{\{i\}}$'s. The incorporation of $\bar{G}_{\{i\}}$'s and $\underline{G}_{\{i\}}$'s was performed sequentially in decreasing number of testable edges.
doi:10.1371/journal.pone.0103812.g008

**Table 2.** Ensemble inference of *E. coli* subnetworks ($n = 100$ genes).

| FPR | FNR | Before Correction | | After Correction | | |
|---|---|---|---|---|---|---|
| | | $\mathrm{m}(G_{\emptyset}-\tilde{G}^{U})$ | $\mathrm{m}(\tilde{G}^{L}-G_{\emptyset})$ | $\mathrm{m}(G_{\emptyset}-\tilde{G}^{U})$ | $\mathrm{m}(\tilde{G}^{L}-G_{\emptyset})$ | $\mathrm{m}(\tilde{G}^{U}-\tilde{G}^{L})$ |
| 0.00 | 0.00 | 0 | 0 | 0 | 0 | 122.62 |
| 0.00 | 0.10 | 183.4 | 35.86 | 3.7 | 1.88 | 118.2 |
| 0.00 | 0.20 | 191.1 | 67.36 | 16.56 | 8.7 | 113.52 |
| 0.10 | 0.00 | 0 | 1412.42 | 0 | 0.24 | 150.4 |
| 0.10 | 0.10 | 183.02 | 1441.68 | 2.72 | 1.42 | 146.3 |
| 0.10 | 0.20 | 191 | 1460.82 | 12.06 | 2.1 | 146.74 |
| 0.20 | 0.00 | 0 | 2324.46 | 0 | 0.18 | 213.42 |
| 0.20 | 0.10 | 184.02 | 2354.16 | 2.72 | 0.46 | 197.28 |
| 0.20 | 0.20 | 191 | 2386.88 | 9.94 | 1.62 | 210.12 |

The reported values represent the averages over 50 subnetworks. FPR (FNR) is the ratio between the number of FP (FN) in the accessibility matrices and the number of edges in $G_{\emptyset}$. Let $\mathrm{m}(\mathrm{A}-\mathrm{B})$ of any two digraphs $A$ and $B$ denote the number of edges in the set $E(A) - E(B)$.
doi:10.1371/journal.pone.0103812.t002

(i.e. number of edges) between 0 and $3n$ randomly with equal probability, and assigned the edges without any preference. The upper size limit of $3n$ was chosen based on the ratio between the number of edges and the number of nodes in *E. coli* and yeast GRNs [24]. For each random network, we generated $n+1$ accessibility matrices associated with $G_{\emptyset}$ and $G_{\{i\}}$ for every $i \in V(G_{\emptyset})$. These accessibility matrices correspond to performing the full set of single- and double-gene KO experiments.

We applied TRaCE without error correction to construct the ensemble lower and upper bounds for each random network using the aforementioned accessibility matrices. The mean network distances of the bounds from $G_{\emptyset}$ are shown in Figs. 4(a) and (b) as a function of network size. Here, we plotted the network distances of the lower bound using negative numbers and those of the upper bound using positive numbers. By doing so, we could illustrate the distance between the lower and upper bounds in the same plot. In particular, the number of edges in the set $G^{U}-G^{L}$ is equal to the distance between the two network distance curves in Fig. 4. Not surprisingly, the network distance increased with the size of the networks, i.e. larger networks are more difficult to infer than smaller networks. The difference between the lower and upper bounds also broadened with network size, indicating higher network uncertainty in the inference of larger GRNs. For networks containing fewer edges than nodes, the GRN $G_{\emptyset}$ could generally be recovered from $\bar{G}_{\emptyset}$ and $\bar{G}_{\{i\}}$'s. Nevertheless, Fig. 4 demonstrated that the GRNs were typically (64% for 10 gene networks and 76% for 100 gene networks) not inferable, since the lower and upper bounds did not converge.

Fig. 5 shows two examples of GRN inference of order $n = 10$ genes. In the first case (case I, $m = 8$ edges), $G_{\emptyset}$ could be recovered from $\bar{G}_{\emptyset}$ and as few as 3 $\bar{G}_{\{i\}}$'s, while in the second case (case II, $m = 13$ edges), the inference problem was underdetermined. Moreover, the results suggested that $\bar{G}_{\{i\}}$'s were not equally informative, as the reduction in the distance between the lower and upper bounds by incorporating an additional $\bar{G}_{\{i\}}$ was not uniform.

**Random scale-free GRNs.** Many cellular networks have been shown to be scale-free with a power-law degree distribution [25], where the majority of the nodes have low degrees (1 to 2) and a few nodes (called hubs) are of high degrees. We also tested the

**Table 3.** Ensemble inference of *E. coli* GRN.

| FPR | FNR | Before Correction | | After Correction | | |
|---|---|---|---|---|---|---|
| | | $\mathrm{m}(G_{\emptyset}-\tilde{G}^{U})$ | $\mathrm{m}(\tilde{G}^{L}-G_{\emptyset})$ | $\mathrm{m}(G_{\emptyset}-\tilde{G}^{U})$ | $\mathrm{m}(\tilde{G}^{L}-G_{\emptyset})$ | $\mathrm{m}(\tilde{G}^{U}-\tilde{G}^{L})$ |
| 0.00 | 0.00 | 0 | 0 | 0 | 0 | 3351 |
| 0.00 | 0.10 | 3550 | 1029 | 25 | 46 | 3281 |
| 0.00 | 0.20 | 3746 | 1685 | 56 | 173 | 3092 |
| 0.10 | 0.00 | 0 | 34796 | 0 | 1 | 4636 |
| 0.10 | 0.10 | 3573 | 35566 | 14 | 1 | 4624 |
| 0.10 | 0.20 | 3746 | 35975 | 49 | 4 | 4556 |
| 0.20 | 0.00 | 0 | 62035 | 0 | 1 | 14933 |
| 0.20 | 0.10 | 3554 | 62632 | 12 | 7 | 14773 |
| 0.20 | 0.20 | 3741 | 63053 | 40 | 5 | 14392 |

FPR (FNR) is the ratio between the number of FP (FN) in the accessibility matrices and the number of edges in $G_{\emptyset}$. Let $\mathrm{m}(\mathrm{A}-\mathrm{B})$ of any two digraphs $A$ and $B$ denote the number of edges in the set $E(A) - E(B)$.
doi:10.1371/journal.pone.0103812.t003

**Table 4.** Ensemble inference of *S. cerevisiae* GRN.

| FPR | FNR | Before Correction | | After Correction | | |
|---|---|---|---|---|---|---|
| | | $m(G_{\emptyset}-\tilde{G}^U)$ | $m(\tilde{G}^L-G_{\emptyset})$ | $m(G_{\emptyset}-\tilde{G}^U)$ | $m(\tilde{G}^L-G_{\emptyset})$ | $m(\tilde{G}^U-\tilde{G}^L)$ |
| 0.00 | 0.00 | 0 | 0 | 0 | 0 | 131595 |
| 0.00 | 0.10 | 4048 | 604 | 9 | 4 | 131879 |
| 0.00 | 0.20 | 6934 | 788 | 19 | 8 | 132155 |
| 0.10 | 0.00 | 0 | 121370 | 0 | 4 | 198624 |
| 0.10 | 0.10 | 4096 | 121563 | 8 | 4 | 198762 |
| 0.10 | 0.20 | 6879 | 121747 | 16 | 6 | 198883 |
| 0.20 | 0.00 | 0 | 227013 | 0 | 2 | 260484 |
| 0.20 | 0.10 | 4113 | 227087 | 4 | 3 | 260443 |
| 0.20 | 0.20 | 6909 | 227150 | 8 | 2 | 260313 |

Let $m(A-B)$ of any two digraphs $A$ and $B$ denote the number of edges in the set $E(A) - E(B)$.

doi:10.1371/journal.pone.0103812.t004

performance of TRaCE using random scale-free networks. Here, we constructed two sets of 5000 scale-free GRNs with order $n=10$ and $n=100$ genes using the Barabási–Albert model [26]. Briefly, the GRNs were grown from a random seed network of small size (with 3 vertices) by sequentially adding nodes to the network. For each node addition, between 1 and 5 new edges were inserted to the network connecting the new node with existing ones, in a manner such that the degree distribution decayed exponentially. Again, for the purpose of inferabiliity analysis, we generated $n+1$ error-free accessibility matrices $\bar{G}_{\emptyset}$ and all $\bar{G}_{\{i\}}$'s, equivalent to having ideal data from single- and double-gene KO experiments.

We used the error-free implementation of TRaCE to construct the ensemble lower and upper bounds for each of the random scale-free GRNs. Fig. 6 shows the mean network distances of the bounds as a function of network size. Similar to the random GRNs, most (79% for 10 gene networks and 75% for 100 gene networks) scale-free GRNs were not inferable from single and double-gene KO experiments, as the ensemble lower and upper bound did not meet for the majority of the networks. The mean network distance of the lower and upper bounds again increased with network size. However, the inference of scale-free GRNs from the accessibility matrices $\bar{G}_{\emptyset}$ and $\bar{G}_{\{i\}}$'s appeared to be more difficult than that of random GRNs, as suggested by the larger distances between the lower and upper bounds for scale-free GRNs than for random GRNs of the same size.

***E. coli* and *S. cerevisiae* GRNs.** Finally, we investigated the inferability of large, realistic GRNs of *E. coli* and *S. cerevisiae* available in GeneNetWeaver [24]. The *E. coli* GRN consists of 1565 genes and 3758 edges, while the yeast GRN comprise 4441 genes and 12873 edges. For *E. coli*, we generated the accessibility matrices of $G_{\emptyset}$ and all $G_{\{i\}}$'s. To reduce computational complexity, in the case of yeast, we used only the 100 most informative $\bar{G}_{\{i\}}$'s based on the number of testable edges (i.e. the number of non-zero elements in the testability matrix $T_{\{i\}}$ in Eq. (4)). The results are shown in Figs. 7 and 8. Not surprisingly, both *E. coli* and yeast GRNs could not be completely inferred from the above accessibility matrices. There was a diminishing return of information after about 25 and 50 $\bar{G}_{\{i\}}$'s for the inference of *E. coli* and yeast GRNs, respectively.

## Ensemble inference from errorneous accessibility matrices

We evaluated the performance of TRaCE with error correction using *E. coli* GRN and subnetworks, as well as yeast GRN. False positive errors were simulated by randomly adding edges to the accessibility matrices, while false negatives were simulated by randomly removing edges from the accessibility matrices. The performance of error correction in TRaCE was judged by the number of erroneous edges that remained in the bounds after correction for different FP and FN rates (abbreviated as FPR and FNR, respectively), defined with respect to the size of $G_{\emptyset}$.

***E. coli* GRNs.** We first used TRaCE with error correction for the ensemble inference of 50 random subnetworks of *E. coli* GRN with $n=100$ genes, generated using GeneNetWeaver [24]. The average number of edges was 192. As in the above case study, we

**Table 5.** Ensemble inference of DREAM4 100-gene gold standard networks: single-gene KO dataset.

| Network | FPR | FNR | $m(G_{\emptyset})$ | $m(G_{\emptyset}-\tilde{G}^U)$ | $m(\tilde{G}^L-G_{\emptyset})$ | $m(\tilde{G}^U-\tilde{G}^L)$ |
|---|---|---|---|---|---|---|
| 1 | 0 | 2.28 | 176 | 44 | 9 | 136 |
| 2 | 0 | 1.96 | 249 | 122 | 6 | 123 |
| 3 | 0 | 11.04 | 195 | 86 | 21 | 144 |
| 4 | 0 | 9.69 | 211 | 91 | 7 | 208 |
| 5 | 0 | 3.47 | 193 | 108 | 12 | 237 |

FPR (FNR) is the ratio between the number of FP (FN) in the accessibility matrices and the number of edges in $G_{\emptyset}$. Let $m(G_{\emptyset})$ denote the number of edges in $G_{\emptyset}$, and $m(A-B)$ of any two digraphs $A$ and $B$ denote the number of edges in the set $E(A) - E(B)$.

doi:10.1371/journal.pone.0103812.t005

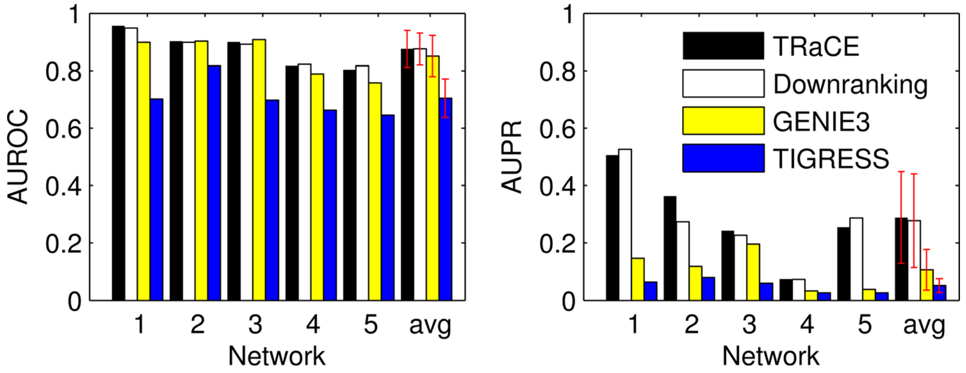


**Figure 9. Comparison of TRaCE and top performing methods in DREAM4 100-gene network inference subchallenge: single-gene KO dataset.** The error bars represent the standard deviations. Based on the AUROC values, TRaCE performed as well as the downranking method ($p=0.5$) and GENIE3 ($p=0.3$), but better than TIGRESS ($p=0.002$). Based on the AUPR values, TRaCE performed as well as the downranking method ($p=0.5$), but better than GENIE3 ($p=0.03$) and TIGRESS ($p=0.01$). The statistical significance was evaluated using two sample t-test.
doi:10.1371/journal.pone.0103812.g009

created the accessibility matrices of $G_\emptyset$ and every $G_{\{i\}}$. We subsequently contaminated these matrices with FP and FN errors at the specified rates without any preference. The accuracy of the lower and upper bounds constructed using TRaCE with and without error correction is summarized in Table 2.

As in many scenarios above, none of the subnetworks was inferable. FP errors could be very effectively eliminated by error correction. FNs errors expectedly led to missing edges from the upper bound, as indicated by the number of edges of $G_\emptyset$ that did not appear in $\tilde{G}^U$ (see $m\left(G_\emptyset-\tilde{G}^U\right)$ in Table 2). The error correction could not completely eliminate Type A errors, leading to erroneous edges that appeared in the lower bound $\tilde{G}^L$ but did not belong to $G_\emptyset$ (see $m\left(\tilde{G}^L-G_\emptyset\right)$ in Table 2). A combination of FP and FN errors were more easily corrected than FN errors alone. While FNs were more difficult to eliminate than FPs, correcting FP errors tended to produce larger network ensembles than FNs, indicating higher network uncertainty (see $m\left(\tilde{G}^U- \tilde{G}^L\right)$ in Table 2). Nevertheless, even in the worst case (0% FP, 20% FN), roughly 90% of the errors in the lower and upper bounds could be removed by the error correction (compare $m(G_\emptyset-\tilde{G}^U)+m\left(\tilde{G}^L-G_\emptyset\right)$ before and after correction).

For the inference of *E. coli* GRN, we generated erroneous accessibility matrices $\bar{G}_\emptyset$ and the 100 most informative $\bar{G}_{\{i\}}$'s corresponding to the top 100 highest numbers of testable edges. The performance of TRaCE with error correction for different FP and FN rates is summarized in Table 3. In addition, the structural Hamming distances of the lower and upper bounds before and after correction are reported in tables S1 and S2. As before, TRaCE with error correction could handle FPs more effectively than FNs, and a mixture of FP and FN errors in the accessibility matrices were more easily eliminated than FN alone. In the worst case (0% FP, 20% FN), more than 95% of the errors were corrected. The size of the ensemble also depended strongly on the FP errors, and at 20% FP, the number of edges between the lower and upper bound reached three times the size of the full GRN.

***S. cerevisiae* GRN.** For yeast GRN, we generated erroneous $\bar{G}_\emptyset$ and the 100 most informative $\bar{G}_{\{i\}}$'s. The results of TRaCE with error correction using these accessibility matrices are summarized in Table 4. The performance of TRaCE here was notably better than the inference of *E. coli* GRN. In all cases, TRaCE could rectify almost all erroneous edges. However, the correction came at a price of high uncertainty, where the difference between the lower and upper bounds exceeded 20 times the number of edges in $G_\emptyset$. Despite such high uncertainty, the gap between the bounds represented only 1.3% of the total possible edges.

## Ensemble inference from expression data

We further evaluated the performance of TRaCE using *in silico* noisy gene expression data generated using GeneNetWeaver [24]. We simulated steady-state gene expression data using the same settings as those in DREAM4 100-gene *in silico* network inference subchallenge. The simulated data are available at http://www.cabsel.ethz.ch/tools/trace or upon request. In the following case studies, we analyzed and converted the data into accessibility matrices following the procedure described in the Numerical Implementation section. Subsequently, we used the resulting

**Table 6.** Ensemble inference of DREAM4 100-gene gold standard networks: single- and double-gene KO dataset.

| Network | FPR | FNR | $\mathrm{m}(G_\emptyset)$ | $\mathrm{m}(G_\emptyset-\tilde{G}^U)$ | $\mathrm{m}(\tilde{G}^L-G_\emptyset)$ | $\mathrm{m}(\tilde{G}^U-\tilde{G}^L)$ |
|---|---|---|---|---|---|---|
| 1 | 0.02 | 2.21 | 176 | 50 | 5 | 27 |
| 2 | 0.01 | 1.91 | 249 | 137 | 5 | 37 |
| 3 | 0 | 10.43 | 195 | 103 | 15 | 30 |
| 4 | 0 | 9.31 | 211 | 118 | 8 | 35 |
| 5 | 0.01 | 3.40 | 193 | 131 | 12 | 21 |

FPR (FNR) is the average ratio between the number of FP (FN) in the accessibility matrices and the number of edges in $G_\emptyset$. Let $\mathrm{m}(G_\emptyset)$ denote the number of edges in $G_\emptyset$, and $\mathrm{m}(A-B)$ of any two digraphs $A$ and $B$ denote the number of edges in the set $E(A)-E(B)$.
doi:10.1371/journal.pone.0103812.t006

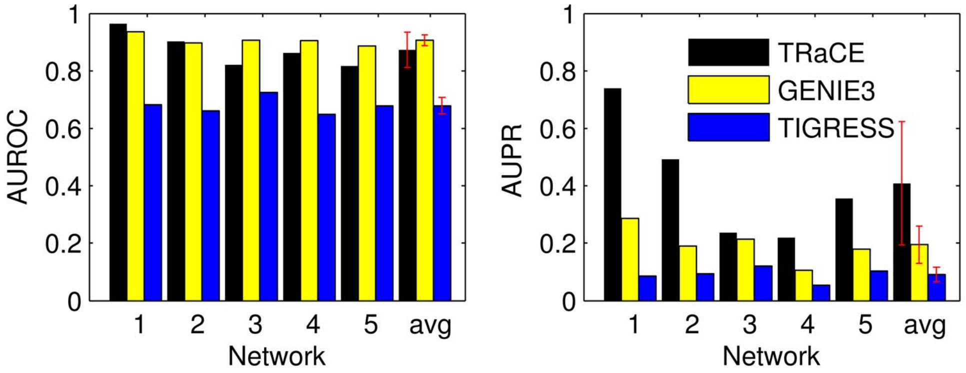


**Figure 10. Comparison of TRaCE and top performing methods in DREAM4 100-gene network inference subchallenge: single- and double-gene KO dataset.** The error bars represent the standard deviations. Based on the AUROC values, TRaCE performed as well as GENIE3 ($p=0.8$) and better than TIGRESS ($p=0.0005$). Similarly, based on the AUPR values, TRaCE performed better than GENIE3 ($p=0.04$) and TIGRESS ($p=0.01$). The statistical significance was evaluated using two sample t-test.
doi:10.1371/journal.pone.0103812.g010

accessibility matrices with TRaCE to produce the ensemble lower and upper bounds. For the purpose of comparison with existing network inference methods, we also ranked the gene regulatory interactions according to their confidence scores. In particular, we compared the rankings with those from top performing inference methods in DREAM4, namely the downranking method by Pinna et al. [27], GENIE3 [28] and TIGRESS [29]. As mentioned earlier, the downranking method follows a two-phase procedure in ranking edges similar to our implementation, but the method only down-ranks feed-forward edges. On the other hand, GENIE3 uses a machine learning strategy called random forest, and TIGRESS is a regression-based method. For each method, we calculated the area under receiver operating characteristics (AUROC) and precision-recall (AUPR) using a redefined confusion matrix, in which methods were not penalized for any error within the set of non-inferable edges. We define non-inferable edges as edges belonging to the upper bound that are missing from the lower bound (i.e. all edges in the set $G^U - G^L$), which are determined from error-free accessibility matrices of the gold standard GRNs. More details of the calculation of the AUROC and AUPR can be found in a recent publication [30].

**DREAM4 *in silico* network inference 100-gene subchallenge.** We first simulated 5 replicates of steady state gene expression data associated with the complete single-gene KO experiments. We then used the data to construct the accessibility matrices of the gold standard GRNs. In this case, the upper bound of the ensemble was simply given by the accessibility matrix, and the lower bound was the ConTREx of the upper bound. Table 5 shows the FPR and FNR in the accessibility matrices, the errors in the lower and upper bounds ($m\left(G_\emptyset - \tilde{G}^U\right)$ and $m\left(\tilde{G}^L - G_\emptyset\right)$), and the size of the ensemble ($m\left(\tilde{G}^U - \tilde{G}^L\right)$) constructed using TRaCE. We noted that the majority (90%) of FN errors in the accessibility matrices were associated with fan-in motifs, in which a gene was regulated by several genes. In such a case, the effect of knocking-out one of the regulator genes could be compensated by the others and thus, the KO experiment did not show any significant differential expression of the downstream genes. As FNs affected the accessibility matrices, the errors in the upper bound $m\left(G_\emptyset - \tilde{G}^U\right)$ were higher than those in the lower bound $m\left(\tilde{G}^L - G_\emptyset\right)$.

Subsequently, we created a ranked list of gene regulatory predictions and compared the list against those produced by the downranking method, GENIE3 and TIGRESS. Fig. 9 provides the comparison of AUROC and AUPR of the four methods. The comparison showed that TRaCE and the downranking method outperformed GENIE3 and TIGRESS, especially considering the AUPR values. Here, TRaCE performed as well as the downranking method, which was the best overall performer in DREAM 4 100-gene network inference subchallenge [27].

**Ensemble inference from single- and double-gene Kos.** We further simulated 5 replicates of steady state gene expression data for the complete set of single- and double-gene KO experiments using the gold standard GRNs of DREAM4 100-gene subchallenge. We processed the data to obtain the accessibility matrices of $G_\emptyset$ and all $G_{\{i\}}$'s. We subsequently applied TRaCE with error correction to the accessibility matrices to obtain the ensemble lower and upper bounds, which are

**Table 7.** Effects of $z_{cutoff}$ and $z_{threshold}$ on AUROC and AUPR of TRaCE in DREAM4 100-gene subchallenge: single gene KO data.

| $z_{threshold}$ | $z_{cutoff}$ | AUROC | AUPR |
|---|---|---|---|
| 1.5 | 2 | 0.871 ± 0.0616 | 0.3085 ± 0.1416 |
| 1.5 | 3 | 0.8723 ± 0.0673 | 0.3209 ± 0.1534 |
| 2 | 2 | 0.8756 ± 0.0595 | 0.2922 ± 0.154 |
| 2 | 3 | **0.8758 ± 0.0642** | **0.2889 ± 0.1596** |
| 2.5 | 3 | 0.8758 ± 0.0642 | 0.2438 ± 0.1545 |

The AUROC and AUPR values are the average ± standard devation over 5 gold standard networks. The values used in the comparison with existing methods are highlighted in bold.
doi:10.1371/journal.pone.0103812.t007

**Table 8.** Effects of $z_{cutoff}$, $z_{threshold}$ and the preprocessing threshold of $\bar{G}_{\emptyset}$ on AUROC and AUPR of TRaCE in DREAM4 100-gene subchallenge: single- and double-gene KO data.

| $z_{threshold}$ | $z_{cutoff}$ | **threshold** | **AUROC** | **AUPR** |
|---|---|---|---|---|
| 1.5 | 2 | 0.5 | 0.8723 ± 0.0595 | 0.4327 ± 0.1931 |
| | | 0.65 | 0.8718 ± 0.0588 | 0.4197 ± 0.1959 |
| | | 0.8 | 0.8724 ± 0.0600 | 0.4184 ± 0.2022 |
| 1.5 | 3 | 0.5 | 0.8728 ± 0.0596 | 0.4204 ± 0.1986 |
| | | 0.65 | 0.8733 ± 0.0610 | 0.4212 ± 0.2059 |
| | | 0.8 | 0.8735 ± 0.0617 | 0.4201 ± 0.2100 |
| 2 | 2 | 0.5 | 0.8726 ± 0.0603 | 0.4177 ± 0.2158 |
| | | 0.65 | 0.8727 ± 0.0607 | 0.4167 ± 0.2170 |
| | | 0.8 | 0.8728 ± 0.0608 | 0.4091 ± 0.2139 |
| 2 | 3 | 0.5 | 0.8735 ± 0.0618 | 0.4170 ± 0.2160 |
| | | 0.65 | **0.8734 ± 0.0620** | **0.4086 ± 0.2149** |
| | | 0.8 | 0.8698 ± 0.0604 | 0.4045 ± 0.2108 |
| 2.5 | 3 | 0.5 | 0.8695 ± 0.0599 | 0.3893 ± 0.2054 |
| | | 0.65 | 0.8696 ± 0.0598 | 0.3837 ± 0.1975 |
| | | 0.8 | 0.8695 ± 0.0600 | 0.3764 ± 0.2023 |

The AUROC and AUPR values are the average ± standard devation over 5 gold standard GRNs. The values used in the comparison with existing methods are highlighted in bold.

doi:10.1371/journal.pone.0103812.t008

summarized in Table 6. The average FPR and FNR were similar to the single-gene KO data since both datasets had the same number of replicates. Again, the majority (80%) of errors in the accessibility matrices were associated with fan-in motifs. By comparing Tables 5 and 6, the errors in the lower bounds improved slightly in comparison with those from only single-gene KO dataset (compare $m(\tilde{G}^L - G_{\emptyset})$ values). However, the errors in the upper bounds increased due to the accumulation of FN errors from fan-in motifs (compare $m(G_{\emptyset} - \tilde{G}^U)$ values). Nevertheless, the additional data from double-gene KO experiments led to lower network uncertainties (compare $m(\tilde{G}^U - \tilde{G}^L)$ values).

For each gold standard GRN, we also generated a ranked list of edges based on the confidence scores and compared the list with those using GENIE3 and TIGRESS. The downranking method could not be applied to double-gene KO data and was left out from the comparison. The AUROC and AUPR of the three methods are compared in Fig. 10. The AUPR of TRaCE was higher than GENIE3 and TIGRESS. Meanwhile, the AUROC values were generally high for all three methods, with TIGRESS having the lowest value. In comparison to single-gene KO data, the inclusion of double-gene KO data led to no change in the average AUROC ($p = 0.52$, two sample t-test) and an increase in AUPR ($p = 0.17$, two sample t-test). Finally, as shown in Tables 7 and 8, the AUROC and AUPR values of TRaCE were insensitive to $z_{cutoff}$ and $z_{threshold}$ used in the construction of the accessibility matrices, and the threshold value in the preprocessing of $\bar{G}_{\emptyset}$. Because of the trade-off between FPs and FNs when using different $z_{cutoff}$ and $z_{threshold}$, the maximum of AUROC and AUPR corresponded to intermediate values within the selected ranges of $z_{cutoff}$ and $z_{threshold}$.

***E. coli* and Yeast GRNs.** Finally, we simulated the complete set of single-gene KO experiments for *E. coli* and yeast. For each organism, we generated 10 replicates of steady state gene expression data. We performed differential expression analysis using the Z-score transformation and obtained the accessibility matrices using either 5 or 10 replicates. We then applied TRaCE with error correction to construct the ensemble lower and upper bounds, and created ranked lists of edges as done earlier. The errors in the accessibility matrices and in the bounds are reported in Table 9, along with the AUROC and AUPR values. Here, FPR in the accessibility matrices decreased with increasing number of replicates, but FNR did not change with the number of replicates. The errors in the upper bounds changed little with increasing technical replicates from 5 to 10, but those in the lower bounds dropped considerably. The sizes of the ensemble also decreased with increasing replicates. Finally, the AUPR values improved slightly with higher replicates, while the AUROC values were insensitive with respect to the number of replicates.

## Discussion

The inference of GRNs from data of gene perturbation experiments is an important but unsolved problem. The difficulty stems from the underdetermined nature of such an inference [7,9], as the data do not contain the necessary information to establish the complete causal interactions among the genes. Consequently, there exist many indistinguishable solutions. We have developed TRaCE with this consequence in mind by employing an ensemble inference strategy. Specifically, we have taken into consideration the fundamental limitation in using steady-state expression data of gene KO experiments for establishing direct causal relationships among genes. In TRaCE, we first transform the expression data into accessibility relationships (matrices) among genes. The novel contribution of TRaCE is an algorithm for the construction of lower and upper bounds of network ensemble, where each member of the ensemble satisfies the accessibility matrices. Edges of the upper bound that do not appear in the lower bound are considered non-inferable, as the existence of such edges can not be verified. Here, the size of the ensemble provides a metric of uncertainty in the network inference problem, with which the GRN inferability can be rigorously assessed. The GRN is inferable

**Table 9.** Ensemble inference of *E. coli* and Yeast GRNs from single-gene KO data.

| Network | Replicates | FPR | FNR | $\mathrm{m}(\mathrm{G}_{\emptyset}-\widetilde{\mathrm{G}}^{\mathrm{U}})$ | $\mathrm{m}(\widetilde{\mathrm{G}}^{\mathrm{L}}-\mathrm{G}_{\emptyset})$ | $\mathrm{m}(\widetilde{\mathrm{G}}^{\mathrm{U}}-\widetilde{\mathrm{G}}^{\mathrm{L}})$ | AUROC | AUPR |
|---|---|---|---|---|---|---|---|---|
| *E. coli* | 5 | 0.117 | 2.61 | 1611 | 687 | 1724 | 0.8601 | 0.4836 |
| *E. coli* | 10 | 0.005 | 2.61 | 1612 | 297 | 1673 | 0.8639 | 0.5192 |
| Yeast | 5 | 0.582 | 24.97 | 8101 | 9159 | 7907 | 0.7699 | 0.2464 |
| Yeast | 10 | 0.141 | 24.98 | 8098 | 3753 | 7200 | 0.7569 | 0.2768 |

FPR (FNR) is the ratio between the number of FP (FN) in the accessibility matrices and the number of edges in $G_{\emptyset}$. Let $\mathrm{m}(\mathrm{A}-\mathrm{B})$ of any two digraphs $A$ and $B$ denote the number of edges in the set $E(A) - E(B)$.
doi:10.1371/journal.pone.0103812.t009

when the lower and upper bounds coincide (i.e. the ensemble only contains one network). Thus, in TRaCE, the inference and inferability analysis are accomplished simultaneously.

In the case studies, we have demonstrated the use of TRaCE for analyzing the inferability of GRNs. With the exception of networks of low order and small size, the majority of the GRNs were not inferable even when using error-free accessibility matrices. As we have used sparse networks in the case studies, the lower bound of the ensemble was a better estimate of the GRN than the upper bound. Finally, the majority of double-gene KO experiments were non-informative as the reduction in the size of the ensemble diminished after only a small number of $\bar{G}_{\{i\}}$'s. The observation above suggests that experimental design contributes significantly to the underdetermined nature of the typical GRN inference. In this regard, the lower and upper bounds of the ensemble could be used for optimizing the gene perturbation experiments, for example by finding the KO experiment that provides the maximum reduction in the difference between the lower and upper bounds. A strategy for optimal design of experiments using ensemble inference will be presented in a future publication.

We have also used the ensemble lower and upper bounds in conjunction with the z-scores to produce a ranked list of gene regulatory predictions. In comparison with the top methods of DREAM4 network inference challenge, TRaCE could match the performance of the downranking method, the best overall performer in the 100-gene subchallenge. For single-gene KO dataset, TRaCE and the downranking method differed only for edges that were involved in cycles of more than 2 nodes. However, the two methods were fundamentally different, as TRaCE was developed for ensemble inference. Furthermore, the downranking method was created for single-gene KO experiments. Meanwhile, TRaCE significantly outperformed GENIE3 and TIGRESS when using single- and double-gene KO data. We note that GENIE3 and TIGRESS were also among the best performers in DREAM5 network inference challenge [10].

As expected, data noise negatively influenced the GRN inference and increased the uncertainty in the GRN inference. In the case studies, random errors in the accessibility matrices expectedly led to a larger ensemble. While the error correction in TRaCE was able to eliminate the majority of errors, some of the errors remained in the bounds. In particular, FN errors were harder to correct than FPs. The reason was that more Type A errors originating from FNs passed through the correction than those originating from FPs (see Fig. S3 and text S4). Meanwhile, FP errors could cancel out some Type A errors associated with FNs at the cost of increased uncertainty (see Fig. S4 and text S4).

The application of TRaCE to simulated noisy gene expression data indicated that the majority of errors in the accessibility matrices were due to FNs associated with fan-in motifs in the GRN. In such motifs, the effects of knocking-out one regulator gene could be compensated by other regulator(s), and differential expression analysis could only reveal the dominant regulator(s) of a gene. Note that such a problem could not be improved by increasing technical replicates (see Table 9). Meanwhile, errors associated with fan-in motifs usually lead to type A errors where the affected edges are absent from the lower and upper bounds. However, if the available experiments permit the construction of $\bar{G}_{V_{KO}}$ in which $V_{KO}$ includes the dominant regulator(s) of a fan-in motif, then the related missing edge(s) may appear in $\underline{G}_{V_{KO}}$, leading to a detectable and correctable type C error. We expect the issue above would improve when using more sensitive measurements of gene expression, for example RNAseq.

## Conclusion

Inferring gene regulatory networks from DNA microarray data is an unsolved problem. Community-wide assessments of inference methods have shown that distinguishing direct and indirect regulatory interactions among genes is a common Achilles' heel of existing algorithms. In this study, we have adopted an ensemble inference strategy and develop new framework and algorithms for the creation of an ensemble of networks. Here, the ensemble represents the uncertainty associated with differentiating direct and indirect regulations using steady-state gene expression data of gene KO experiments. In particular, TRaCE produces the lower and upper bound of the ensemble. Using the bounds of the ensemble, a ranked list of gene regulatory predictions can also be generated. The case studies demonstrate that except for networks with few edges, most GRNs can not be fully inferred even when error-free data from complete single- and double-gene knock out experiments are available. In comparison with top performing methods of DREAM4 *in silico* network inference challenge, TRaCE performed equally well with the downranking method, the best overall performer in the challenge. However, the downranking method is not designed to handle data from multi-gene KO experiments. Meanwhile, TRaCE outperformed GENIE3 and TIGRESS when using single- and double-gene KO data. Nevertheless, the uncertainty in GRN inference is still significant, and systematic KOs of genes are often suboptimal as only a small fraction of the experiments are informative. We therefore hope that the shift of paradigm to ensemble inference will trigger further developments of methodologies for inferring and analyzing GRNs, in which network inference uncertainty is explicitly taken into consideration.

## Supporting Information

**Figure S1 An example of ConTREx of a simple GRN.** The directed edges indicated by blue arrows in $\bar{G}_{\emptyset}$ are in the set of indirect regulations. The edges between A and B are retained,

because the cycle contains only two nodes. If the cycle had contained more than two nodes, all edges among the nodes would have been removed.
(TIFF)

**Figure S2 Example of an ensemble involving GRN with cycle.** Consider a GRN consisting of genes A, B and C, all of which are involved in a directed cycle. Further, let us assume that the edge C→A belongs to the lower bound and A→C is not in the the upper bound. In this case, the ensemble comprises the graphs shown in (a)-(d).
(TIFF)

**Figure S3 Type A errors due to an FN.** (a) $G_\emptyset$; (b) $\bar{G}_\emptyset$; (c) $\bar{G}_{\{B\}}$; (d) $\underline{G}_{\{B\}}$; (e) $\bar{G}^M_{\{B\}}$ with an FN at C→D. In this case, $\underline{G}^M_{\{B\}} = \bar{G}^M_{\{B\}}$. (f) $\bar{G}^M_{\{B\}}$ with an FN at C→D and an FP at C→A (g) ConTREx reduction of (f). (h) $\bar{G}^M_{\{B\}}$ with an FN at C→D and an FP at C→E (i) ConTREx reduction of (h). See text S4 for details.
(TIFF)

**Figure S4 Type A error due to an FP.** $G_\emptyset$ and $\bar{G}_\emptyset$ are shown in Fig. S3 (a) and (b), respectively. (a) $\bar{G}_{\{C\}}$. In this case, $\underline{G}_{\{C\}} = \bar{G}_{\{C\}}$. (b) $\bar{G}^M_{\{C\}}$with an FP at B→D. Here, $\underline{G}^M_{\{C\}} = \bar{G}^M_{\{C\}}$.
(TIFF)

**Table S1 Performance of TRaCE on inference of *E. coli* subnetworks ($n = 100$ genes).** The reported values represent the average over 50 subnetworks. Let $\mathrm{D}(\mathrm{A}-\mathrm{B})$ of any two digraphs $A$ and $B$ denote the structural Hamming distance (SHD) between them. The SHD is defined as the number of edges which differ or have opposite orientation between two networks [31].
(PDF)

**Table S2 Performance of TRaCE on inference of *E. coli* GRN.** Let $\mathrm{D}(\mathrm{A}-\mathrm{B})$ of any two digraphs $A$ and $B$ denote the structural Hamming distance between them.
(PDF)

**Text S1 Preprocessing of $\bar{G}_\emptyset$.**
(PDF)

**Text S2 Filter Algorithms.**
(PDF)

**Text S3 Consistency check.**
(PDF)

**Text S4 Type A errors due to FN.**
(PDF)

## Acknowledgments

The authors would like to thank Caroline Siegenthaler, Erica Manesso and Heeju Noh for useful comments.

## Author Contributions

Conceived and designed the experiments: SMMU RG. Performed the experiments: SMMU. Analyzed the data: SMMU RG. Contributed reagents/materials/analysis tools: SMMU RG. Wrote the paper: SMMU RG.

## References

1. Apic G, Ignjatovic T, Boyer S, Russell RB (2005) Illuminating drug discovery with biological pathways. FEBS Lett 579: 1872–1877.
2. Alper H, Stephanopoulos G (2009) Engineering for biofuels: exploiting innate microbial capacity or importing biosynthetic potential? Nat Rev Microbiol 7: 715–723.
3. Oberhardt MA, Palsson B, Papin JA (2009) Applications of genome-scale metabolic reconstructions. Mol Syst Biol 5: 320.
4. Zhao XM, Iskar M, Zeller G, Kuhn M, van Noort V, et al. (2011) Prediction of drug combinations by integrating molecular and pharmacological data. PLoS Comput Biol 7: e1002323.
5. Hecker M (2009) Gene regulatory network inference: data integration in dynamic models–a review. Biosystems 96: 86–103.
6. Margolin AA, Califano A (2007) Theory and limitations of genetic network inference from microarray data. Ann N Y Acad Sci 1115: 51–72.
7. Smet RD, Marchal K (2010) Advantages and limitations of current network inference methods. Nature Reviews Microbiology 8: 717–729.
8. Stolovitzky G, Prill RJ, Califano A (2009) Lessons from the dream2 challenges. Ann N Y Acad Sci 1158: 159–195.
9. Marbach D, Prill RJ, Schaffter T, Mattiussi C, Floreano D, et al. (2010) Revealing strengths and weaknesses of methods for gene network inference. Proc Natl Acad Sci USA 107: 6286–6291.
10. Marbach D, Costello JC, Küffner R, Vega NM, Prill RJ, et al. (2012) Wisdom of crowds for robust gene network inference. Nat Methods 9: 796–804.
11. Gadkar KG, Gunawan R, Doyle FJ 3rd (2005) Iterative approach to model identification of biological networks. BMC Bioinformatics 6: 155.
12. Srinath S, Gunawan R (2010) Parameter identifiability of power-law biochemical system models. J Biotechnol 149: 132–140.
13. Chis OT, Banga JR, Balsa-Canto E (2011) Structural identifiability of systems biology models: A critical comparison of methods. PLoS ONE 6: e27755.
14. Szederkényi G, Banga J, Alonso A (2011) Inference of complex biological networks: distinguishability issues and optimization-based solutions. BMC systems biology 5: 177.
15. Kuepfer L, Peter M, Sauer U, Stelling J (2007) Ensemble modeling for analysis of cell signaling dynamics. Nat Biotechnol 25: 1001–1006.
16. Miskovic L, Hatzimanikatis V (2011) Modeling of uncertainties in biochemical reactions. Biotechnol Bioeng 108: 413–423.
17. Tan Y, Liao JC (2012) Metabolic ensemble modeling for strain engineers. Biotechnol J 7: 343–353.
18. Jia G, Stephanopoulos G, Gunawan R (2012) Ensemble kinetic modeling of metabolic networks from dynamic metabolic profiles. Metabolites 2: 891–912.
19. Wagner A (2001) How to reconstruct a large genetic network from n gene perturbations in fewer than n(2) easy steps. Bioinformatics 17: 1183–1197.
20. Aho AV, Garey MR, Ullman JD (1972) The transitive reduction of a directed graph. SIAM J Comput 1: 131–137.
21. Jackson S (2011) Research Methods and Statistics: A Critical Thinking Approach. Cengage Learning, 138 pp.
22. Harray F (1969) Graph Theory. Addison-Wesley, Reading, Massachusetts.
23. Pinna A, Soranzo N, de la Fuente A (2010) From knockouts to networks: Establishing direct cause-effect relationships through graph analysis. PLoS ONE 5: e12912.
24. Schaffter T, Marbach D, Floreano D (2011) Genenetweaver: in silico benchmark generation and performance profiling of network inference methods. Bioinformatics 27: 2263–2270.
25. Albert R (2005) Scale-free networks in cell biology. Journal of cell science 118: 4947–4957.
26. Albert R, Barabási AL (2002) Statistical mechanics of complex networks. Rev Mod Phys 74: 47–97.
27. Pinna A, Soranzo N, de la Fuente A (2010) From knockouts to networks: Establishing direct cause-effect relationships through graph analysis. PLoS ONE 5: e12912.
28. Huynh-Thu VA, Irrthum A, Wehenkel L, Geurts P (2010) Inferring regulatory networks from expression data using tree-based methods. PLoS ONE 5: e12776.
29. Haury AC, Mordelet F, Vera-Licona P, Vert JP (2012) Tigress: Trustful inference of gene regulation using stability selection. BMC Systems Biology 6: 145.
30. Siegenthaler C, Gunawan R (2014) Assessing network inference methods: How to cope with an underdetermined problem. PLoS One 9: e90481.
31. Tsamardinos I, Brown LE, Aliferis CF (2006) The max-min hill-climbing bayesian network structure learning algorithm. Machine Learning 65: 31–78.