Inferability and inference of gene regulatory networks

Author(s):
Ud-Dean, S.M. Minhaz

Publication Date:
2016

Permanent Link:
https://doi.org/10.3929/ethz-a-010686553

Rights / License:
In Copyright - Non-Commercial Use Permitted
INFERABILITY AND INFERENCE OF GENE REGULATORY NETWORKS

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH

(Dr. sc. ETH Zurich)

Presented by

S M MINHAZ UD-DEAN

Master of Science in Applied Physics
Delft University of Technology
&
Master of Science in Photonics
Friedrich-Schiller University Jena

Born on 24.09.1985

Citizen of BANGLADESH

Accepted on the recommendation of

Prof. Dr. Rudiyanto Gunawan (examiner)
Prof. Dr. Konrad Hungerbuehler (co-examiner)

2016
Acknowledgement

First and foremost, I would like to mention that I am deeply grateful to my supervisor Prof. Rudiyanto Gunawan for providing me the opportunity to pursue research under his supervision, for guiding me with care during my doctoral studies, for his continued support, and for patiently and painstakingly reviewing my work.

I would like to thank my current and former colleagues at the Chemical and Biological Systems Engineering Laboratory (CABSEL), Dr. Thanneer Malai Perumal, Dr. Sridharan Srinath, Dr. Lakshmi Narayanan Lakshmanan, Dr. Jia Gengjie, Dr. Zhi Yang Tam, Mr. Yang Liu, Ms Caroline Siegenthaler, Mr. Sandro Hutter, Dr. Erica Manesso, Ms Heeju Noh, Dr. Manuel Garcia-Alborno, Ms Nadia Vertti Quintero, Ms Nan Papili Gao and Mr. Sudharshan Ravi for their assistance and companionship during my doctoral studies. Specifically, I would like to thank Dr. Thanneer Malai Perumal for introducing this project to me during the initial stage of my doctoral studies. I would like to thankfully mention Dr. Zhi Yang Tam for managing the computational facilities during the first two years of my thesis. I would also like to thank Dr. Lakshmi Narayanan Lakshmanan for managing the webpage hosting the softwares developed in this thesis, and for his comments on my manuscripts and thesis drafts. I would also like to thank Dr. Erica Manesso for her comments on one of my manuscripts. I would like to thankfully mention Mr. Sandro Hutter’s assistance in the German translation of the abstract of this thesis and his help in revision of the symbols and equation. I am also thankful to Ms Nan Papili Gao and Ms Heeju Noh for helping me with managing the computational facilities of the laboratory. In this regard, I would also like to mention Dr. Erol Eedeoglu and Dr. Mathias Egloff, the former and current IT coordinators of the Institute for Chemical and Bioengineering, for their support in obtaining and installing the computers used to perform the computational analyses of this thesis.

I would also like to thank Dr. Steffen Klamt, of Max Planck institute for dynamics of complex technical systems (Magdeburg), and his Student Ms. Sandra Heise for their assistance in implementing the Local Transitive Reduction algorithm used in this thesis. I would also like to mention Dr. Andrea Pinna, formerly of the center for advanced studies, research and development (Sardinia), for helping me in this regard.

I would also like to acknowledge the support of the Swiss National Science Foundation (grants 137614 and 157154) for funding my doctoral research.

Finally, I would like to express my gratitude to my parents and siblings for their understanding and support, and for allowing me to study in a foreign land too far away to perform my filial duties. Most importantly, I would like to thank and appreciate my wife for her understanding and compromises during the course of my doctoral studies. I am greatly indebted to her for her continued efforts and sacrifices that permitted me to concentrate on my research.
# Table of Contents

Acknowledgement .................................................................................................................. i
Table of Contents ................................................................................................................... ii
Abstract ................................................................................................................................... v
Zusammenfassung ...................................................................................................................... viii
Chapter 1 Introduction ............................................................................................................ 1
  1.1 Gene Regulatory Networks ............................................................................................. 3
    1.1.1 Central Dogma of Molecular Biology ....................................................................... 4
    1.1.2 Transcriptional Regulation ....................................................................................... 5
  1.2 Properties of Gene Regulatory Networks .......................................................................... 7
    1.2.1 Sparsity ..................................................................................................................... 7
    1.2.2 Scale-free topology ................................................................................................... 8
    1.2.3 Robustness ............................................................................................................... 8
  1.3 Gene Expression Data ...................................................................................................... 9
    1.3.1 Northern Blot .......................................................................................................... 9
    1.3.2 RT-PCR ................................................................................................................... 10
    1.3.3 Microarray .............................................................................................................. 10
    1.3.4 Next Generation Sequencing .................................................................................. 11
    1.3.5 Differential Expression .......................................................................................... 11
  1.4 Inference of Gene Regulatory Networks .......................................................................... 12
    1.4.1 Inference from Gene Expression Data ....................................................................... 12
    1.4.2 Inference from DNA sequence ................................................................................. 17
  1.5 Scope of the Thesis .......................................................................................................... 18
Chapter 2 TRaCE .................................................................................................................... 22
  2.1 Introduction ..................................................................................................................... 22
  2.2 Method .......................................................................................................................... 22
    2.2.1 Theoretical Foundation ........................................................................................... 22
    2.2.2 Numerical Implementation ...................................................................................... 34
  2.3 Results ........................................................................................................................... 42
    2.3.1 Inferability Analysis ............................................................................................... 42
    2.3.2 Ensemble inference from erroneous accessibility matrices ..................................... 47
    2.3.3 Ensemble inference from expression data ............................................................... 50
  2.4 Discussion ....................................................................................................................... 57
Abstract

Understanding how different genes regulate each other is a prerequisite for systems-level modelling of molecular biology. Despite the development of a large number of inference algorithms, the inference of gene regulatory networks (GRN) from gene expression data remains an important unsolved problem of systems biology. The difficulty can be attributed to the fact that the accuracy of inferred GRNs depends not only on the ability of the algorithm to extract causal relationships from data, but also on the availability of relevant information in the data. Often it is not possible to accurately infer a GRN irrespective of the applied method because the available data do not contain sufficient information relevant to the causal structure of the network. This lack of relevant causal information causes the GRN inference problem to become underdetermined, and consequently there could be many equivalent (indistinguishable) solutions, implying that the GRN is not uniquely inferable. While the underdetermined nature of the GRN inference and its significance have been recognized in the community, the attention given to the inferability of GRNs still pales in comparison to the attention given to the development of inference algorithms.

The inferability of a GRN is determined by the causal information in a dataset, which in turn depends on the experiments performed to obtain the dataset. In consequence, during GRN inference it is necessary to take the performed experiments into consideration and analyze the inferability of the network from those experiments. Additionally, one should optimize the experiments in order to obtain new dataset that would improve the inferability of the network. Once again, the design of optimal experiments for GRN inference has received little attention compared to the development of inference methods. Motivated by these gaps, we sought to address the underdetermined issue of GRN inference in this thesis. Specifically, we have developed new framework and ensemble inference algorithms, called Transitive Reduction and
Closure Ensemble (TRaCE) for inferability analysis of gene regulatory networks from steady-state data of gene knock-out (KO) experiments. In addition, we have created REDuction of UnCertain Edges (REDUCE), an algorithm for designing optimal gene KO experiments.

Using data from a set of gene KO experiments, TRaCE and its extension TRaCE+ generate the upper bound (most complex network) and the lower bound (least complex network) of an ensemble of networks consistent with the data. The regulatory interactions that belong to the upper bound but not to the lower bound are the uncertain interactions that could not be verified by prior data. The number of uncertain interactions represents the inferability of the network, with fewer uncertain edges indicating better inferability. TRaCE analyzes the inferability of the GRN only accounting for the existence of regulatory interactions, while TRaCE+ improves upon TRaCE by considering both the existence and mode of the regulation (activation/repression). In consequence, usually TRaCE+ can extract more information from the same data.

In the case studies we applied TRaCE to analyze the inferability of random GRNs and the GRNs of *Escherichia coli* and yeast from single- and double-gene KO experiments. The results showed that, with the exception of networks with very few interactions, GRNs are typically not inferable. Using realistic simulation data, we compared the performance of TRaCE with the top performing methods of DREAM 4, a community-wide network inference challenge. The results demonstrated that TRaCE performs better than the top performing state-of-the-art gene regulatory network inference algorithms.

Our design of experiments algorithm, REDUCE employs uncertain gene interactions that could not be verified by available data during ensemble inference by TRaCE or TRaCE+. REDUCE generates the optimal gene KO experiment by maximizing the number of uncertain interactions that can be verified by the resulting data. For this purpose, we introduced the
concept of edge separatoid which gave a list of nodes (genes) that upon their removal would allow the verification of a specific gene regulation. We also developed an iterative inference strategy using TRaCE (or TRaCE+) and REDUCE. The iterative strategy involves estimating the bounds of a GRN by TRaCE (or TRaCE+), design of experiments by REDUCE, performing the designed experiment and using the data to update the ensemble bounds. Importantly, this strategy can infer the true network from error-free data, thereby resolving the issue of GRN inferability. Even using noisy data, the iterative strategy can converge to a single network. The case studies including the inference of *E. coli* GRN and DREAM 4 100-gene GRNs, demonstrated the efficacy of the iterative GRN inference. In comparison to systematic KOs, REDUCE could provide much higher information return per gene KO experiment and consequently more accurate GRN estimates. In the case studies involving the inference of the DREAM 4 100-gene GRNs, the iterative strategy required fewer iterations and KO experiments when using TRaCE+ compared to using TRaCE. The iterative inference of GRNs using TRaCE(+) and REDUCE represents an enabling tool for tackling the underdetermined GRN inference. Along with advances in gene deletion and automation technology, the iterative procedure brings an efficient and fully automated GRN inference closer to reality.
Zusammenfassung


Die Identifizierbarkeit eines GRNs wird durch die kausale Information in einem Datensatz bestimmt, welche wiederum von den Experimenten, mit denen der Datensatz erzeugt wurde, abhängt. Daher ist es notwendig die durchgeführten Experimente zu berücksichtigen, um die Identifizierbarkeit des Netzwerkes mit Bezug auf die Experimente zu analysieren. Weiter sollte man die Experimente optimieren, welche den neuen Datensatz erzeugen, der wiederum die Identifizierbarkeit des Netzwerks verbessern würde. Wiederum ist die Aufmerksamkeit, welche dem Design von Experimenten zur Identifizierung des GRNs zuteil wird, im Vergleich zur Entwicklung von Inferenzmethoden sehr klein. Motiviert durch diese Kenntnislücke, haben

Aus einem Datensatz einer Serie von KO Experimenten erzeugen TRaCE und seine Erweiterung TraCE+ die obere (das komplexeste Netzwerk) und die untere Grenze (das einfachste Netzwerk) eines Ensembles der Netzwerke, welche konsistent mit dem Datensatz sind. Die regulatorischen Beziehungen, welche zur oberen aber nicht zu der unteren Grenze gehören, sind die unbestimmten Beziehungen, welche aus dem Datensatz nicht verifizierbar sind. Die Zahl der unbestimmten Beziehungen stellt die Identifizierbarkeit des Netzwerks dar, wobei wenige unbestimmte Beziehungen für eine bessere Identifizierbarkeit stehen. Während die Analyse der Identifizierbarkeit des GRNs durch TRaCE nur die Existenz der regulatorischen Beziehungen betrachtet, stellt TRaCE+ eine Verbesserung von TRaCE dar, indem es nicht nur die Existenz, sondern auch den Typ (Aktivierung/Hemmung) der Regulation betrachtet. Folglich kann TRaCE+ aus demselben Datensatz mehr Information gewinnen als TRaCE.

Die Ergebnisse zeigen, dass die Leistung von TRaCE besser als die Leistungen der stärksten bekannten GRN Inferenzalgorithmen ist.

Unser Algorithmus zum Design von Experimenten, REDUCE, verwendet die unbestimmten Beziehungen, welche aus den verfügbaren Daten durch TRaCE oder TRaCE+ nicht identifiziert werden konnten. REDUCE erzeugt das optimale Gen-KO Experiment durch Maximierung der Anzahl an unbestimmten Beziehungen, welche aus dem entstehenden Datensatz verifiziert werden kann. Zu diesem Zweck haben wir das Konzept des Kanten-Separatoids eingeführt, welcher eine Liste von Knoten (Genen) ergibt, durch deren KO eine spezifische Genregulation verifiziert werden kann. Wir haben ebenfalls eine iterative Inferenzstrategie, welche TRaCE (oder TRaCE+) und REDUCE verwendet, entwickelt. Diese iterative Strategie besteht aus der Schätzung der Grenzen eines GRNs durch TRaCE (oder TRaCE+), dem Design von Experimenten mittels REDUCE, der Durchführung der Experimente und der Anwendung der Daten zur Aktualisierung der Ensemblregelenzen. Es ist wesentlich, dass diese Strategie aus fehlerfreien Daten das echte Netzwerk identifizieren kann und somit das Problem der GRN Identifizierbarkeit löst. Sogar aus verrauschten Daten kann diese Strategie ein eindeutiges Netzwerk identifizieren. Die Fallstudien, einschließlich der Identifizierung des E. coli GRN und DREAM 4 100-Gen GRN, zeigen die Wirksamkeit dieser iterativen Strategie zur GRN Inferenz auf. Im Vergleich zu systematischen KOs, kann REDUCE deutlich mehr Information pro KO-Experiment generieren und damit genauere GRN erzeugen. In den Fallenstudien zur Inferenz von DREAM 4 100-gene GRN brauchte die iterative Strategie weniger Iterationen und KO-Experimente wenn TraCE+ statt TRaCE angewendet wurde. Die iterative Inferenz von GRN mit TRaCE(+) und REDUCE stellt ein Verfahren dar, welches es ermöglicht die Unterbestimmtheit der GRN-Inferenz zu bewältigen. Zusammen mit den Fortschritten in der Gen-Deletion und der Automatisierungstechnik, bringt das iterative Verfahren eine effiziente und vollständig automatisierte GRN Identifizierung der Realität näher.
Chapter 1 Introduction

The reductionist approach of studying the causes of cellular behavior and diseases has dominated biological research for more than half a century. The completion of the human genome project at the beginning of the 21st century [1, 2] and continued advances in high-throughput techniques give rise to a new appreciation that biological functions can rarely be attributed to one or a small number of genes. Instead cell’s characteristics (phenotypes) arise from the spatio-temporal interactions of a large number of diverse and frequently multifunctional cellular components [3, 4]. A cell phenotype can therefore be considered as an emergent property of a network of biological interactions [5]. A thorough comprehension of such emergent properties demands systems-level perspectives and cannot be accomplished from reductionist approaches [6]. Consequently, a systems approach in studying biological phenomena has become much more common in the previous decade. Systems biology was coined to describe such an approach, which entails an integration of computational modelling, system analysis, design of experiments, refinements of experimental techniques, and quantitative experiments (Fig. 1-1) in order to generate and iteratively refine mechanistic models of cellular functions [7].

Systems biology could profoundly influence pharmaceutical industry and medical practice. One of the most promising applications is to construct a detailed model of functional regulation of interest in cells, concentrating on key pathways and molecules, that is capable of generating system-level insights for network-based drug discovery. For example, such systems-level analysis may assist the recognition of cellular feedback mechanisms that compensate drug activity. Simulation studies using detailed models of cellular regulation may also predict systemic side effects and the combined effects of multiple drugs, allowing the optimization of the combination of several drugs for treating specific disorders while minimizing side effects.
Such systemic response cannot be reasonably estimated without a mechanistic model of intracellular genetic and biochemical interactions [7]. Numerous networks of interactions arise from putting together the interactions among various biomolecules [3].

Figure 1-1 Systems biology approach in biological knowledge discovery. Figure was adapted from [4].

Many biological functions of a cell are controlled by biological networks [3]. In this regard, many diseases can be explained as a consequence of dysfunctional networks [8]. The related research areas, known respectively as network biology and network medicine, are geared toward a better understanding of biology and pathology by analyzing the network of interactions among biomolecules. The knowledge of healthy and diseased networks may lead to novel diagnostics procedures and biomarkers, as well as facilitate drug development [5]. Moreover, network-based pharmacological interventions may be tailored towards preventing disease-mediated transitions [6]. In this regard, elucidating the regulatory pathways, through which different genes control each other, represents a key step toward these goals.
1.1 Gene Regulatory Networks

The term regulation in a biological system describes the process by which one molecular species controls the abundance or activity of another molecule. A gene regulatory network (GRN) describes how the product(s) of a gene regulate the expression of other genes. For example, the gene Nkx2-5 is part of a regulatory network that integrates numerous signals and controls the expression of several target genes, which are important in the development of heart (Fig. 1-2). Multiple types of congenital heart defects have been attributed to mutations in NKx2-5 which disrupt this network [9]. Below we review some concepts of molecular biology relevant to GRNs.

![Gene regulatory network](image)

**Figure 1-2** Gene regulatory network involved in the cardiac tissue development. Solid lines indicate direct transcriptional regulations, and dotted lines indicate regulations not yet shown to be direct. Figure is reproduced with permission from [9] and includes information from [10].
1.1.1 Central Dogma of Molecular Biology

The central dogma of molecular biology, first proposed by Francis Crick in 1956, describes the directional flow of genetic information encoded in a sequence of deoxyribonucleic acid (DNA) to ribonucleic acid (RNA) and protein molecules through the processes of transcription and translation processes [11]. As shown in Fig.1-3, during cell division, the genetic information can be copied into new DNA molecules through the process of replication. Meanwhile, the genetic information can also be propagated into protein in three steps. In the first step, DNA is transcribed into RNA. The second step involves the translation of RNA into polypeptide chain (the translated RNA molecules are known as messenger RNA or mRNA). The polypeptide chain then folds into a biochemically active protein in the third step. Additionally, two special cases of information transfers are known in viruses, including reverse transcription of RNA into DNA, and RNA-dependent RNA replication. Regulatory molecules, usually proteins and less frequently RNA or small metabolites, can control (regulate) the synthesis and the form of the product from each step of the gene expression process [12].

Figure 1-3 Central dogma of molecular biology. The sequence information contained in DNA is expressed in three steps: transcription into RNA from DNA, translation of RNA into polypeptide chain, and folding of polypeptide chain into protein. The red arrows indicate the special cases of reverse transcription and RNA replication that are found in viruses. The figure was adapted from [12].
1.1.2 Transcriptional Regulation

Transcriptional regulation describes the control of the gene transcription into RNA in a cell. The synthesis of RNA is carried out by the enzyme RNA polymerase (RNAP). Briefly, RNAP recognizes and binds to a control sequence called a promoter, and thereby starts the gene transcription. The promoter sequence is located upstream of a particular gene, in a region called the promoter region (Fig. 1-4(a)). RNAP complex unzips the DNA double helix and transcribes the DNA by elongating the RNA transcript through adding ribonucleotides complementary to the DNA sequence, until it finds a stop sequence. Any molecules that interact with the enzyme complex can modulate the rate and specificity of binding of the RNAP complex, as well as, the rate of transcription, and termination of transcription [12].

Proteins that do not directly participate in the formation of the RNAP complex can also alter the activity of this enzyme complex. These auxiliary proteins are known as transcription factors (TF). TFs bind to unique DNA sequence motifs close to or inside the promoter region. A TF can bind near a promoter region and increase the binding rate of RNAP to the promoter (Fig. 1-4(b)). A TF can also recruit the polymerase to unusual promoters [12]. As a result, the gene next to the promoter is transcribed at a higher level than it would have been in absence of the TF. This type of TF is known as an activating TF or an activator, and the regulation is known as activation, upregulation or positive regulation. In contrast, another TF can also bind to a motif in the promoter region, and block the binding of the RNAP enzyme (Fig. 1-4(c)). Consequently, the gene next to the promoter is expressed at a lower level than it would have been in absence of the TF. This type of TF is known as a repressing TF, a repressor or an inhibitor, and the regulation is known as repression, downregulation, negative regulation or inhibition.

TFs can regulate the expression levels of other TFs in addition to their direct target genes. The collection of such regulatory interactions creates a network known as gene or
transcriptional regulatory network. Fig.1-5 gives an example of a GRN. Here, TF A regulates TF B, which in turn regulates the expression level of its target gene C. In this case, TF A indirectly regulates gene C but directly controls genes B and D. The regulatory relationships among TFs and target genes in a GRN can be drawn as a directed graph, where the nodes represent the genes, and the directed edges (arrows) represent the gene regulations.

Figure 1-4 Illustration of transcriptional regulation. (a) RNAP complex binds to the promoter, and transcribes the DNA sequence. (b) Activating TF binds to a motif, and recruits RNAP (c) Repressing TF binds to a motif and blocks RNAP binding. Figure was drawn based on [12].

In addition to regulation through TFs, gene expression is also modulated by epigenetic regulations. Epigenetic regulations refer to stable alterations in gene expression potential, which arise during cell proliferation and development [13]. Such regulation is mostly mediated through the methylation of DNA molecules. Epigenetic regulation plays a key role in the differentiation and proliferation of various cell types in multicellular organisms [13]. However, the scope of this thesis does not include epigenetic regulation.
Figure 1-5 Example of a gene regulatory network. (a) The TF $A$ regulates the expression level of gene $B$. The transcript of gene $B$ is eventually translated into protein $B$ which is another TF. TF $B$ further regulates the expression level of gene $C$. Meanwhile, TF $A$ also regulates the expression of gene $D$. (b) Network graph representation of the transcriptional regulations in (a). Figure was adapted from [12].

1.2 Properties of Gene Regulatory Networks

Several properties have been observed about biological networks in general, and GRNs in particular [14]. These properties are related to sparsity, scale-free topology and robustness of GRNs.

1.2.1 Sparsity

One of the most important overt properties of GRNs is that they are sparse, where a single gene is usually regulated by a small number of genes. In other words, the average number of regulatory edges per gene is low [14]. For bacteria, the average number of regulators has been observed to be between 2 to 4 [15]. For eukaryotes, this number has been shown to be between 5 and 10 [16].
1.2.2 Scale-free topology

The number distribution of edges per node for biological networks tends to be longer tailed than a normal distribution [17]. It has therefore been proposed that the distribution of the number of edges per node follows a power-law distribution [18], where the fraction of nodes in the network that are connected with \( k \) other nodes is given by: \( P(k) = k^{-\gamma} \) with \( \gamma \) being a network-specific constant. Networks with this kind of connectivity distribution are known as scale-free networks. It has also been proposed that evolutionary selection leads to scale-free structures in naturally emerging networks [19]. In scale-free networks, several properties emerge due to their power-law connectivity distribution. First of all, while the majority of the nodes have very few partners, a few nodes, known as hub nodes, have a large number of edges. These hubs are central to the function of GRNs as they interact many other genes [14]. Next, due to the existence of hub genes, genes are connected to each other by a short path, i.e. the degree of separation between any two genes is small. Therefore, scale-free networks can often lead to small-world networks.

1.2.3 Robustness

Robustness is the ability of a system to conserve its functions in the face of external and internal uncertainty and perturbations [20, 21]. GRNs are known to be very robust to perturbations such as variable environment and mutations [14, 21]. It has been suggested that such robustness arises from specific topology of the GRN [17, 22]. For example, scale-free GRNs have structural robustness against random perturbations since most of the nodes in the network are connected to only a few other nodes, and thus their failure (removal) would not greatly affect the structure of the GRN [3]. However, these networks are known to be susceptible (fragile) to perturbations that affect the hub nodes [3]. In addition to robustness against perturbation of different nodes, GRNs are also insensitive to changes in concentration.
of different molecules, especially during development of animals, in order to ensure that the
same anatomy is produced under various metabolic conditions [23]. Taken together, GRNs
are structurally robust against random perturbation of nodes as well as functionally robust
against variation in the concentrations of different biomolecules.

1.3 Gene Expression Data

There exist several assays for measuring the level of gene transcription in cells. Northern
blotting, the early method of detecting gene expression [24], did not provide accurate
quantitative information. Subsequent development of Reverse Transcriptase Polymerase Chain
Reaction (RT-PCR) permitted quantification of the expression levels of specific genes.
However, this assay was not suitable for parallel quantification of the expression levels of a
large number of genes. In the last decade, transcriptional expression data predominantly come
from DNA microarrays. More recently, Next Generation Sequencing (NGS) technology,
specifically RNA sequencing (RNA-seq), has become more common. This technology can
provide more sensitive quantification of low-level gene transcription.

1.3.1 Northern Blot

Northern blotting involves isolation and separation of RNA according to size using gel
electrophoresis followed by transfer of the RNA from the gel to a membrane. Then the
membrane is treated with radioactive or fluorescent labeled DNA or RNA probe which can
bind specifically to a target RNA. Following removal of unbound probe, the presence or
absence of the target RNA can be established from the presence or absence of radioactivity or
fluorescence [24]. However, Northern blotting does not provide reliable quantitative
information about the expression level, and can only detect one target gene in an experiment.
1.3.2 RT-PCR

At the beginning of the last decade, RT-PCR was the most sensitive technique available for detecting and quantifying gene expression. RT-PCR can quantify mRNA as low as a few hundred copies per sample. The procedure comprises two steps: the first step involves the synthesis of complementary DNA (cDNA) from mRNA by reverse transcription, and the second step involves the amplification of a target cDNA using PCR. Compared to Northern blotting, RT-PCR is more sensitive and tolerant to degraded RNA. However, quantitative comparison between different samples is difficult because small variations in amplification efficiencies can lead to substantial differences in product yield [25]. Additionally, RT-PCR can quantify only one target gene per experiment.

1.3.3 Microarray

Microarrays or gene-chips have become the most common platform for measuring gene transcriptional expression level. On a microarray, there is an array of spots, each containing picomolar quantity of a DNA probe which can specifically bind to a complementary sequence [26], e.g. cDNA of a certain mRNA transcript. Similar to the first step of RT-PCR, mRNA extracted from the sample are converted to cDNA using reverse transcriptase enzyme. Then the cDNA are labeled with fluorescent dye and the labeled cDNA are applied on the microarray, allowing it to hybridize. Following the removal of unbound cDNA, fluorescence intensity at each spot is measured. The intensity readings are normalized in order to obtain relative expression levels of different genes [27]. Because of their small volumetric requirements, a single microarray chip can contain thousands of spots and therefore give measurements of gene expression levels for the whole genome of a species.

In addition to variability during sample preparation, microarray expression data are also affected by hybridization dynamics, saturation of fluorescent molecules and saturation of
fluorescence detectors. For example, when some genes are expressed at high levels, the cDNA molecules can produce saturating fluorescence intensity at their corresponding spots. In other words, the intensity readings are longer proportional to the mRNA concentrations beyond a certain threshold. On the other hand, if a gene is expressed at very low level, the hybridization of cDNA to the appropriate probe can be stochastic. As a result, genes expressed at low levels are more prone to noise. The experimental noise in microarray experiments have been suggested to follow a log-normal distribution [28].

1.3.4 Next Generation Sequencing

Next Generation Sequencing data depend on the sequencing technology instead of DNA hybridization. Whole Transcriptome Shotgun Sequencing (WTSS), commonly known as RNA-seq is the NGS technology used for analyzing gene expression [29]. In RNA-seq experiments, isolated RNA are first hydrolyzed into short RNA fragments using chemical or enzymatic hydrolysis. Then the RNA fragments of appropriate length (e.g. 200-250 nucleotides) are converted into cDNA using reverse transcriptase Adapter oligonucleotides are subsequently ligated to the cDNA to permit amplification and sequencing, and the fragments are sequenced. Finally, the sequence data are further analyzed by mapping the short reads to a transcriptome database, and the relative abundance of different transcripts in the data are calculated [30]. Compared to microarray technology, RNA-seq is more flexible because one does not need a complete set probes for the genome. RNA-seq is also more sensitive than microarray in quantifying gene expression. However, RNA-seq data require more processing, and eventually more powerful computational facilities.

1.3.5 Differential Expression

Differential expression analysis is one of the fundamental techniques to determine the effects of a perturbation on gene expression. A gene is said to be differentially expressed
between two different experimental conditions, if its expression level under one condition is statistically significantly different from that under the other condition. However, analyzing differential expression is often complicated by the nature of the transcriptional expression data. As mentioned earlier, if a certain gene is expressed at a high level beyond the saturation threshold of microarray fluorescence, any further increase in its expression level caused by a perturbation could not be detected. In contrast, owing to disproportionately high noise in genes with low expression level, it would be very difficult to quantify any further decrease in the expression level of such genes after a perturbation.

1.4 Inference of Gene Regulatory Networks

Inference of GRNs refers to the identification of the regulatory relationships among a set of genes. Different strategies exist for inferring GRNs which can be divided into two main categories depending on the source data. In the first strategy, one measures the gene expression levels under different conditions, and identifies causal relationships among different genes using appropriate computational methods [12, 31]. In the second strategy, one employs DNA sequence information to identify TF binding motifs, providing information of the TF-target gene relationship in the GRN [12, 32]. Below we will briefly review prominent GRN inference (GRNI) algorithms in both categories.

1.4.1 Inference from Gene Expression Data

A large number of methods have been developed to infer GRNs from gene expression data [31, 33, 34]. These methods can be grouped based on their working principles. Below we describe some of the common groups of GRNI methods. Other reviews of GRNI methods and the comparison of their performance can be found in [31, 33, 34].
Correlation-based Methods

Correlation-based methods are based on the assumption that a correlation in the expression levels between two genes indicates the presence of a regulatory relationship. Different types of correlation metrics have been used, such as Pearson correlation, Spearman’s Rank correlation and Kendall’s τ analysis of the gene expression data [31]. Since correlation does not necessarily indicate causation, correlation-based methods are ineffective in identifying causal directions in the gene-gene interactions.

Information Theory-based Methods

GRNI methods based on information theory commonly employs mutual information to predict interactions among genes. A few notable examples of these methods include Relevance Networks (RN) [35], Context Likelihood of Relatedness (CLR) [36], Algorithm for the Reconstruction of Accurate Cellular Networks (ARACNE) [37], Partial Correlation and Information Theory (PCIT) [38], MRNET [39], and MRNET-B [40]. For example, RN uses the mutual information between two genes, i.e. mutual dependence between the expression data from two genes, as a likelihood measure of their regulatory interaction [35]. Similar to RN, CLR also applies mutual information for finding regulatory relations. However, CLR extends RN by taking the background distribution of mutual information into account, and assumes that the gene pairs whose mutual information deviates more from the background distribution of mutual information are more likely to have regulatory interaction [36]. Meanwhile, ARACNE, the most prominent among mutual information-based algorithms, further applies the Data Processing Inequality (DPI) in order to differentiate between direct and indirect regulations [37]. Similar to ARACNE, PCIT also applies DPI, but to partial correlation [38]. Finally, MRNET [39] and MRNET-B [40] apply a feature selection algorithm (particularly, maximum relevance minimum redundancy [41]) on the mutual information among gene expression profiles. Briefly, these methods select the best subset of regulators with
the maximal relevance with each target gene, while at the same time have maximal pairwise independence among themselves. The difference between the two is the use of forward selection in MRNET and backward elimination in MRNET-B in the feature selection.

Information theory-based methods can deal with data from both observational studies and targeted perturbation studies, such as gene knock-outs. However, these methods do not take the information regarding the types of experiments into consideration, and thus, they often perform worse than perturbation graph based methods when targeted perturbation experiments are available. Moreover, similar to correlation-based methods, information theory-based methods are also ineffective in identifying causal directions.

Regression-based Methods

Regression-based methods establish regulatory relation by regressing the expression level of a (target) gene on the expression levels of other (regulator/TF) genes. One of the important regression-based methods for GRNI is GEne Network Inference with Ensemble of trees (GENIE3) \[42\]. GENIE3 transforms the GRNI problem into a feature selection problem using an ensemble of classification and regression trees \[43, 44\]. GENIE3 was among the best performing algorithms in a community-wide challenge on GRNI using data from multifactorial perturbations of the network. Another regression-based method, Trustful Inference of Gene REgulation with Stability Selection (TIGRESS) also transforms the GRNI problem into a variable selection problem. More specifically, TIGRESS \[45\] uses Least Angle Regression (LARS) \[46\] coupled with stability selection to solve the variable selection problem. LARS is a model selection algorithm, similar to forward selection strategy, for creating sparse linear regression models. For each gene \(j\), TIGRESS employs LARS to a linear regression model where the expression level of a target gene is described as a linear combination of the expression levels of TF genes. Starting from a model with no regressor, LARS iteratively adds a regressor (a regulator gene) to refine the prediction of the expression level of gene \(j\). Unlike
forward selection, the addition of a regressor is based on equi-angular criterion with respect to the residuals.

Similar to information theory-based methods, regression-based methods can deal with data from both observational studies and targeted perturbation studies, and again do not take the information of the experiments into account. While regression-based methods can provide causal directions of the gene regulations, they have difficulties in differentiating between direct and indirect regulations.

**Support Vector Machine-based Method**

Support Vector Machine (SVM) is a popular supervised learning algorithm for classification and regression. An important SVM-based GRNI method is SIRENE (Supervised Inference of Regulatory Networks) [47]. SIRENE requires prior information on the set of target genes (and preferably also the set of non-target genes) of each TF in the GRN. In particular, SIRENE uses SVM to train predictors for local models associated with each TF. In this manner, for each TF, SIRENE trains a binary classifier to distinguish between target and non-target genes of a TF, and applies this classifier for identifying genes regulated by the TF [47].

Similar to regression-based and information theory-based methods, SIRENE can also deal with data from both observational and targeted perturbation studies. Additionally, the SIRENE is trained to identify the genes regulated by a TF, as a result the causal direction is automatically known to be from the TF to the target gene. Nevertheless, the requirement of prior information can limit the application of SIRENE.

**Perturbation Graph-based Methods**

Perturbation graphs are a directed graph generated from data of perturbation experiments. For instance, if upon perturbing (knocking-out) a certain gene $i$, the expression level of another gene $j$ changes significantly, then the gene $j$ is considered to be regulated by the gene $i$ [48].
Here, the causal directions are immediately obtained from the perturbation information. In many methods within this class, the perturbation graph is subsequently pruned to give a parsimonious GRN, following the overt sparsity of biological networks (see Section 1.2.1). For example, Wagner [49] employed a transitive reduction algorithm to obtain the smallest network (fewest regulatory edges) that retains the same transitive relationships among genes as in the original perturbation graph. The down-ranking of feed-forward loops algorithm by Pinna et al. [48], TRANSitive reduction for WEighted Signed Digraphs (TRANSWESD) [50] and Local Transitive Reduction (LTR) [51] also use transitive reduction strategy. TRANSWESD and LTR further take the sign and weight of the regulatory edges into account during the transitive reduction. Down-ranking and TRANSWESD were among the top three best performing algorithms in a community-wide challenge on GRNI using data from targeted perturbations of the network.

Perturbation graph-based methods perform better than information theory-based, regression-based and correlation-based methods for data from targeted perturbation experiments. However, perturbation graph-based methods cannot handle data from observational studies in which the precise targets of the perturbations are unknown.

**ODE-based methods**

Ordinary Differential Equations (ODE)-based methods are based on modeling the GRN using a system of ODEs. The GRNI comprises estimating model parameters from the expression data. Prominent ODE-based methods include Network Identification by multiple Regression (NIR) [52, 53], Microarray Network Identification (MNI) [54], and Time Series Network Identification (TSNI) [55]. NARROMI is a hybrid method that combines mutual information and ODE-based recursive optimization [56]. ODE-based methods can utilize data from both perturbation experiments and observational studies in order to estimate model parameters. In addition, the parameters of ODE models already models casual direction. The
drawback of ODE-based methods is the need to identify a large number of parameters from expression data, a challenging problem due to high numerical requirement and parameter identifiability issue.

**Bayesian Networks-based methods**

Methods in this class use Bayesian Networks (BN), a probabilistic graphical model based on conditional dependencies among gene expressions [57, 58]. One of the advantages of BN is their ability to infer regulatory relations from observational data. The disadvantages of BN include their inability to deal with feedback cycles in the GRN and high computational cost when dealing with many variables. An extension of BN, called Dynamic Bayesian Networks (DBN) can handle cycles of regulatory interactions, as well as make use of time series expression data for the purpose of GRNI.

**1.4.2 Inference from DNA sequence**

Prediction of TF binding sites can provide opportunities for *ab initio* inference of GRN from DNA sequence information. In this approach, if a TF is predicted to bind to the promoter region of a gene, the gene is considered to be regulated by the TF. There have been several genome-wide studies that map the binding sites of TFs. The DNA binding specificities of human TFs have been studied using high-throughput methods [59]. Furthermore, the transcription start sites in humans and mice have been mapped by the FANTOM project [60, 61]. Additionally, a genome-wide study has been carried out to discover binding sites for human transcription factors [62]. These studies along with computational methods for predicting TF binding sites [32] will significantly improve the accuracy of TF binding-site prediction, and eventually the inference of GRNs from DNA sequence.
1.5 Scope of the Thesis

Inferring GRNs have received tremendous attention in recent years and a deluge of inference algorithms have been developed [12, 31, 33, 36, 53, 63]. Consequently, there is a need for fair comparison among these methods. The Dialogue for Reverse Engineering Assessments and Methods (DREAM) challenges [64] 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 [65-68]. Specifically, the inference of GRN was a major focus of DREAM2, DREAM3, DREAM4 and DREAM5 challenges. The outcomes of these challenges indicate that the state-of-the-art algorithms for GRN inference, some of which have been mentioned above, 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) [66-68]. More precisely, there are systemic errors in predicting fan-out, fan-in and cascade motifs. In a fan-out motif, a single gene regulates two or more genes. Some algorithms however would find one or more spurious regulatory interactions between the target genes (e.g. a falsely identified regulatory relation from B to C when A regulates both). Meanwhile, in a fan-in motif, two or more genes simultaneously regulate the same target gene. Many algorithms however could identify only one of the regulators. In a cascade motif, a certain gene A regulates another gene B, which in turn regulates gene C, leading to indirect regulation of C by A. But, most algorithms would identify a spurious direct regulation of gene C by gene A [68]. The failure of most algorithms in correctly identifying these motifs has been attributed to the limited data which may leave the inference problem underdetermined [68, 69].

Whether or not the 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 i.e. the problem does not have sufficient information to obtain a unique solution. 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 [70-73]. 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 [74-77]. 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.

The goal of this thesis is to address the aforementioned issue of identifiability or inferability of GRNs within the scope of inference from steady-state data of gene KO experiments. The objectives of this thesis are to analyze the inferability of a GRN from a given set of KO experiments, and to design the most informative experiments in order to improve the inferability.

In Chapter 2 of this thesis, we introduced a new theoretical framework and computational algorithms, called Transitive Reduction and Closure Ensemble (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. Following the network inference formulation in DREAM challenges, we initially ignored the signs of the edges and sought only to establish the existence of gene regulations in a GRN digraph. 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. Using TRaCE, we demonstrated that systematic KO experiments, the most common perturbations such as performing single- and double-gene KOs is a strongly suboptimal strategy for inferring GRN.

In consequence, Design of Experiments (DOE) for inferring GRNs is crucial. To this end, in Chapter 3 of this thesis, we developed an algorithm called REDuction of UnCertain Edges (REDUCE) for selecting the optimal gene KO experiment based on an ensemble of GRNs, particularly using the upper and lower bounds of the ensemble produced by TRaCE. REDUCE was formulated as a constrained optimization problem to maximize the number of uncertain edges that could potentially be verified. For this purpose, we introduced the concept of edge separatoid, as the basis to count the number of possible edge verification associated with a given gene KO combination. Finally, we proposed an iterative procedure for the GRN inference, in which the upper and lower bounds of the ensemble are continually updated with every iteration of wet-lab KO experiments and dry-lab optimal DOE using REDUCE. As a proof of concept, we applied the iterative procedure to infer the GRN of *E. coli* under ideal conditions. We further demonstrated the efficacy of REDUCE using benchmark gene expression simulator GeneNetWeaver [78] in the inference of five 100-gene gold standard networks from DREAM 4 *in silico* network inference challenge [78].

The edge signs in a GRN digraph are often of great interest and significance, as they indicate the modes of the gene regulations. A positive edge here reflects an activation, while a negative edge describes a repression. Several notable network inference algorithms such as TRANSWESD [50] and LTR [51] previously considered the inference of GRN digraph with signed (and weighted) edges. However, these algorithms were not designed for inferring an
ensemble of GRN structures. In Chapter 4, we addressed the aforementioned drawback of TRaCE by developing a new ensemble inference method, called TRaCE+. The new method uses a signed digraph model of the GRN, i.e. the edges have signs. Like TRaCE, TRaCE+ generates the upper and lower bounds of the ensemble, but in the form of signed digraphs. The ensemble bounds from TRaCE+ are also compatible with REDUCE [79]. We demonstrated the advantages of TRaCE+ over TRaCE in the ensemble inference of Escherichia coli GRN and in the iterative inference of 100-gene gold standard GRNs from DREAM 4 in silico network inference challenge.
Chapter 2 TRaCE

2.1 Introduction

The GRN inference has been stated to be underdetermined implying that there could be many equivalent (indistinguishable) solutions. Motivated by this fundamental limitation, in this chapter, we introduce a new theoretical framework and computational algorithms, called Transitive Reduction and Closure Ensemble (TRaCE), for ensemble inference and inferability analysis of GRNs. Using data from a set of gene KO experiments, TRaCE produces the lower and upper bounds of an ensemble of GRNs consistent with the data, where the distance between the bounds reflects the uncertainty about the GRN inference. Consequently, the bounds can be used to analyze the inferability of GRNs. Here, we have used TRaCE in two applications. First, we investigated the inferability of random GRNs and the GRNs of *E. coli* and *S. cerevisae* 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.

2.2 Method

2.2.1 Theoretical Foundation

2.2.1.1 Definitions

Here, we provide a short synopsis of basic concepts in graph theory that are necessary for the development of our algorithms. 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

---

1 The content of this chapter is adapted from the peer-reviewed publication: Ud-Dean SM, Gunawan R. Ensemble inference and inferability of gene regulatory networks. PloS one. 2014 Aug 5;9(8):e103812. © The authors. The introduction and discussion parts have been modified, and the conclusion part has been merged with Chapter 5 in order to put it into the context of this thesis.
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 \(\text{Adj}(G)\), is an \(n \times n\) matrix with \(\text{Adj}_{i,j} = 1\) when \((i, j) \in E(G)\), and \(\text{Adj}_{i,j} = 0\) 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 [49].

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 $\overline{G}$, is the smallest transitive supergraph of $G$ (with the fewest edges) [80]. When $G$ is a DAG, we denote the transitive closure of $G$ as $G'$. As shown in Fig. 2-1(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. $Acc(G) = Adj(\overline{G})$. For a digraph $G$, the set of digraphs that have the same transitive closure $\overline{G}$ is denoted by $S(G) = \{G_i : \overline{G}_i = \overline{G}\}$. The transitive reduction of $G$, denoted by $G'$, 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 $G' = \bigcap_{G_i \in S(G)} G_i$ [80]. An algorithm for obtaining transitive reduction has been previously developed [49], 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. 2-1(c)). Note that the transitive reduction of a digraph with cycles is not unique.

![Diagram](image)

Figure 2-1 (a) An example of a directed graph $G_\varnothing$. (b) The transitive closure $\bar{G}_\varnothing$ (in this case, $\bar{G}_\varnothing = G_\varnothing^L$ since $G_\varnothing$ is a DAG). (c) The transitive reduction $G_\varnothing^t$ of $G_\varnothing$. (d) The directed graph $G_{(\beta)}$ associated with $G_\varnothing$. (e) The transitive closure $\bar{G}_{(\beta)}$. In this case, the transitive reduction $G_{(\beta)}^t$ happens to be the graph $G_{(\beta)}$. (f) The ensemble upper bound $G^U$ obtained from $\bar{G}_\varnothing$ and $\bar{G}_{(\beta)}$. The ensemble lower bound $G^L$ obtained from $G_\varnothing^t$ and $G_{(\beta)}^t$ happens to be the graph $G_\varnothing$.

2.2.1.2 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 $G_\varnothing$. For any set of genes $V_{\text{ko}} \subset V(G_\varnothing)$, we use $G_{V_{\text{ko}}}$ to denote a subgraph of $G_\varnothing$ that results from removing all edges incident to the genes in $V_{\text{ko}}$. 

25
the set $V_{KO}$, $E_{G_G}(V_{KO})$. In other words, $G_{V_{KO}}$ is the digraph with $V(G_{V_{KO}})=V(G_G)$ and $E(G_{V_{KO}})=E(G_G)-E_{G_G}(V_{KO})$. For example, $G_{\{B\}}$ associated with $G_G$ in Fig. 2-1(a) is the graph with all edges incident to gene $B$ removed, as shown in Fig. 2-1(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 [63, 68], 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 $G_G$ described by the graph in Fig.2-1(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 [49]. 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 $G_G$ 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 $\overline{G}_G$. More specifically, the $j$-th element in the $i$-th row of $\overline{G}_G$ (i.e.
(\vec{G}_\mathcal{O})_{i,j}) is set to 1 when gene j is differentially expressed in the KO experiment of gene i. The other elements of \vec{G}_\mathcal{O} 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 \( G_{\text{\tiny{V_ko}}} \) for an appropriately chosen set of genes \( V_{\text{\tiny{KO}}} \). In principle, we can determine \( \vec{G}_{\text{\tiny{V_ko}}} \) from the complete set of experiments involving KOs of the genes in the set \( V_{\text{\tiny{KO}}} \) and an additional gene \( i \) for all \( i \in \{V(\vec{G}_\mathcal{O})-V_{\text{\tiny{KO}}}\} \). These experiments are equivalent to performing single-gene KOs of the GRN \( G_{\text{\tiny{V_ko}}} \), and therefore \( \vec{G}_{\text{\tiny{V_ko}}} \) can be obtained by following the same procedure as that for \( \vec{G}_\mathcal{O} \) above. As an illustration, consider the GRN in Fig. 2-1(a) with \( V_{\text{\tiny{KO}}} = \{B\} \). The graph \( G_{\{B\}} \) is given in Fig. 2-1(d). In this case, we can construct the accessibility matrix \( \vec{G}_{\{B\}} \) 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. 2-1(e).

We can generalize the simple example above to any set of genes \( V_{\text{\tiny{KO}}} \) that could be derived from the available multi-gene KO experiments. More specifically, we set \((\vec{G}_{\text{\tiny{V_ko}}})_{i,j}\) to 1 when knocking-out \( \{i,V_{\text{\tiny{KO}}}\} \) 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 $\bar{G}_{v_{ko}}^k$ 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 $\bar{G}_{v_{ko}} = \bar{G}_{\{k\}}$ 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 $\bar{G}_{v_{ko}}^k$'s, which is the set:

$$\mathcal{A} = \left\{ G' : \bar{G}' = \bar{G}_\emptyset \text{ and } \bar{G}_{v_{ko}}^k = \bar{G}_{v_{ko}}^k, k = 1, 2, \ldots, K \right\}$$

Eq. 2-1

where $G_{v_{ko}}^k$ is the digraph with $V(G_{v_{ko}}^k) = V(G')$ and $E(G_{v_{ko}}^k) = E(G') - E_{G'}(V_{ko}^k)$. 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 $G_{v_{ko}}^T$'s (i.e. $G_{v_{ko}}^{T}$ and $G_{v_{ko}}^{T}$) and their transitive reductions (i.e. $G_{v_{ko}}'$ and $G_{v_{ko}}'$), using the following equations (for details see the Numerical Implementation section):

\[
G^L = G_\emptyset \cup \bigcup_{k=1}^{K} G_{v_{ko}}^k
\]  
Eq. 2-2

\[
G^U = G_\emptyset \cap \bigcap_{k=1}^{K} \left( G_{v_{ko}}^T \cup E_{G_{v_{ko}}}^{k} (V_{ko}^k) \right)
\]  
Eq. 2-3

where $G_{v_{ko}}^T \cup E_{G_{v_{ko}}}^{k} (V_{ko}^k)$ denotes the digraph with vertices $V(G_{v_{ko}}^k)$ and edges $E(G_{v_{ko}}^T) \cup E_{G_{v_{ko}}}^{k} (V_{ko}^k)$. Without any $G_{v_{ko}}^T$, the upper bound of the ensemble is simply given by the accessibility matrix $G_\emptyset^T$ and the lower bound is the transitive reduction $G_\emptyset'$. As $G_{v_{ko}}^k$ is a subgraph of $G_\emptyset$, the transitive closure $G_{v_{ko}}^T$ is also a subgraph of $G_\emptyset^T$. In Eq. 2-3, the upper bound is constructed starting from $G_{v_{ko}}^T$ in which edges are removed based on $G_{v_{ko}}^T$. Here, edges incident to $V_{ko}^k$ are not altered during the intersection of the accessibility matrix $G_{v_{ko}}^T$. For example, consider again the GRN in Fig. 2-1(a) with the accessibility matrices $G_{v_{ko}}^T$ and $G_{v_{ko}}^T$ in Figs. 2-1(b) and 2-1(c). 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. 2-1(f)). Thus, the size of the upper bound is generally reduced with the incorporation of $G_{v_{ko}}$'s. On the other hand, according to Eq. 2-2, the lower bound becomes larger with the inclusion of
every available \( G'_{v_{K_0}} \). In the same example above, the transitive reduction of \( G^T_{(B)} \) happens to be the graph \( G_{(B)} \) (i.e. in this case \( G^T_{(B)} = G_{(B)} \)). Here, the combination of \( G^T_{(B)} \) and \( G^T_{v_{K_0}} \) in Figs. 2-1(c) and 2-1(d), respectively, gives the lower bound \( G^L \) that is equal to \( G_{(B)} \). However, in general, \( G^L \) is a subgraph of \( G_{(B)} \).

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

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

\[
G^L \subseteq G \subseteq G^U \quad \forall \ G \in A
\]

**Proof of \( G^L \subseteq G \forall \ G \in A \):** For any edge \( e \in E \left( G^L \right) \), Eq. 2-2 implies that either \( e \in E \left( G'_{(B)} \right) \) or \( e \in E \left( G'_{v_{K_0}} \right) \). Therefore, we have either:

- \( e \in E \left( G'_{(B)} \right) = E \left( G' \right) \subseteq E \left( G \right) \quad \forall \ G \in A \), or

- \( e \in E \left( G'_{v_{K_0}} \right) = E \left( G - E_G \left( V_{K_0}^k \right) \right) \subseteq E \left( G \right) \quad \forall \ G \in A \).

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

- \( e \in E \left( G - E_G \left( V_{K_0}^k \right) \right) \subseteq E \left( \left( G - E_G \left( V_{K_0}^k \right) \right)^T \right) = E \left( G_{v_{K_0}}^T \right) \), or

- \( e \in E_G \left( V_{K_0}^k \right) \subseteq E_G \left( V_{K_0}^{k_2} \right) = E_G \left( V_{K_0}^{k_2} \right) \).

Therefore, \( e \in E \left( G^U \right) \).
Remark: Since $G_{\emptyset}$ is a member of $A$, $G^{l}$ 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 $A$ can be obtained by combinatorially adding edges in the set $(G^{u} - G^{l})$ to $G^{l}$. Therefore, the dimension of $A$ 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. 2-3 with $G$ 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 [49], Wagner proposed a procedure in which digraphs are first condensed into DAGs before constructing the transitive reduction [49]. 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 Condensation, Transitive Reduction and Expansion (ConTREx). The ConTREx of an accessibility matrix $G$, denoted by $G_{\emptyset}$, may no longer be a valid transitive reduction (i.e. $G^{T}$ may not necessarily be equal to $G_{\emptyset}$). Nonetheless, the lower bound constructed using Eq. 2-2 with $G$’s replacing $G^{*}$’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 $A$ (see Fig. A1 in appendix A). 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. A2 in appendix A).

2.2.1.3 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. 2-2 and 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 $G_{\emptyset}$ ($e \notin E(G_{\emptyset})$) 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 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 erroneously missing only from $\tilde{G}^U$ ($e \in E(\tilde{G}^L \cap \tilde{G}^U)^c$, where $c$ denote the complement of a set). Here, the bounds become inconsistent with each other (i.e. $\tilde{G}^L \nsubseteq \tilde{G}^U$). 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 2-1 illustrates the three types of errors mentioned above.
<table>
<thead>
<tr>
<th>Error</th>
<th>Type A</th>
<th>Type B</th>
<th>Type C</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{Adj}(G_{\emptyset})_{i,j}$</td>
<td>0</td>
<td>1</td>
<td>0 or 1</td>
</tr>
<tr>
<td>$\text{Adj}(	ilde{G}^U)_{i,j}$</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$\text{Adj}(	ilde{G}^L)_{i,j}$</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2-1 Types of Errors in $\tilde{G}^L$ and $\tilde{G}^U$.

A closer scrutiny of Eqs. 2-2 and 2-3 reveals that errors from the input accessibility matrices are passed on and compounded in the bounds. For example, FN errors in $\tilde{G}_{\emptyset}$ or any $\tilde{G}_{V^k_{\text{ko}}}$ will end up in $\tilde{G}^U$, while FPs in $G_{\emptyset}$ and any $G_{V^k_{\text{ko}}}$ 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 $\tilde{G}_{V^k_{\text{ko}}}$ 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 $\tilde{G}_{V^k_{\text{ko}}}$ are any edge $(i,j) \in E(\overline{G}_{\emptyset})$ with $i,j \notin V^k_{\text{ko}}$, such that there exists a directed path from $i$ to $j$ involving one or more genes in $V^k_{\text{ko}}$. As directed paths involving genes in $V^k_{\text{ko}}$ are disconnected in $G_{V^k_{\text{ko}}}$, we can potentially verify the existence of the testable edges from $\tilde{G}_{V^k_{\text{ko}}}$ and $\overline{G}_{V^k_{\text{ko}}}$. For example, the existence of the edge $(A,D)$ in Fig. 2-1(a) can be verified using the transitive reduction of $\tilde{G}_{i_{\{B\}}}$, which is the graph shown in Fig. 2-1(d). Meanwhile, we can establish the absence of the edge $(A,C)$ in $G_{\emptyset}$ using the accessibility matrix $\overline{G}_{i_{\{B\}}}$ (see Fig. 2-1 (e)). The number of testable edges can also be used to estimate the information content of a $\tilde{G}_{V^k_{\text{ko}}}$, where a higher number of testable edges indicates more informative $\tilde{G}_{V^k_{\text{ko}}}$.
For a given $V_{ko}$, it is straightforward to show that the testable edges are the non-zero entries of the testability matrix:

$$T_{ko} = \sum_{i \in V_{ko}} (\bar{G}_\varnothing)^i \otimes (\bar{G}_\varnothing)_i,$$

Eq. 2-4

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

### 2.2.2 Numerical Implementation

The pseudo-codes and the MATLAB implementations of TRaCE are provided in the appendix A 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.
2.2.2.1 Constructing Accessibility Matrices from Expression Data

In the case studies, we have employed the Z-score transformation for differential gene expression analysis [81]. Without loss of generality, we describe below the procedure for constructing $\overline{G}_{v_{\text{ko}}}$ from the complete set of single-gene KOs of $G_{v_{\text{ko}}}$, i.e. all possible combinations of $\{i, V_{\text{KO}}\}$ 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.

**Figure 2-2** Construction of accessibility matrix $\overline{G}_{v_{\text{ko}}}$ from expression data. The data come from KOs of genes in the set $V_{\text{KO}}$ and an additional gene $i$, $\forall i \in V(G) - V_{\text{KO}}$. 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. 2-5. (3) Subsequently, the $z$-score matrices are averaged over the technical replicates to give $Z_{G_{v_{\text{ko}}}}$. (4) The accessibility matrix $\overline{G}_{v_{\text{ko}}}$ is determined from $Z_{G_{v_{\text{ko}}}}$ based on a threshold criterion in Eq. 2-6.
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. 2-2.

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 $z_{\text{cutoff}}$ of the standard deviation. We subsequently recompute the sample mean and standard deviation $\mu_j$ and $s_j$ by excluding the data beyond $z_{\text{cutoff}}$. 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 $i, j \in V(G)$ according to [81]

$$
\begin{cases}
  g_{i,j} - \mu_j \\ 
  s_j
\end{cases} \quad \forall \, i, j \notin V_{KO}
\]

$$
\begin{cases}
  0
\end{cases} \quad \forall \, i, j \in V_{KO}
$$

where $g_{i,j}$ 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 $G_{v_{ko}}$ from $Z_{G_{v_{ko}}}$ using a threshold, as follows:
In our experience, $z_{\text{cutoff}} = 3$ and $z_{\text{threshold}} = 2$ provide reliable network ensembles. In general, choosing higher $z_{\text{cutoff}}$ and $z_{\text{threshold}}$ 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 $z_{\text{threshold}}$ between 1.5 and 2.5 and $z_{\text{cutoff}}$ between 2 and 3 (see section 2.3).

### 2.2.2.2 TRaCE without error correction

TRaCE without error correction is implemented as matrix-operations of Eqs. 2-2 and 2-3. 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 $V_{K_{V}}^{k}$. On the other hand, the transitive reduction is based on the algorithm by Wagner [49], which has been re-implemented using matrix operations. When there is no cycle in $\overline{G}_{V_{k}}$ and $\overline{G}_{k_{V}}$’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. 2-2. Cycles and genes involved in cycles can be detected from entries of $\text{Acc}(G_{V}) \times \text{Acc}(G_{V})^T$ [82]. For GRNs with cycles, the ConTREx procedure is applied to each available accessibility matrix, and the resulting $G$ matrices are again combined using binary additions to produce $G^L$. The schematic diagram of the error-free implementation is shown in Fig. 2-3(a).
Figure 2-3 Schematic diagrams of TRaCE with and without error correction. (a) Construction of the lower bound $G^L$ and upper bound $G^U$ from $\tilde{G}_{\emptyset}$ and $\tilde{G}_{V_{\text{ko}}}^L$'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 $\tilde{G}_{\emptyset}^M$ and $\tilde{G}_{V_{\text{ko}}}^M$'s using TRaCE with error correction. Expression data from gene KO experiments are converted into accessibility matrices $G^M_{\emptyset}$ and $G^M_{V_{\text{ko}}}$'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 $G^M_{\emptyset}$ and $G^M_{V_{\text{ko}}}$'s are created, denoted respectively by $\tilde{G}_{\emptyset}^M$ and $\tilde{G}_{V_{\text{ko}}}^M$'s, and the ConTREx of these closures are
evaluated. TRaCE with error correction begins with the preprocessing of $G^M$'s and $\tilde{G}$'s to produce the corrected matrices $\left\{\tilde{G}^M_\varnothing\right\}$ and $\left\{\tilde{G}_\varnothing\right\}$, 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 $\cup$ and $\cap$, 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 $\tilde{G}^U_1$, and the second from the matrices $\tilde{G}$'s, denoted by $\tilde{G}^U_2$. Meanwhile, the initial lower bound estimate, denoted by $\tilde{G}^L_1$, is obtained from the ConTREx matrices. The consistency check (CC) is first applied to the pair $\tilde{G}^L_1$ and $\tilde{G}^U_1$ to produce the corrected lower bound $\tilde{G}^L_2$, and then to the pair $\tilde{G}^L_2$ and $\tilde{G}^U_2$ 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 appendix A (sections A1.4 and A1.5).

2.2.2.3 TRaCE with error correction

The procedure for TRaCE with error correction is illustrated in Fig. 2-3(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. 2-2 and 2-3 in which the intersection and union operations involving $G_{\nu k_0}$ are performed only for testable edges associated with non-zero entries of $T_{\nu k_0}$. As testable edges are determined from $\tilde{G}_\varnothing$, a pre-processing step is performed to reduce errors in $\tilde{G}_\varnothing$. The premise behind the pre-processing step is that an error unlikely affects the same edge, and that testable edges of any $\tilde{G}_{\nu k_0}$ constitute only a small subset of edges in $\tilde{G}_\varnothing$ (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 section 2.3). A more detailed description of the pre-processing method can be found in section A1.3, while the filtering algorithm is provided in section A1.4 of appendix A.

The schematic diagram of TRaCE with error correction is given in Fig. 2-3(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 appendix A section A1.5. As shown in Fig. 2-3(b), the consistency check and correction are first performed for the pair $\tilde{G}_1^U$.
and $\tilde{G}_1$, 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 $\tilde{G}^U$.

### 2.2.2.4 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 $\Sigma_1 = G^L$, edges in $\Sigma_2 = G^L - \tilde{G}^L$, edges in $\Sigma_3 = G^U - G^L$, edges in $\Sigma_4 = \tilde{G}^U - G^U$, and finally edges in $\Sigma_5 = (G^U)^c$. 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 $\overline{z}_{i,j}$ according to

$$
\overline{z}_{i,j} = \begin{cases} 
(Z_{G_\emptyset})_{i,j} + \max_{(k,l)\in\Sigma_1} \overline{z}_{k,l} & \text{if } (i, j) \in \Sigma_1 \\
(Z_{G_\emptyset})_{i,j} + \max_{(k,l)\in\Sigma_2} \overline{z}_{k,l} & \text{if } (i, j) \in \Sigma_2 \\
(Z_{G_\emptyset})_{i,j} + \max_{(k,l)\in\Sigma_3} \overline{z}_{k,l} & \text{if } (i, j) \in \Sigma_3 \\
(Z_{G_\emptyset})_{i,j} + \max_{(k,l)\in\Sigma_4} \overline{z}_{k,l} & \text{if } (i, j) \in \Sigma_4 \\
(Z_{G_\emptyset})_{i,j} & \text{if } (i, j) \in \Sigma_5 
\end{cases} 
$$

Eq. 2-7

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:

$$
x_{i,j} = \frac{\overline{z}_{i,j}}{\max_{i,j} \overline{z}_{i,j}} 
$$

Eq. 2-8

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 scores. A similar procedure, called down-ranking, has been presented in Pinna et al. [48], 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.

2.3 Results

2.3.1 Inferability Analysis

We first applied TRaCE to error-free accessibility matrices of $G_\emptyset$ and $G_{v_{ko}}$ 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 [71]. 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 $G^{ul} - G_\emptyset$, respectively.

2.3.1.1 Random GRNs

We investigated the inferability of random GRNs of orders $n=10$ and $n=100$ genes. We set the network size (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 [78]. For each random network, we generated $n+1$ accessibility matrices associated with $G_\emptyset$ and $G_{v_{i}}$ 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 $G_0$ are shown in Figs. 2-4(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. 2-4. 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 $G_0$ could generally be recovered from $\overline{G}_0$ and $\overline{G}_{(i)}$’s. Nevertheless, Fig. 2-4 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.

**Figure 2-4** 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 $G_0$ are shown as a function of network size (i.e. number of edges). The error bars indicate the standard deviations.
Figure 2-5 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 $G_{\mathcal{G}}$ and as few as three $G_{\mathcal{G}}$'s. In case II, the GRN has 13 edges and is not inferable.

Fig. 2-5 shows two examples of GRN inference of order $n=10$ genes. In the first case (case I, $m=8$ edges), $G_{\mathcal{G}}$ could be recovered from $\tilde{G}_{\mathcal{G}}$ and as few as 3 $\tilde{G}_{\mathcal{G}}$'s, while in the second case (case II, $m=13$ edges), the inference problem was underdetermined. Moreover, the results suggested that $\tilde{G}_{\mathcal{G}}$'s were not equally informative, as the reduction in the distance between the lower and upper bounds by incorporating an additional $\tilde{G}_{\mathcal{G}}$ was not uniform.

2.3.1.2 Random scale-free GRNs

Many cellular networks have been shown to be scale-free with a power-law degree distribution [83], 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 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 [18]. 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 inferability analysis, we generated $n+1$ error-
free accessibility matrices $\bar{G}_2$ 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. 2-6 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 $\bar{G}_2$ 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.

**Figure 2-6** 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 $G_2$ are shown as a function of network size (i.e. number of edges). The error bars indicate the standard deviations.
2.3.1.3 *E. coli* and *S. cerevisiae* GRNs

Finally, we investigated the inferability of large, realistic GRNs of *E. coli* and *S. cerevisiae* available in GeneNetWeaver [78]. 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 $G_{(i)}$’s based on the number of testable edges (i.e. the number of non-zero elements in the testability matrix $T_{(i)}$ in Eq. 2-4). The results are shown in Figs. 2-7 and 2-8. 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 $G_{(i)}$’s for the inference of *E. coli* and yeast GRNs, respectively.

![Figure 2-7](image)

**Figure 2-7** Ensemble inference of *E. coli* GRN from error-free $G_\emptyset$ and the complete set of $G_{(i)}$’s. The plot shows the network distances of the lower and upper bounds from $G_\emptyset$ as a function of the number of $G_{(i)}$’s for the 50 most informative $G_{(i)}$’s, i.e. the top 50 highest number of testable edges. The incorporation of $G_{(i)}$’s and $G_{(i)}$’s was performed sequentially.
in decreasing number of testable edges. The inset shows the result for the complete set of $\bar{G}_{(i)}$ ’s.

![Network Distance vs Number of $G_{(i)}$'s](image)

**Figure 2-8** Ensemble inference of *S. cerevisiae* GRN from error-free $\bar{G}_0$ 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 $G_0$ as a function of the number of $\bar{G}_{(i)}$’s. The incorporation of $\bar{G}_{(i)}$’s and $G_{(i)}$’s was performed sequentially in decreasing number of testable edges.

**2.3.2 Ensemble inference from erroneous 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 $G_0$. 
2.3.2.1 *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 [78]. The average number of edges was 192. As in the above case study, we created the accessibility matrices of \( G_\varnothing \) 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 2-2.

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 \( G_\varnothing \) that did not appear in \( \tilde{G}^u \) (see \( m(G_\varnothing - \tilde{G}^u) \) in Table 2-2). 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 \( G_\varnothing \) (see \( m(\tilde{G}^l - G_\varnothing) \) in Table 2-2). 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 \( m(\tilde{G}^u - \tilde{G}^l) \) in Table 2-2). 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 \( m(G_\varnothing - \tilde{G}^u) + m(\tilde{G}^l - G_\varnothing) \) before and after correction).

<table>
<thead>
<tr>
<th>FPR</th>
<th>FNR</th>
<th>Before Correction</th>
<th>After Correction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( m(G_\varnothing - \tilde{G}^u) )</td>
<td>( m(\tilde{G}^l - G_\varnothing) )</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.1</td>
<td>183.4</td>
<td>35.86</td>
</tr>
<tr>
<td>0</td>
<td>0.2</td>
<td>191.1</td>
<td>67.36</td>
</tr>
<tr>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>1412.42</td>
</tr>
<tr>
<td>0.1</td>
<td>0.1</td>
<td>183.02</td>
<td>1441.68</td>
</tr>
<tr>
<td>0.1</td>
<td>0.2</td>
<td>191</td>
<td>1460.82</td>
</tr>
<tr>
<td>0.2</td>
<td>0</td>
<td>0</td>
<td>2324.46</td>
</tr>
</tbody>
</table>
Table 2-2 Ensemble inference of *E. coli* subnetworks (n=100 genes). 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 $m(A-B)$ of any two digraphs A and B denote the number of edges in the set $E(A)-E(B)$.

For the inference of *E. coli* GRN, we generated erroneous accessibility matrices $\bar{G}_\emptyset$ and the 100 most informative $\bar{G}_{i,j}$'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 2-3. In addition, the structural Hamming distances of the lower and upper bounds before and after correction are reported in tables A1 and A2 in appendix A. 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.

<table>
<thead>
<tr>
<th>FPR</th>
<th>FNR</th>
<th>Before Correction</th>
<th>After Correction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$m(G_\emptyset - \bar{G}^u)$</td>
<td>$m(\bar{G}^u - G_\emptyset)$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1029</td>
</tr>
<tr>
<td>0</td>
<td>0.1</td>
<td>3550</td>
<td>1685</td>
</tr>
<tr>
<td>0.2</td>
<td>0</td>
<td>3749</td>
<td>34796</td>
</tr>
<tr>
<td>0.1</td>
<td>0.1</td>
<td>3573</td>
<td>35975</td>
</tr>
<tr>
<td>0</td>
<td>0.2</td>
<td>3746</td>
<td>34566</td>
</tr>
<tr>
<td>0.2</td>
<td>0</td>
<td>0</td>
<td>62035</td>
</tr>
<tr>
<td>0.1</td>
<td>0.1</td>
<td>3554</td>
<td>62632</td>
</tr>
<tr>
<td>0.2</td>
<td>0.1</td>
<td>3741</td>
<td>63053</td>
</tr>
</tbody>
</table>

Table 2-3 Ensemble inference of *E. coli* GRN.
2.3.2.2 *S. cerevisiae* GRN

For yeast GRN, we generated erroneous $\tilde{G}_\emptyset$ and the 100 most informative $\tilde{G}_{(i)}$’s. The results of TRaCE with error correction using these accessibility matrices are summarized in Table 2-4. 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.

<table>
<thead>
<tr>
<th>FPR</th>
<th>FNR</th>
<th>Before Correction</th>
<th>After Correction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$m(G_\emptyset - \tilde{G}^{u})$</td>
<td>$m(G_\emptyset - \tilde{G}^{l})$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.1</td>
<td>4048</td>
<td>604</td>
</tr>
<tr>
<td>0</td>
<td>0.2</td>
<td>6934</td>
<td>788</td>
</tr>
<tr>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>121370</td>
</tr>
<tr>
<td>0.1</td>
<td>0.1</td>
<td>4096</td>
<td>121563</td>
</tr>
<tr>
<td>0.1</td>
<td>0.2</td>
<td>6879</td>
<td>121747</td>
</tr>
<tr>
<td>0.2</td>
<td>0</td>
<td>0</td>
<td>227013</td>
</tr>
<tr>
<td>0.2</td>
<td>0.1</td>
<td>4113</td>
<td>227087</td>
</tr>
<tr>
<td>0.2</td>
<td>0.1</td>
<td>6909</td>
<td>227150</td>
</tr>
</tbody>
</table>

Table 2-4 Ensemble inference of *S. cerevisiae* GRN.

2.3.3 Ensemble inference from expression data

We further evaluated the performance of TRaCE using in silico noisy gene expression data generated using GeneNetWeaver [78]. 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 section 2.2.2. Subsequently, we used the resulting
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. [48], GENIE3 [42] and TIGRESS [45]. 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 [84].

2.3.3.1 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 2-5 shows the FPR and FNR in the accessibility matrices, the errors in the upper and lower bounds ($m(G_G - \tilde{G}_G^u)$ and $m(\tilde{G}_G^l - G_G)$), and the size of the ensemble ($m(\tilde{G}_G^u - \tilde{G}_G^l)$) 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 $m(G_\varnothing - \tilde{G}_U)$ were higher than those in the lower bound $m(G_\varnothing - \tilde{G}_L)$.

<table>
<thead>
<tr>
<th>Network</th>
<th>FPR</th>
<th>FNR</th>
<th>$m(G_\varnothing)$</th>
<th>$m(G_\varnothing - \tilde{G}_U)$</th>
<th>$m(\tilde{G}<em>L - G</em>\varnothing)$</th>
<th>$m(\tilde{G}_U - \tilde{G}_L)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>2.28</td>
<td>176</td>
<td>44</td>
<td>9</td>
<td>136</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1.96</td>
<td>249</td>
<td>122</td>
<td>6</td>
<td>123</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>11.04</td>
<td>195</td>
<td>86</td>
<td>21</td>
<td>144</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>9.69</td>
<td>211</td>
<td>91</td>
<td>7</td>
<td>208</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>3.47</td>
<td>193</td>
<td>108</td>
<td>12</td>
<td>237</td>
</tr>
</tbody>
</table>

Table 2-5 Ensemble inference of DREAM4 100-gene gold standard networks: single-gene KO dataset.

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. 2-9 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 [48].

![Graph showing AUROC and AUPR values for different methods.](image)
**Figure 2-9** 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.

### 2.3.3.2 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 $G_0$ 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 summarized in Table 2-6. 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 2-5 and 2-6, the errors in the lower bounds improved slightly in comparison with those from only single-gene KO dataset (compare $m(G_0^{L} - G_0)$ values). However, the errors in the upper bounds increased due to the accumulation of FN errors from fan-in motifs (compare $m(G_0 - G_0^{U}$) values). Nevertheless, the additional data from double-gene KO experiments led to lower network uncertainties (compare $m(G_0^{U} - G_0^{L})$ values).

<table>
<thead>
<tr>
<th>Network</th>
<th>FPR</th>
<th>FNR</th>
<th>$m(G_0)$</th>
<th>$m(G_0^{-}G_0^{U})$</th>
<th>$m(G_0^{L} - G_0)$</th>
<th>$m(G_0^{U} - G_0^{L})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.02</td>
<td>2.21</td>
<td>176</td>
<td>50</td>
<td>5</td>
<td>27</td>
</tr>
<tr>
<td>2</td>
<td>0.01</td>
<td>1.91</td>
<td>249</td>
<td>137</td>
<td>5</td>
<td>37</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>10.43</td>
<td>195</td>
<td>103</td>
<td>15</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>9.31</td>
<td>211</td>
<td>118</td>
<td>8</td>
<td>35</td>
</tr>
<tr>
<td>5</td>
<td>0.01</td>
<td>3.40</td>
<td>193</td>
<td>131</td>
<td>12</td>
<td>21</td>
</tr>
</tbody>
</table>
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. 2-10. 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 2-7 and 2-8, the AUROC and AUPR values of TRaCE were insensitive to $z_{\text{cutoff}}$ and $z_{\text{threshold}}$ used in the construction of the accessibility matrices, and the threshold value in the preprocessing of $\overline{G}_a$. Because of the trade-off between FPs and FNs when using different $z_{\text{cutoff}}$ and $z_{\text{threshold}}$, the maximum of AUROC and AUPR corresponded to intermediate values within the selected ranges of $z_{\text{cutoff}}$ and $z_{\text{threshold}}$.

**Table 2-6** Ensemble inference of DREAM4 100-gene gold standard networks: single- and double-gene KO dataset.

**Figure 2-10** 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.

\[
\begin{array}{|c|c|c|c|}
\hline
z_{\text{threshold}} & z_{\text{cutoff}} & \text{AUROC} & \text{AUPR} \\
\hline
1.5 & 2 & 0.871 \pm 0.0616 & 0.3085 \pm 0.1416 \\
1.5 & 3 & 0.8723 \pm 0.0673 & 0.3209 \pm 0.1534 \\
2 & 2 & 0.8756 \pm 0.0595 & 0.2922 \pm 0.154 \\
2 & 3 & 0.8758 \pm 0.0642 & 0.2889 \pm 0.1596 \\
2.5 & 3 & 0.8758 \pm 0.0642 & 0.2438 \pm 0.1545 \\
\hline
\end{array}
\]

Table 2-7 $z_{cutoff}$ and $z_{\text{threshold}}$ on AUROC and AUPR of TRaCE in DREAM4 100-gene subchallenge: single gene KO data. The AUROC and AUPR values are the average ± standard deviation over 5 gold standard networks. The values used in the comparison with existing methods are highlighted in bold.

\[
\begin{array}{|c|c|c|c|}
\hline
z_{\text{threshold}} & z_{\text{cutoff}} & \text{threshold} & \text{AUROC} & \text{AUPR} \\
\hline
1.5 & 2 & 0.5 & 0.8723 \pm 0.0595 & 0.4327 \pm 0.1931 \\
 & & 0.65 & 0.8718 \pm 0.0588 & 0.4197 \pm 0.1959 \\
 & & 0.8 & 0.8724 \pm 0.0600 & 0.4184 \pm 0.2022 \\
1.5 & 3 & 0.5 & 0.8728 \pm 0.0596 & 0.4204 \pm 0.1986 \\
 & & 0.65 & 0.8733 \pm 0.0610 & 0.4212 \pm 0.2059 \\
 & & 0.8 & 0.8735 \pm 0.0617 & 0.4201 \pm 0.2100 \\
2 & 2 & 0.5 & 0.8726 \pm 0.0603 & 0.4177 \pm 0.2158 \\
 & & 0.65 & 0.8727 \pm 0.0607 & 0.4167 \pm 0.2170 \\
 & & 0.8 & 0.8728 \pm 0.0608 & 0.4091 \pm 0.2136 \\
2 & 3 & 0.5 & 0.8735 \pm 0.0618 & 0.4170 \pm 0.2160 \\
 & & 0.65 & 0.8734 \pm 0.0620 & 0.4086 \pm 0.2149 \\
 & & 0.8 & 0.8698 \pm 0.0604 & 0.4045 \pm 0.2108 \\
2.5 & 3 & 0.5 & 0.8695 \pm 0.0599 & 0.3893 \pm 0.2054 \\
 & & 0.65 & 0.8696 \pm 0.0598 & 0.3837 \pm 0.1957 \\
 & & 0.8 & 0.8695 \pm 0.0600 & 0.3764 \pm 0.2023 \\
\hline
\end{array}
\]

Table 2-8 Effects of $z_{cutoff}$, $z_{\text{threshold}}$ and the pre-processing threshold of $\vec{G}_{\phi}$ on AUROC and AUPR of TRaCE in DREAM4 100-gene subchallenge: single- and double-gene KO data. The
AUROC and AUPR values are the average ± standard deviation over 5 gold standard GRNs. The values used in the comparison with existing methods are highlighted in bold.

2.3.3.3 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 2-9, 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.

<table>
<thead>
<tr>
<th>Network</th>
<th>Replicates</th>
<th>FPR</th>
<th>FNR</th>
<th>( m(\hat{G}_e - \hat{G}^-) )</th>
<th>( m(\hat{G}^+ - \hat{G}_e) )</th>
<th>( m(\hat{G}^- - \hat{G}^+) )</th>
<th>AUROC</th>
<th>AUPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>E. coli</td>
<td>5</td>
<td>0.117</td>
<td>2.61</td>
<td>1611</td>
<td>687</td>
<td>1724</td>
<td>0.8601</td>
<td>0.4836</td>
</tr>
<tr>
<td>E. coli</td>
<td>10</td>
<td>0.005</td>
<td>2.61</td>
<td>1612</td>
<td>297</td>
<td>1673</td>
<td>0.8639</td>
<td>0.5192</td>
</tr>
<tr>
<td>Yeast</td>
<td>5</td>
<td>0.582</td>
<td>24.97</td>
<td>8101</td>
<td>9159</td>
<td>7907</td>
<td>0.7699</td>
<td>0.2464</td>
</tr>
<tr>
<td>Yeast</td>
<td>10</td>
<td>0.141</td>
<td>24.98</td>
<td>8098</td>
<td>3753</td>
<td>7200</td>
<td>0.7569</td>
<td>0.2768</td>
</tr>
</tbody>
</table>

Table 2-9 Ensemble inference of E. coli and Yeast GRNs from single-gene KO data. FPR (FNR) is the ratio between the number of FP (FN) in the accessibility matrices and the number of edges in \( G_e \). Let \( m(A-B) \) of any two digraphs A and B denote the number of edges in the set \( E(A)-E(B) \).
2.4 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 [63, 68], 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 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 $\tilde{G}_{(i)}$’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 is presented in a Chapter 3.

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 [67].

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. A3 and section A1.6 in appendix A). Meanwhile, FP errors could cancel out some Type A errors associated with FNs at the cost of increased uncertainty (see Fig. A3 and section A1.6 in appendix A).
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 2-9). 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 $G_{V_{ko}}$, leading to a detectable and correctable type C error. We expect the issue above would improve when using more sensitive measurements of gene expression.
Chapter 3 REDUCE\(^2\)

3.1 Introduction

As mentioned in Chapter 1, a large number of algorithms have been developed to infer GRNs from expression data [12, 31, 33, 36, 53, 63]. The crux of the problem is the underdetermined nature of this inference, leading to the lack of network identifiability or inferability [73, 85]. The inferability of a network from a dataset depends on the availability of relevant information in the data, which is in turn determined by the experiments generating the data. We have shown in the previous chapter that systematic KO experiments are severely suboptimal. In consequence, designing optimal experiments for GRN inference is important in tackling the inferability issues.

The attention given to designing experiments for GRN inference pales in comparison to developing inference methods. Only a handful of strategies having been proposed previously. For example, Ideker et al. [86] proposed an optimization of perturbation experiments for a Boolean model of acyclic GRNs using minimum set cover. Tegner et al. [87] formulated a heuristic strategy of ranking gene perturbations where genes with weaker differential expression or those associated with more uncertain interactions, are more likely selected for KOs. Meanwhile, Spieth et al. [88] employed an evolutionary strategy (ES) to create an ensemble of S-system models of GRN. The DOE involved performing virtual gene KOs using the model ensemble and choosing the most informative KO experiment. On the other hand, Steinke et al. [89] developed a DOE strategy based on Bayesian linear regression with a sparse prior distribution of the GRN, where experiments were selected according to the possible information gain. More recently, Lang et al. [90] proposed a DOE strategy for cellular reaction networks based on selecting a set of measurements, using which low confidence reactions are isolated in disjoint subnetworks or modules. Finally, Birget et al. [91] used a graph theory

---

\(^2\) The content of this chapter is derived from the peer-reviewed publication: Ud-Dean SM, Gunawan R. Optimal design of gene knockout experiments for gene regulatory network inference. Bioinformatics. 2015 Nov 14:btv672. © The authors. The introduction, methods and discussion parts have been modified in order to put it into the context of this thesis.
concept called node cut-sets or vertex separators, to formulate a systematic procedure for inferring GRN digraphs. Briefly, the strategy involves systematically knocking-out the vertex separators of gene pairs with an indirect regulation, i.e. gene pair $i$ and $j$ where gene $i$ regulates gene $j$ through other gene(s). The authors showed theoretically that the inference of acyclic GRN with $n$ genes would require $O(n)$ gene KO experiments, while those with cycles would need $O(n^2)$ experiments.

In this chapter, we have developed an algorithm called REDuction of UnCertain Edges (REDUCE) for selecting the optimal gene KO experiment based on an ensemble of GRNs, particularly using the upper and lower bounds of the ensemble. REDUCE was formulated as a constrained optimization problem to maximize the number of uncertain edges that could potentially be verified. We introduced the concept of edge separatoid, similar to vertex separators, as the basis to count the number of possible edge verification associated with a given gene KO combination. Finally, we proposed an iterative procedure for the GRN inference, in which the upper and lower bounds of the ensemble are continually updated with every iteration of wet-lab KO experiments and dry-lab optimal DOE using REDUCE. As a proof of concept, we applied the iterative procedure to infer the GRN of *E. coli* under ideal conditions. We further demonstrated the efficacy of REDUCE using benchmark gene expression simulator GeneNetWeaver [78] in the inference of five 100-gene gold standard networks from DREAM 4 *in silico* network inference challenge [78]. We compared the performance of the iterative inference with performing systematic KO experiments.

### 3.2 Method

#### 3.2.1 Definitions

In this section, we review several concepts of graph theory which will be used in the development of REDUCE in addition to the concepts already defined in Chapter 2. For a directed edge $(i,j)$ in a digraph, vertex $i$ is called a *parent* of vertex $j$, and correspondingly vertex $j$ is a *child* of vertex $i$. In this chapter, we consider a digraph model of GRN where the
edges are unsigned and unweighted. In such a digraph, the nodes represent genes while the edges describe the gene regulatory interactions. The directed edge \((i,j)\) indicates that gene \(i\) regulates gene \(j\). In the following, the graph \(G_\emptyset\) denotes the digraph of the GRN of interest, which is also referred to as wild-type GRN. Meanwhile, the digraph corresponding to knocking out or deleting a set of genes \(V_{ko} \subseteq V(G_\emptyset)\) is denoted by \(G_{V_{ko}}\). Fig. 3-1(a) shows an example of a GRN digraph \(G_\emptyset\) with 7 genes \((n(G) = 7)\) and 7 gene interactions \((m(G) = 7)\). Here, genes \(B\) and \(C\) are parents of gene \(D\), and genes \(D\) and \(E\) are children of gene \(C\). Fig. 3-1(b) further shows the digraph \(G_{\{A,E\}}\) corresponding to knocking out genes \(A\) and \(E\) from the GRN \(G_\emptyset\) in Fig. 3-1(a).

**Figure 3-1.** (a) Example of a digraph of GRN \(G_\emptyset\). (b) The GRN \(G_{\{A,E\}}\) after knocking out genes \(A\) and \(E\) from \(G_\emptyset\).

In a digraph, if vertex \(j\) is accessible from vertex \(i\) (i.e. a directed path from \(i\) to \(j\) exists), vertex \(i\) is called an *ancestor* of vertex \(j\), and vertex \(j\) is correspondingly called a *descendant* of vertex \(i\). In Fig. 3-1(a), gene \(F\) is accessible from gene \(A\). Furthermore, genes \(A, B, C, D\) and \(E\) are the ancestors of gene \(H\), while genes \(D, E, F\) and \(H\) are the descendants of gene \(B\).

As mentioned in the previous chapter, from a KO expression dataset TRaCE produces the upper and lower bounds (denoted by \(G^U\) and \(G^L\) respectively) of an ensemble of networks consistent with the data. The edges in \(G^U\) that do not appear in \(G^L\), defined by the set
\( E_U = \{(i,j): (i,j) \in G^U, (i,j) \notin G^L\}, \) are referred to as uncertain edges since their existence could not be verified by the available data. The number of uncertain edges (i.e., the cardinality of \( E_U \) or \( N(E_U) \)) gives a measure of the uncertainty in a particular GRN inference problem.

### 3.2.2 Design of Experiments by REDUCE

The premise behind REDUCE is to identify the optimal set of genes whose knock out or deletion would enable the verification of the highest number of uncertain edges. As inputs, REDUCE requires the upper and lower bounds of the ensemble, such as those generated by TRaCE. We will illustrate the main concept of REDUCE using the following example. Consider the upper and lower bound digraphs shown in Fig. 3-2. Here, there are two uncertain edges \((A, F)\) and \((B, H)\) since these edges appear in \( G^U \) but are missing from \( G^L \). In order to confirm the uncertain edge \((A, F)\), we consider knocking-out (disconnecting) all indirect paths from gene \(A\) to gene \(F\) in \( G^U \). Removing any one of the genes in the set \(\{C, E\}\) or both genes would accomplish this task. When there exists no indirect path from gene \(A\) to gene \(F\), the verification of the edge \((A, F)\) becomes simple. For example, if perturbing gene \(A\) leads to a differential expression of gene \(F\) in the background of gene \(C\) knock-out, then we can confirm the existence of \((A, F)\). Otherwise, the edge \((A, F)\) does not exist. Similarly, for the edge \((B, H)\), knocking out one of the genes in the set \(\{D, E\}\) or both genes would remove all indirect paths from \(B\) to \(H\) in \( G^U \). In this exercise, the optimal KO experiment would therefore be to knock out gene \(E\) as this would simultaneously disconnect the indirect paths from \(A\) to \(F\) and from \(B\) to \(H\). Note that we still need to perturb or knock out gene \(A\) and \(B\) individually in the background of gene \(E\) deletion to verify the uncertain edges.
Figure 3-2. Example of upper and lower bounds of a GRN.

In the above illustration, we call the sets of genes \{C\}, \{E\} and \{C, E\} as the edge separatoids of (A, F). More precisely, we define an edge separatoid of \((i, j) \in \mathcal{E}_U\) as the set of nodes whose removal would disconnect all indirect paths from node \(i\) to node \(j\) in \(G^U\). There could be more than one (edge) separatoid for an uncertain edge as demonstrated in the previous example. In addition, two or more uncertain edges could share the same separatoid as in the case of gene \(E\) with respect to the uncertain edges \((A, F)\) and \((B, H)\). In REDUCE, we consider the following separatoids for each uncertain edge \((i, j) \in \mathcal{E}_U\):

1. \(S_1(i, j) = \text{children of } i \text{ in } G^U \cap \text{ancestors of } j \text{ in } G^U\)
2. \(S_2(i, j) = \text{descendants of } i \text{ in } G^U \cap \text{parents of } j \text{ in } G^U\)
3. \(S_3(i, j) = \text{descendants of } i \text{ in } G^U \cap \text{ancestors of } j \text{ in } G^U\)

For the example in Fig. 3-2, we have \(S_1(A, F) = \{C\}\), \(S_2(A, F) = \{E\}\), and \(S_3(A, F) = \{C, D, E\}\). The separatoids \(S_1(i, j)\) and \(S_2(i, j)\) are both subsets of \(S_3(i, j)\), and \(S_3(i, j)\) is the largest separatoid of \((i, j)\). For the optimization of gene KO, we further define the following sets of separatoids:

1. \(S_1 = \{S_1(i, j) : (i, j) \in \mathcal{E}_U\}\)
2. \(S_2 = \{S_2(i, j) : (i, j) \in \mathcal{E}_U\}\)
3. \(S_3 = \{S_3(i, j) : (i, j) \in \mathcal{E}_U\}\)
The three separatoids above do not, by any means, represent the complete set of separatoids for an uncertain edge \((i, j)\). Rather, they are selected because of the ease in computing and storing them. One could also consider the grandchildrens of \(i\) or grandparents of \(j\), and so on. However, delineating all possible separatoids would constitute finding the longest path between two nodes in a graph, a problem which is known to be NP-hard [92]. In addition, the memory requirement would become prohibitively large as the separatoids are computed \textit{a priori} and stored in memory during KO optimization.

As mentioned above, REDUCE involves finding the optimal combination of nodes whose removal would enable the verification of the highest number of uncertain edges. Given the aforementioned sets of edge separatoids \(S_1\), \(S_2\) or \(S_3\), we solve the following optimization problem

\[
(q^*, V^*_K) = \arg \max_{q \in \{1, 2, 3\}} \max_{V_{KO}} N\left(E_{T,q}(V_{KO})\right)
\]

where \(E_{T,q}(V_{KO}) = \{(i, j) : (i, j) \in E_U, S_q(i, j) \subseteq V_{KO}, i, j \not\in V_{KO}\}\) and \(N(E_{T,q}(V_{KO}))\) is the cardinality of \(E_{T,q}(V_{KO})\). In this optimization, \(E_{T,q}(V_{KO})\) represents the set of uncertain edges that could potentially be verified by \(G_{V_{KO}}\) according to the set of separatoids \(S_q\) \((q = 1, 2, 3)\). One can impose constraints in the above optimization, such as to exclude essential genes or combinations of genes whose KOs are lethal, and to limit the number of KO genes \(\text{i.e. the cardinality of } V_{KO}, \text{ or } N(V_{KO})\). In the implementation of REDUCE, the optimization is carried out for each \(S_q\) \((q = 1, 2, 3)\) separately using a modified genetic algorithm (GA) [93], the maximum of which is selected after the completion of the GA optimizations (see pseudo-code in Appendix B).

Following the simple illustration above, the verification of uncertain edges involves obtaining gene expression data from the following experiments:
(1) deletion of genes in the set $V_{KO}^*$

(2) perturbation of knock out of each gene $i$ from the set $I^*$ in the background $V_{KO}^*$

deletion, where $I^* = \{i: (i, j) \in E_{T,q}^*(V_{KO}^*)\}$.

Using the data from the above experiments, we perform a two-sample t-test to determine if the
perturbation of gene $i$ leads to a differential expression of gene $j$ in the background of $V_{KO}^*$
KO for each edge $(i, j) \in E_{T,q}^*(V_{KO}^*)$. Based on the t-test, we update the ensemble bounds as
follows:

(1) if the null hypothesis is rejected, then we add $(i, j)$ to the lower bound $G_L$;

(2) otherwise, we remove $(i, j)$ from the upper bound $G_U$.

One can repeat applying REDUCE, carrying out KO experiments, and updating the
ensemble bounds until the upper and lower bounds of the ensemble converge or until the
distance between these bounds does not reduce further or until a given number or budget of
experiments is reached. As outlined in Fig. 3-3, the inference of GRN can therefore be carried
out iteratively. The total number of KO experiments is thus given by the summation between
the number of iterations and the cumulative number of elements of $I^*$'s. In our experience, when
the GRNs contain cycles, the sets of edge separatoids defined above may become sensitive to
errors, particularly to false negatives in the upper bound $G_U$. In the implementation of the
iterative procedure, we employed the following sets for REDUCE:

(1) $S_1(i, j) =$ children of $i$ in $G_{U,k}^* \cap$ ancestors of $j$ in $G_{U,0}^*$

(2) $S_2(i, j) =$ descendants of $i$ in $G_{U,0}^* \cap$ parents of $j$ in $G_{U,k}^*$

(3) $S_3(i, j) =$ descendants of $i$ in $G_{U,0}^* \cap$ ancestors of $j$ in $G_{U,0}^*$

where $G_{U,k}^*$ denotes the upper bound of the ensemble in the $k$-th iteration and is $G_{U,0}^*$ the initial
upper bound.
In practice, multiplex assay such as in RNA sequencing plays an important role in cost- and time-saving by processing a large number of samples simultaneously. If desired, the iterative inference procedure can be modified for multiplexing. Briefly, the modified procedure involves running REDUCE sequentially without bound updates until the uncertain edges are exhausted or until no feasible solution can be found or until a desired number of KO experiments (see Appendix B). In each iteration, we thus obtain a ranked list of \( \{V_{KO,j}^{*}\} \), instead of a single optimal \( V_{KO}^{*} \), with non-increasing \( N\left(E_{T,q}\left(V_{KO}^{*}\right)\right) \).

**Figure 3-3.** Iterative procedure for GRN inference. The procedure starts with an initial \( G^U \) and \( G^L \), for example from the outputs of TRaCE. REDUCE uses the ensemble bounds to find the optimal set of gene KOs \( V_{KO}^{*} \) for the subsequent experiments. The resulting gene expression data are then used to update the ensemble bounds. The procedure is repeated until convergence.

### 3.2.3 Comparison to other DOE strategies

A direct comparison between REDUCE and several existing DOE algorithms for GRN inference in the case studies is complicated by (1) differences in the modeling framework used
to represent GRNs (e.g. Boolean acyclic graph [86] and ordinary differential equations [88]), (2) the types or the parameterizations of network perturbations (e.g. using network input functions [89]), and (3) ambiguity in the published procedure (e.g. in the generation of ensemble of solutions to the linear regression problem in the method by Tegnér et al. [87]). For these reasons, in the next section we compared REDUCE to systematic gene KO procedures, including the complete set of double-gene KOs (DKOs), DKO s from Ancestor-Descendant (AD) pairing, and the DOE proposed by Birget et al. which used a similar graph theoretic concept to edge separatoids, namely vertex separators [91]. In DKO s from AD pairing, we knock out a gene pair \( i \) and \( j \), if gene \( j \) is accessible from gene \( i \) and if there exist at least one gene \( k \) \((k \notin \{i, j\})\) which is accessible from both \( i \) and \( j \). When the KO of genes \( i \) and \( j \) leads to a differential expression of gene \( k \) with respect to the KO of only gene \( j \), then we add the edge \((i, k)\) to the lower bound. Otherwise, we remove the edge \((i, k)\) from the upper bound.

### 3.3 Results

In this section, we demonstrate the performance of REDUCE by applying the iterative procedure in Fig. 3-3 to three case studies, involving 100-gene random scale-free networks under ideal conditions, \( E. \ coli \) GRN under ideal conditions and five 100-gene gold standard GRNs from DREAM 4 challenge using GeneNetWeaver data [78]. As the starting point, we assume that the complete set of single-gene KO (SKO) experiments have been performed. We used TRaCE to construct the initial \( G^U \) and \( G^L \) from the expression data from SKO. Note that in this case \( G^{U,0} \) is the accessibility matrix of \( G_\emptyset \). For the 100-gene DREAM4 GRNs, we employed GeneNetWeaver (GNW) for the data generation using the same settings as in the challenge. In the implementation of REDUCE, we used GA with a population size of 100 and a maximum generation of 50, which we found to give a good balance between finding globally
optimal solution and reducing computational cost. All other GA optimization parameters were set to the default values (see Appendix B).

3.3.1 Comparison to the DOE by Birget et al.

We first compare REDUCE to the KO design procedure by Birget et al using vertex separators. We applied the DOEs under the ideal scenario where we can accurately detect differential expressions of genes that are accessible from the deleted genes. In this case data noise does not play any role in the inferability of GRN. REDUCE has several key advantages over this strategy. First, REDUCE allows constraints of practical significance in the gene KO optimization, such as excluding essential genes whose knock out would leave the cell inviable. In addition, prior knowledge and expression data could be easily taken into account in the ensemble bounds input for REDUCE (see discussion section). Meanwhile, the systematic KO procedure of Birget et al. was developed specifically for GRN digraphs without cycles. The procedure also did not consider any constraints nor allow incorporation of prior data.

As shown in Table 3-1, for ten randomly generated 100-gene scale-free acyclic GRNs [18], the DOE of Birget et al. prescribed significantly more KO experiments than the iterative inference using REDUCE. Furthermore, the KO experiments from the systematic procedure involved a very high number of genes (up to 50 genes). Using our iterative network inference procedure, we could consistently reach the true GRNs using fewer KO experiments, while limiting the number of KO genes in a given experiment (up to 10 genes). Because of the clear advantages of REDUCE over the DOE of Birget et al, in the remaining case studies we will compare REDUCE only to strategies using DKO's.
<table>
<thead>
<tr>
<th>Network</th>
<th>Number of KO Experiments: Iterative DOE (at most 10 genes in $V_{ko}$)</th>
<th>Number of KO Experiments: Birget et al.</th>
<th>Maximum Number of Genes in KO Experiments: Birget et al.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>678</td>
<td>1320</td>
<td>35</td>
</tr>
<tr>
<td>2</td>
<td>626</td>
<td>1884</td>
<td>28</td>
</tr>
<tr>
<td>3</td>
<td>644</td>
<td>1292</td>
<td>36</td>
</tr>
<tr>
<td>4</td>
<td>666</td>
<td>1574</td>
<td>37</td>
</tr>
<tr>
<td>5</td>
<td>744</td>
<td>2120</td>
<td>46</td>
</tr>
<tr>
<td>6</td>
<td>738</td>
<td>2222</td>
<td>26</td>
</tr>
<tr>
<td>7</td>
<td>760</td>
<td>2138</td>
<td>50</td>
</tr>
<tr>
<td>8</td>
<td>747</td>
<td>2766</td>
<td>44</td>
</tr>
<tr>
<td>9</td>
<td>666</td>
<td>2234</td>
<td>24</td>
</tr>
<tr>
<td>10</td>
<td>670</td>
<td>1830</td>
<td>39</td>
</tr>
</tbody>
</table>

Table 3-1. Inference of random 100-gene scale-free acyclic GRNs under ideal conditions. The inference was carried out until completion, leading to the true GRN in all cases.

### 3.3.2 Inference of *E. coli* GRN under ideal condition

As a proof of the applicability of REDUCE to large realistic GRNs, in this case study, we used the *E. coli* GRN in GNW with 1565 nodes and 3758 edges [78]. Here, we again considered data under ideal conditions. Assuming that we started with data from the complete SKO experiments, we constructed the accessibility matrix of $G_{\emptyset}$ and its reduction by ConTREx, and set these as the initial ensemble bounds $G^{U,0}$ and $G^{L,0}$, respectively. We constrained REDUCE such that the optimal solution excludes essential genes, whose KOs are detrimental to *E. coli* viability [94]. We implemented two versions of the iterative procedure. In the first implementation used a fixed maximum number of 10 genes in $V_{ko}$ ($N(V_{ko}) \leq 10$). In the second implementation, we started with a maximum of one gene in $V_{ko}$ and gradually increased this limit when the GA optimization could not find any feasible solution (i.e. when the remaining
separatoids involved more genes than the prescribed limit. The ensemble bounds update followed a modified procedure as outlined in Appendix B.

Fig. 3-4 shows the Jaccard distances between $G^U$ and $G_\emptyset$, and between $G^L$ and $G_\emptyset$. The Jaccard distance between two digraphs $G_1$ and $G_2$ is a measure of the difference between the sets of edges in $G_1$ and $G_2$, defined by [95]:

$$JD(G_1, G_2) = \frac{|N(E(G_1) \cup E(G_2)) - |N(E(G_1) \cap E(G_2))|}{|N(E(G_1) \cup E(G_2))|}$$

A Jaccard distance of 1 means that there exist no common edge in $G_1$ and $G_2$, while a Jaccard distance of 0 means that $G_1$ and $G_2$ have the same sets of edges. When limiting the number of genes in the $V_{KO}$ to 10, the iterative procedure converged to the true $G_\emptyset$ in 166 iterations with a total of 437 KO experiments, as shown in Fig. 3-4(a). We could also obtain the true GRN by gradually increasing the limit of genes in $V_{KO}$ (see Fig. 3-4(b)), but not surprisingly, this implementation required more iterations and more KO experiments (247 iterations and 539 KO experiments). Meanwhile, $G^U$ and $G^L$ from the complete set of double-gene KO (DKO) experiments with a total of ~1.22 million KO experiments did not meet (see dotted lines in Fig. 3-4). The DKOs from AD pairing produced ensemble bounds $G^U$ and $G^L$ with Jaccard distances similar to those using the complete DKOs (see dashed lines in Fig. 3-4), but using much fewer experiments (700 KO experiments). This case study thus demonstrated that the iterative network inference using REDUCE could provide much more informative experiments than systematic designs using DKOs.
Figure 3-4. Iterative inference of *E. coli* GRN under ideal conditions. The plot shows the Jaccard distances of $G^U$ and $G_{\emptyset}$, and the negative Jaccard distances of $G^L$ and $G_{\emptyset}$. The number of genes in $V_{KO}$ is limited to 10 genes in (a), and increased from 1 to 9 genes (demarcated by vertical lines) in (b).

### 3.3.3 Inference of 100-gene DREAM 4 challenge GRNs

In this case study, we applied the iterative procedure to infer five 100-gene gold standard networks in the DREAM 4 *in silico* network inference challenge [65, 78]. For each KO experiment, we simulated 10 replicates of steady state gene expression data using GNW [78]. GeneNetWeaver employed two types of biological noise: inherent stochastic noise associated with gene transcription process and additive measurement noise. The intrinsic stochastic noise was simulated using stochastic differential equations with independent Gaussian white-noise, while log-normal measurement noise was added to the simulated expression data [96].

As before, we started with initial data from the complete set of SKO experiments and constructed $G^{U,0}$ and $G^{L,0}$ using TRaCE. The differential expression analysis in TRaCE was performed using a procedure described previously (using $z_{\text{cutoff}} = 3$ and $z_{\text{threshold}} = 2$) [85]. We applied the iterative procedure using $\alpha = 0.01$ for the two-sample t-test during the ensemble bounds update. Here, we gradually increased the maximum number of genes in $V_{KO}$ starting from 1 and incrementing this number by 1 when REDUCE could not find any feasible solution.
For all five gold standard networks, the iterative procedure terminated in the convergence between $G^U$ and $G^L$. The iterations for the inference of these networks involved at most 3 gene KOs ($N(V_{KO}) \leq 2$), except for Network 2 which required only 2 gene KOs ($N(V_{KO}) = 1$). The accuracy of the resulting GRNs is summarized in Fig. 3-5. For each gold standard network, we compared the iterative procedure to performing the complete set of DKO experiments and DKOs based on AD pairing.

**Figure 3-5.** Comparison of REDUCE DOE and DKOs on DREAM 4 100-gene networks: (a) number of uncertain edge verifications and (b) total network distance. (a) The number of uncertain edges verified by the iterative procedure using REDUCE, in comparison to the complete set of DKO experiments and to DKOs based on AD pairing. (TN: True Negative, FN: False Negative, TP: True Positive and FP: False Positive) (b) The total distance among $G^U$, $G^L$ and $G_{\emptyset}$. The total distances are reported as a fraction of the size (the number of edges) of $G_{\emptyset}$.

Under non-ideal scenario, we could not obtain the true GRN even when using the iterative procedure. For all gold standard networks, $G^U$ and $G^L$ from the iterative procedure converged to a GRN that was different from $G_{\emptyset}$. Nevertheless, as shown in Fig. 3-5(a), the iterative procedure consistently led to the verification of more uncertain edges than DKO data (paired t-test $p=0.001$ against complete DKOs and $p = 0.003$ against DKOs using AD pair). The
fractions of false positives and false negatives among the verified edges did not significantly correlate with the number of uncertain edges (for FP: $\rho = 0.46$, $p = 0.43$; for FN: $\rho = -0.20$, $p = 0.75$). The numbers of experiments using the iterative procedure were between 30 to 110 times lower (see Table 3-2) than the complete DKO s (4950 experiments), but were higher than the DKO experiments using AD pairing. Again, we did not notice significant correlations between the number of uncertain edges and the number of iterations as well as the number of KO experiments (for iterations: $\rho = 0.75$, $p = 0.14$; for KOs: $\rho = 0.69$, $p = 0.20$). Fig. 3-5(b) shows the total distance (TD) among $G^U$, $G^L$ and $G_\emptyset$ which is calculated as follows:

$$TD(G^U, G^L, G_\emptyset) = N\left(E(G^U) \cup E(G^L) \cup E(G_\emptyset)\right) - N\left(E(G^U) \cap E(G^L) \cap E(G_\emptyset)\right)$$

The total distance gives a combined measure of uncertainty and accuracy of the ensemble with respect to the gold standard network. As shown in Fig. 3-5(b), for all five GRNs, the iterative inference procedure could provide lower TD than the complete DKO s (paired t-test $p = 0.009$) and DKO s from AD pairs ($p = 0.001$).

<table>
<thead>
<tr>
<th>Network</th>
<th>Number of Iterations using REDUCE</th>
<th>Number of KO Experiments using REDUCE\textsuperscript{a}</th>
<th>Number of KO Experiments using AD pairing\textsuperscript{a}</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>34</td>
<td>48</td>
<td>41</td>
</tr>
<tr>
<td>2</td>
<td>21</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>3</td>
<td>55</td>
<td>106</td>
<td>57</td>
</tr>
<tr>
<td>4</td>
<td>80</td>
<td>120</td>
<td>64</td>
</tr>
<tr>
<td>5</td>
<td>54</td>
<td>77</td>
<td>57</td>
</tr>
</tbody>
</table>

\textbf{Table 3-2.} Number of iterations and total KO experiments in the iterative inference of 100-gene DREAM 4 GRNs \textsuperscript{a}not including KOs involving single gene.
Additionally we computed the Jaccard distances between $G^U$ and $G_\emptyset$ and between $G_\emptyset$ and $G^L$, which are shown in Fig. 3-6. The upper bounds $G^U$ from the proposed iterative inference had similar Jaccard distances to those from the complete DKOs ($p=0.48$) despite using much fewer experiments. On the other hand, the optimal DOE consistently produced lower Jaccard distances for the lower bound $G^L$ than the complete DKO ($p=0.005$). In comparison to DKO from AD pairs, the iterative procedure led to lower Jaccard distances for both $G^U$ (paired t-test $p=0.006$) and $G^L$ ($p=0.001$). We also implemented the modified iterative procedure for multiplex assay where we again gradually increased the cardinality of $V_{KO}$. The resulting GRNs were of similar accuracy as those from the original procedure above (see Appendix B Fig. B2). As expected, the use of multiplex assay led to fewer iterations (~5 fold decrease) at the cost of a small increase in the total number of KO experiments (see Appendix B Table B1). Taken together, the results of the case studies demonstrated the power of REDUCE for inferring GRN, and further suggested that systematic KOs of genes can be severely suboptimal for such a purpose.
3.3.4 Runtime and computational complexity

The computational complexity of the optimization problem in REDUCE depend on the number of genes \( n \); the number of uncertain edges \( N(E_U) \); GA parameters, specifically the population size \( (n_{pop}) \) and the number of generations \( (n_{gen}) \); and the constraint on the maximum number of genes \( (N_{max}) \) allowed in \( V_{KO} \). Specifically, the calculation of the separatoids prior to the GA optimization, has a computational complexity that scales linearly with the number of gene \( n \). Furthermore, the complexity of GA optimization scales linearly with the population size and the number of generations, while the computation of the objective function \( N(E_{T,q}(V_{KO})) \) (see Appendix B) scales with \( O((N(E_c))^2 N_{max}) \). Meanwhile, the memory requirement of REDUCE is dominated by the storing of separatoids, which scales with \( O(nN(E_c)) \). For the \textit{E. coli} GRN example with 1,565 genes and 11,411 initial uncertain edges, the GA optimization in REDUCE completed in around 90 seconds on a workstation with 3.33 GHz Intel® Xeon® W3680 Processor (6 cores), and used 1.4 GB of RAM.

3.4 Discussion

In this work, we developed a method called REDUCE for optimizing gene KO experiments for the purpose of inferring GRN digraphs. The method builds on the ensemble inference of GRNs using gene expression data [85]. In particular, REDUCE uses the upper and lower bounds of an ensemble of GRNs to find the optimal set of gene KOs which would potentially reduce the most number of uncertain edges. We further proposed an iterative procedure which cycled over performing gene KO experiments, updating ensemble bounds, and optimizing gene KO by REDUCE. As a proof of principle, we successfully applied the iterative procedure to infer the GRN of \textit{E. coli} under ideal conditions (no data noise, infinite sensitivity). The iterative inference could converge to the true GRN, whereas performing all combinations of DKO or DKOs based on AD pairing could not despite the larger number of experiments.
Using benchmark data generator and 100-gene gold standard networks of DREAM 4 challenge, the proposed iterative inference procedure could significantly outperform DKO experiments providing informative data, as judged by network distances from the true GRNs. In particular, the iterative procedure could converge to a unique digraph with a lower total distance than the ensemble bounds from the complete set of DKO experiments, while using 1-2 orders of magnitude fewer experiments. For roughly the same number of experiments, DKO experiments based on AD pairing led to verifications of fewer uncertain edges and much larger total distances from the gold standard network.

We recommend implementing the iterative inference procedure using a gradual increase of KO genes, i.e. starting with \( N(V_{KO}) = 1 \) and increasing \( N(V_{KO}) \) when REDUCE could not find any feasible solution. Such a strategy is preferred because knocking out a large number of genes simultaneously could be detrimental to cell viability. As shown in Fig. 3-4(b), even with limiting the number of KO genes to no more than 3 (\( V_{KO} \) with at most 2 genes), the iterative procedure could verify a large fraction (93%) of the uncertain edges in \( E. coli \) GRN. Also, in practice one would not necessarily want to carry out the iterations until completion (e.g. due to budget constraint). Here, REDUCE will still prove to be useful in maximizing the information return per KO experiment.

For the 100-gene networks in the DREAM 4 challenge, the iterative procedure produced a unique digraph (i.e., \( G^U \) and \( G^L \) converge), but the inferred GRN deviated from the true GRN. Here, errors in the input upper and lower bounds contributed significantly (79%) to the total errors in the final GRNs (i.e. the majority of the errors were not due to REDUCE but due to errors in the initial bounds generated by TRaCE.). False negatives (FNs) dominated over false positives (86% of errors were FN). We further noted that on average, 90% of the FNs were associated with fan-in motifs where a gene was regulated by two or more genes. The inference of such motifs from steady-state gene KO data is fundamentally challenging because perturbing
one of the regulators may not lead to any differential expression of the target gene due to compensatory effects. Meanwhile, the remaining errors could not be associated with any network motifs. Since errors in the input to REDUCE were not included in the set of uncertain edges, the iterative inference could not correct these errors.

Beside the above issue, a recent study demonstrated that the ordering of gene deletions could influence cell’s transcriptomic profiles in KO with multiple genes [97]. Analyzing the data from this study, we found that the order of deleting two genes aceA and pgi in E. coli affected the expression of 53 out of 4690 genes (FDR < 5%). Furthermore, gene aceA showed a differential expression upon deleting gene pgi, suggesting that aceA is accessible from pgi. However, pgi was not differentially expressed in the KO of aceA. Consequently, we should be able to verify edges emanating from pgi to all genes that were accessible from both pgi and aceA by analyzing the differential expression of DKO ∆pgi and ∆aceA in the background of ∆aceA. Among the verifiable edges satisfying the condition above (ignoring antisense reading), the transcriptomic discrepancy between DKO s of ∆aceA-then-∆pgi and ∆pgi-then-∆ace did not cause any difference in the verification outcomes of these edges (0 out of 3). Nevertheless, the influence of the order of gene deletions could further complicate the GRN inference, which future DOEs and network inference algorithms would need to address.

The issue of false negative above could be addressed by considering other types of data, for example time-series gene expression data and transcription factor binding sites. When considering time-series data, we ideally need (1) fast sampling to capture transient changes in the expression of target genes, and (2) slow compensation by other regulators. Meanwhile, if the binding sites of transcription factors are known (see for example FANTOM project [60]), one could then construct a transcriptional regulatory network. Any edges in the transitive closure of this transcriptional network which do not appear in the input upper bound, are possible false negatives and should be added to the upper bound.
The initial ensemble bounds for the iterative procedure in the case studies came from applying TRaCE to expression data of SKO experiments. If the transcription factor genes are known *a priori*, then TRaCE requires only the SKO data of each transcription factor gene to construct the initial upper and lower bounds of the GRN ensemble. Beside using TRaCE, one could also construct the initial upper and lower bounds of the ensemble from prior knowledge. When the members of the initial ensemble are known *a priori*, the upper and lower bounds could be constructed by taking the union and intersection of the members, respectively. For example, one could obtain the initial ensemble from the GRN predictions of different network inference algorithms, following the idea of wisdom of crowds [67]. When observational data (as opposed to KO data) are available, one could also construct a Markov equivalence class, for example using PC algorithm [98, 99]. The Markov equivalence class represents the ensemble of DAGs encoding the same independence and conditional relationships that result from a Bayesian network learning using such data. Again, the upper and lower bound could be constructed by taking the union and intersection of the DAGs in this equivalence class.
Chapter 4 TRaCE+³

4.1 Introduction

In Chapter 2, we have developed TRaCE for constructing an ensemble of GRN structures that are consistent with the input expression data from gene KO experiments [85]. In developing TRaCE, we sought only to establish the existence of gene regulations and ignored the signs of the edges, following the DREAM GRN inference challenges. Nevertheless, the edge signs in a GRN digraph are often of great interest and significance, as they indicate the modes of the gene regulations. Here, a positive edge reflects an activation, while a negative edge describes a repression. Several notable network inference algorithms such as TRANSitive reduction for WEighted Signed Digraphs (TRANSWESD) [50] and Local Transitive Reduction (LTR) [51] previously considered the inference of GRN digraph with signed (and weighted) edges. However, these algorithms were not designed for inferring an ensemble of GRN structures. In this chapter, we address the above-mentioned limitation of TRaCE by developing a new ensemble inference method, called TRaCE+. The new method uses a signed digraph model of the GRN, i.e. the edges have signs. Like TRaCE, TRaCE+ generates the upper and lower bounds of the ensemble, but in the form of signed digraphs. The ensemble bounds from TRaCE+ are also compatible with REDUCE, the optimal design of gene KO strategy described in the previous chapter [79]. We demonstrated the advantages of TRaCE+ over TRaCE in the ensemble inference of *Escherichia coli* GRN and in the iterative inference of 100-gene gold standard GRNs from DREAM 4 *in silico* network inference challenge

---

³ The content of this chapter is submitted as the manuscript: Ud-Dean SM, Heise S, Klamt S and Gunawan R. TRaCE+: Ensemble inference of gene regulatory networks from gene knock-out experiments. The introduction, methods and discussion parts have been modified in order to put it into the context of this thesis.
4.2 Method

4.2.1 Definitions

In this section we provide few more concepts of graph theory that are relevant to the development of TRaCE+ in addition to the concepts already introduced in Chapters 2 and 3. A signed digraph \( G^+ = (V(G^+), E(G^+), S(G^+)) \) is the digraph \( (V(G^+), E(G^+)) \) with an edge mapping \( S:E \rightarrow \{+,-\} \) that assigns a positive or negative sign to each edge. In the following sections, we focus on the inference of GRN structure in the form of a signed digraph. We denote the GRN of interest as \( G_{\varnothing}^+ \), which is also referred to as the wild-type GRN. In such a graph, the nodes represent the genes, the signed directed edges indicate the gene regulations. A positive (negative) edge pointing from gene \( i \) to gene \( j \) implies that the products of gene \( i \) upregulates (downregulates) the expression of gene \( j \). Fig. 4-1(a) gives an example of a signed digraph of a GRN with 5 genes \( (n(G) = 5) \) and 7 gene regulatory edges \( (m(G) = 7) \). Three of the gene regulations are negative (downregulations), while four are positive (upregulations). Here, gene \( C \) is a parent of genes \( D \) and \( E \), while gene \( D \) is a child of genes \( C, B \) and \( E \). Meanwhile, gene \( A \) is an ancestor of the genes \( C, D \) and \( E \), while gene \( E \) is a descendant of the genes \( A, B, C \) and \( D \). We further denote the GRN corresponding to knocking-out (deleting) a set of genes \( V_{ko} \subset V(G_{\varnothing}^+) \) by \( G^+_{V_{ko}} \). Fig. 4-1(b) illustrates the network \( G^+_{[D]} \), where gene \( D \) has been deleted from the GRN \( G_{\varnothing}^+ \) in Fig. 4-1(a).
Figure 4-1. Illustration of signed digraph GRNs. (a) Example of a signed digraph $G^+_\emptyset$. The pointed black arrows indicate positive regulations while the flathead red arrows represent negative regulations. (b) The network $G^+_\{D\}$ in which gene D has been deleted.

4.2.2 Steps of TRaCE+

As we have already mentioned, one of the biggest drawbacks of TRaCE is its disregard of the signs (modes) of the regulatory edges. For this reason, we have developed TRaCE+. Like TRaCE, TRaCE+ also generates the upper and lower bounds of the ensemble, but with signed edges (i.e., these bounds are signed digraphs). TRaCE+ comprises two main steps: (1) constructing the ensemble upper and lower bounds using data from the complete set of single-gene KOs, and (2) updating the ensemble bounds using additional gene KO data. Below, we provide more details of the two algorithms.

Construction of Upper Bound from Single-gene KOs

In constructing the upper and lower bounds of the ensemble, we start by putting together the accessibility matrix using the transcriptional expression data from the complete set of single-gene KOs. Here, we follow the same procedure as that in TRaCE [85]. Briefly, for each technical replicate, we obtain the sample mean $\mu'_j$ and standard deviation $s'_j$ of the expression
of gene \(j\). Subsequently, we calculate the corrected sample mean \(\mu_j'\) and standard deviation \(s_j'\) by excluding the expression data of gene \(j\) \(g_{ij}\) that differ from \(\mu_j'\) by more than \(z_{\text{cutoff}}s_j'\).

Using \(\mu_j\) and \(s_j\), we then compute the z-scores \(z(i, j) = \frac{g_{i,j} - \mu_j}{s_j}\) which indicates the differential expression of gene \(j\) in the KO of gene \(i\). Finally, we average the z-scores over the technical replicates to give the overall z-score matrix \(Z(i,j)\), based on which we obtain the accessibility matrix by the following criteria:

\[
\text{Acc}(i, j) = \begin{cases} 
1 & \text{if } |Z(i, j)| > z_{\text{threshold}} \\
0 & \text{if } |Z(i, j)| \leq z_{\text{threshold}}
\end{cases}
\]

This \(\text{Acc}\) gives the adjacency matrix of the upper bound \(G^U\). In the case studies, we employed \(z_{\text{cutoff}}=3\) and \(z_{\text{threshold}}=2\) following the recommendations in TRaCE [85]. In contrast to TRaCE, we also set a sign for each edge (non-zero element in \(\text{Acc}\)) in the upper bound \(G^U\) as follows:

\[
S(i, j) = \begin{cases} 
+ & \text{if } Z(i, j) \leq 0 \\
- & \text{if } Z(i, j) > 0
\end{cases}
\]

**Construction of Initial Lower Bound**

In TRaCE, the lower bound of the ensemble \(G^L\) came from applying ConTREx (condensation, transitive reduction, and expansion) to the upper bound \(G^U\) above without considering the signs [85]. The transitive reduction in ConTREx involved removing from this DAG any edge \((i,j)\) for which there exist a directed path from node \(i\) to node \(j\) not involving this edge. Extending the transitive reduction procedure to a signed digraph means that we should remove any edge \((i,j)\) for which there exist a directed path from node \(i\) to node \(j\) not involving the edge \((i,j)\) and the cumulative product of the edge signs on this path is equal to the sign of the edge \((i,j)\). This simple procedure may not work when \(G^U\) contains a *negative cycle* (a directed cycle with an odd number of negative edges), since the cumulative sign of such a cycle alternates depending.
on how many times one traverses through it. A recent study comparing different ways to obtain
the transitive reduction of a signed GRN digraph (with and without cycles) recommended a
simple procedure called Local Transitive Reduction (LTR) [51]. In the following, we have
adapted LTR to generate the lower bound signed digraph $G^l$ for TRaCE+.

For the determination of the lower bound $G^l$, we first assign a weighting factor $W(i,j)$ for
each edge $(i,j)$ in $G^U$, which equals to the magnitude of the average correlation coefficient
between gene $j$ and gene $i$ taken over the technical replicates. In the calculation of this
correlation, we exclude data from gene $j$ KO experiment. Adapting the LTR procedure, we first
equate the two bounds ($G^l = G^U$), and subsequently remove any edge $(i,j)$ in $G^l$ for which
there exists a path of length 2 in $G^U$ explaining the edge $(i,j)$, or more precisely there exists a
node $k$ with $(i,j),(i,k),(k,j) \in G^U$ and $k \notin \{i,j\}$, such that

1. the directed path $i,k,j$ is sign consistent with $(i,j)$, i.e. $S(i,j) = S(i,k)S(k,j)$.

2. the weight of the edge $(i,j)$ satisfies $w_{cut}W(i,j) < W(i,k)W(k,j)$, where

$$w_{cut} \in [0,1]$$

Note that by setting $w_{cut} = 0$, we effectively ignore the contribution of the edge weights. By
considering only directed paths of length 2, the procedure above avoids the problem associated
with negative cycles since any path going through a cycle more than once would necessarily
have a length longer than 2. In addition, LTR does not require condensation and expansion
steps as in TRaCE’s ConTREx. Unlike the original version of LTR, here we do not check
whether a removal of an edge would change the outcome of previous edge removals, and as a
result, $G^l$ may not have the same accessibility relationships as $G^U$. While implementing the
check would lead to fewer uncertain edges, it would also cause more false positive errors that
could neither be corrected by additional data nor new experiments (see section 4.4).
Fig. 4-2 illustrates the computations of the lower bound $G^l$ in TRaCE and in TRaCE+ (using $w_{cut}$ of 0.3) for the wild-type GRN $G^*_w$ in Fig. 4-1(a). The comparison showed that accounting for signs and weights of the edges could lead to a higher retention of true edges in $G^l$, and thus to fewer uncertain edges. This difference demonstrated that some information could be lost by disregarding edge signs in the GRN inference, as done in TRaCE.

![Figure 4-2](image)

**Figure 4-2.** Comparison of ensemble upper and lower bounds obtained from single-gene KO data by (a & b) TRaCE and (c & d) TRaCE+ (with randomly assigned weights). The true GRN is shown in Fig. 4-1(a). (a & c) Ensemble upper bound $G^U$. (b & d) Ensemble lower bound $G^L$. Note that the lower bound from TRaCE is not the transitive reduction of the upper bound due to the existence of a cycle between D and E.

### 4.2.3 Ensemble Bounds Update

The ensemble bounds update algorithm allows the incorporation of transcriptional expression data from gene KO experiments beyond single-gene KOs. The update follows an iterative procedure involving (1) evaluation of *separatoids* for uncertain edges, (2) determination of verifiable uncertain edges, and (3) refinement of the ensemble bounds. For this purpose, we define a separatoid of an uncertain edge $(i, j) \in E_u$, denoted by $Sep(i, j)$, as the set of nodes whose removal would eliminate any directed path of length 2 or longer from node $i$ to node $j$ [79]. The deletion of genes in $Sep(i, j)$ would therefore give the GRN $G_{Sep(i,j)}$. 

85
where the only remaining directed path from gene \( i \) to gene \( j \) is the edge \((i, j)\) itself.

Consequently, we can verify the existence of the uncertain edge \((i, j)\) by assessing whether gene \( j \) is differentially expressed upon knocking-out gene \( i \) in the background of \( \text{Sep}(i, j) \) gene deletions.

An uncertain edge may have more than one separatoid, while several uncertain edges may share a separatoid. Following our previous work [79], in the first step, we compute three separatoids for each uncertain edge \((i, j)\) \(\in E_U\):

1. \(\text{Sep}_1(i, j) = \text{children of } i \text{ in } G_U \cap \text{ancestors of } j \text{ in } G_{U,0}\)
2. \(\text{Sep}_2(i, j) = \text{descendants of } i \text{ in } G_{U,0} \cap \text{parents of } j \text{ in } G_U\)
3. \(\text{Sep}_3(i, j) = \text{descendants of } i \text{ in } G_U \cap \text{ancestors of } j \text{ in } G_{U,0}\)

where \(G_U\) is the most updated upper bound and \(G_{U,0}\) is the initial upper bound from single-gene KO data. The separatoids above are not the only separatoids for the uncertain edge \((i, j)\). We limit our analysis only to these separatoids because they are easy to compute. Finding all separatoids for a given edge constitutes finding the longest path between two nodes, which is a NP-hard problem [92]. For the ensemble bounds in Fig. 4-2(c & d), there exist three uncertain edges \((A,D)\), \((A,E)\) and \((C,E)\). Here, the three separatoids of \((A,E)\) are given by \(\text{Sep}_1(A,E) = \{C\}, \text{Sep}_2(A,E) = \{D\}, \text{and } \text{Sep}_3(A,E) = \{C,D\}\).

In the second step we identify among the set of input gene KOs \(V_{\text{KO}}\) (including single-gene KOs), pairs of KO experiments whose data would allow the verification of the edge \((i,j)\). More precisely, for each \((i,j) \in E_U\), we search for the pair of KO experiments \((V_{\text{KO}}, V_{\text{KO}})\) such that \(\text{Sep}_l(i,j) \subseteq V_{\text{KO}}\) for any \(l = 1, 2, 3\), and \(i, j \notin V_{\text{KO}}\). Following the definition of a separatoid above, if gene \( j \) is differentially expressed between any
of such pairs of KO experiments, then we have evidence supporting for the existence of the uncertain edge \((i,j)\).

In the third step, for each uncertain edge \((i,j)\), we perform a (two-tailed) two-sample \(t\)-test with \(\alpha = 0.01\) to determine whether the expression of gene \(j\) is significantly different between the KO of \(V_{k_{i, ko}} \cup i\) in comparison to the KO of \(V_{k_{ko}}\). In the case that we only find one pair of such experiments for an uncertain edge \((i,j)\), we remove this edge from the upper bound \(G^U\) upon a failure to reject the null hypothesis in the \(t\)-test above. Otherwise, we add the edge \((i,j)\) to the lower bound \(G^L\). Further, if the average expression of gene \(j\) in the KO of \(V_{k_{ko}} \cup i\) is lower (higher) than that in the KO of \(V_{k_{ko}}\), we assign a positive (negative) sign to this edge.

In some cases, we may find more than one pair of KO experiments for an uncertain edge \((i,j)\). For each of these KO pairs, we again employ a two-sample \(t\)-test with \(\alpha = 0.01\). The result of each \(t\)-test counts as a vote for the existence of the edge \((i,j)\) in the case of rejection of the null hypothesis, or a vote against the existence of this edge in the case of failure to reject the null hypothesis. If the votes against the edge exceed those for the edge, then we remove this edge from the upper bound \(G^U\). Otherwise, we add the edge \((i,j)\) to the lower bound \(G^L\). In case of a tie, we do not change the bounds, i.e. the edge \((i,j)\) remains uncertain. We also determine the edge sign by voting. Specifically, we set a positive (negative) sign when the average expression of gene \(j\) in the KO of \(V_{k_{ko}} \cup i\) is more frequently lower (higher) than that in the KO of \(V_{k_{ko}}\), among the KO pairs giving confirmatory votes for the edge. When a tie occurs, we keep the original sign of the edge \((i,j)\) from \(G^U\).

Once the ensemble bounds are updated, we recalculate the separatoids to reflect the changes brought by the additions and removals of edges to and from the bounds. We repeat the steps
described above until we can not find any suitable pairs of KO experiments for the remaining uncertain edges.

4.2.4 Iterative Gene Regulatory Network Inference

In Chapter 3, we proposed an iterative GRN inference procedure which combines TRaCE and REDUCE [79]. We demonstrated that this iterative procedure could resolve the underdetermined issue of the GRN inference, producing a unique GRN. TRaCE+ can substitute TRaCE in this iterative inference to enable the inference of a signed digraph model of GRN. As shown in Fig. 4-3, the iteration starts with the construction of signed digraphs of the ensemble bounds using single-gene KO data. Based on these bounds, we optimize the next set of gene KO experiments using REDUCE. Briefly, REDUCE employs the edge separatoids and a constrained optimization to obtain the optimal set of gene KOs that would enable the verification of the maximum number of uncertain edges. The next step in the iteration is to carry out the optimized gene KOs experiments and obtain new transcriptional expression data. Subsequently, we feed the data back to TRaCE+ to update the ensemble bounds following the procedure described in the previous section. We repeat these steps until the ensemble bounds converge or does not improve further, or until a given quota on the number of KO experiments has been reached.

If desired, one can also perform REDUCE multiple times in a given iteration to generate a large list of gene KO experiments (see Fig. 4-3, dashed arrows). This implementation is particularly suitable for using multiplexing assay technology. Here, at the end of one REDUCE optimization, we remove the set of verifiable uncertain edges from subsequent runs. We perform REDUCE until all uncertain edges become verifiable or until we hit constraint(s) on the optimization. When only a subset of these KO experiments could be performed, one should select gene KOs from the list above in the order that they are generated (since earlier runs of REDUCE are associated with more verifiable uncertain edges).
4.2.5 Case Studies and Performance Evaluation

In order to evaluate the performance of TRaCE+, we applied the algorithms to the ensemble inference of *Escherichia coli* GRN from single-gene KO experiments, and to the iterative inference of 100-gene gold-standard GRNs from DREAM 4 *in silico* network inference challenge [65, 78]. For each KO experiment, we generated 10 replicates of *in silico* (simulated) gene KO data using the benchmark GRN data generator GeneNetWeaver with the default parameters [78]. The quality of the ensemble bounds was assessed by using true positive rate, total distance and Jaccard distance with respect to the reference GRNs. The true positive rate (TPR) was calculated as the ratio between the number of edges in the reference GRN that were correctly identified in the lower bound $G^l$ and the total number of edges in the reference network $G^+_\varnothing$, or more precisely:

$$TPR = \frac{N(E(G^l) \cap E(G^+_\varnothing))}{N(E(G^+_\varnothing))}$$
where \( N\left( E(G^+_{\varnothing}) \right) \) denotes the cardinality of the set \( E(G^+_{\varnothing}) \). Meanwhile, the total distance between the ensemble bounds and the reference GRN was computed as follows:

\[
TD = \frac{N\left( E(G^U) \cup E(G^L) \cup E(G^+_{\varnothing}) \right) - N\left( E(G^U) \cap E(G^L) \cap E(G^+_{\varnothing}) \right)}{N\left( E(G^+_{\varnothing}) \right)}
\]

A higher TD value indicates larger uncertainty in the GRN inference (i.e. worse inferability).

Finally, the Jaccard distance (JD) was evaluated using the following formula:

\[
JD(G_1, G_2) = \frac{N\left( E(G_1) \cup E(G_2) \right) - N\left( E(G_1) \cap E(G_2) \right)}{N\left( E(G_1) \cup E(G_2) \right)}
\]

The JD gives a measure of similarity between two digraphs \( G_1 \) and \( G_2 \). A JD of 1 indicates that the two digraphs have no common edges and a JD of 0 implies that the two digraphs share the same set of edges. In the case studies, we evaluated the JDs between \( G^U \) and \( G^+_{\varnothing} \), as well as between \( G^L \) and \( G^+_{\varnothing} \). When dealing with ensemble bounds in the form of signed digraphs such as those generated by TRaCE+, the intersections among the sets of edges in the evaluations of TPR, TD and JD were done by respecting the sign of the edges (i.e. edges of unequal signs were not counted).

### 4.3 Results

#### 4.3.1 Ensemble inference of \( E. coli \) GRN

In this case study, we employed a signed digraph \( E. coli \) GRN with 1565 genes and 3758 regulatory interactions, available in GeneNetWeaver [78]. We generated \textit{in silico} data for the complete set of single-gene KOs as described in Method. Using this dataset, we constructed unsigned digraph ensemble bounds using TRaCE [85] and signed digraph bounds using TRaCE+. For TRaCE+, we also studied how the ensemble bounds, particularly \( G^L \), depend on \( w_{\text{cut}} \) by varying this parameter between 0 and 1 (at 0.1 increments). We compared the
performance of TRaCE and TRaCE+ according the TPRs, TDs and JDs of the resulting bounds with respect as described in Method. The comparison in Fig. 4-4 shows that TRaCE+ could provide ensemble bounds with higher TPRs and lower JDs and TDs than TRaCE. These trends demonstrated TRaCE+’s ability to extract information contained in the gene regulatory signs that were disregarded by TRaCE. The TDs of the bounds from TRaCE+ generally improved with higher values of $w_{cut}$, but the improvements reached a plateau after $w_{cut}$ of 0.4. However, the lower bounds from TRaCE+ slightly worsened with increasing $w_{cut}$. The JDs between the upper bound and the reference network differed little between TRaCE+ and TRaCE due to the consideration of edge signs in computing JDs for the upper bounds from TRaCE+. As expected, the upper bound from TRaCE+ did not vary with $w_{cut}$.

![Diagram](image)

**Figure 4-4.** Ensemble bounds from TRaCE and TRaCE+ for *E. coli* GRN: True positive rate (TPR), total distance (TD) and Jaccard distance (JD).
4.3.2 Iterative GRN inference of DREAM4 100-gene networks

In this case study, we applied the iterative GRN inference using either TRaCE or TRaCE+ to the five signed digraph gold-standard networks from DREAM 4 100-gene in silico network inference challenge [65, 78]. At the start of the iterative procedure, we simulated the complete set of single-gene KO data as described in Method. Figs. 4-5 (a & b) show respectively the TPRs and TDs of the ensemble bounds. According to the TDs and TPRs, the ensemble bounds from TRaCE+ consistently outperformed those from TRaCE regardless of the parameter $w_{cut}$. Here, TDs and TPRs improved slightly with increasing $w_{cut}$. Figs. 4-5(c & d) provide the JDs of the ensemble bounds from TRaCE and from TRaCE+ with different $w_{cut}$ values. Like in the E. coli case study, the JDs of the ensemble upper bounds did not differ significantly between TRaCE and TRaCE+, nor did they depend on $w_{cut}$. The JDs of the lower bounds from TRaCE+ were mostly better than those from TRaCE, where the best JDs corresponded to $w_{cut}$ values between 0.2 and 0.4. In the following, we compared the performance of the iterative inference using TRaCE and using TRaCE+ with a $w_{cut}$ of 0 (i.e. ignoring edge weights) and an intermediate $w_{cut}$ of 0.3.

In the implementation of REDUCE, we put a constraint on the maximum number of genes in the optimal KO experiments. We started with a maximum of 2 genes, and incremented this constraint by 1 when the optimization within REDUCE could not produce any feasible solution. We again employed GeneNetWeaver to generate in silico data for the optimal KO experiments. We performed the iterative procedure until the ensemble bounds converged.
Figure 4-5. Ensemble bounds from TRaCE and TRaCE+ using single-gene KOs for DREAM 4 100-gene GRNs: (a) True positive rate (TPR), (b) total distance (TD), (c) Jaccard distance (JD) between $G^U$ and reference network, and (d) Jaccard distance (JD) between $G^L$ and reference network.

For all of the five gold-standard GRNs, the iterations terminated in the convergence of the ensemble bounds, i.e. we obtained a unique GRN. Fig. 4-6 shows the TPRs, TDs and JDs of the inferred GRNs, as well as the total number of KO experiments required (excluding single-gene KOs). The iterations using TRaCE+ ($w_{cut} = 0$ and $w_{cut} = 0.3$) produced slightly better GRNs than TRaCE in terms of TPRs and JDs. Of course, the edges in the GRNs from TRaCE+ had signs, while those from TRaCE did not. More importantly, the iterations using TRaCE+ required much fewer KO experiments to reach convergence than TRaCE ($p = 0.013$ for $w_{cut} = 0$ and $p = 0.027$ for $w_{cut} = 0.3$), by as much as 19%. This trend signified the ability of TRaCE+ to extract more information from the data. Table 4-1 further compares the number of iterations and the highest number of genes involved the KO experiments. The numbers of iterations using TRaCE+ were generally lower than using TRaCE.
Figure 4-6. Comparison of iterative GRN inference using TRaCE and TRaCE+ ($w_{cut} = 0$ and $w_{cut} = 0.3$). (a) True positive rate (TPR), (b) total distance (TD), (c) Jaccard distance (JD), (d) total number of KO experiments.

<table>
<thead>
<tr>
<th>Gold-standard Network</th>
<th>TRaCE</th>
<th>TRaCE+ ($w_{cut} = 0$)</th>
<th>TRaCE+ ($w_{cut} = 0.3$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iterations</td>
<td>Max KO</td>
<td>Iterations</td>
</tr>
<tr>
<td>1</td>
<td>34</td>
<td>3</td>
<td>31</td>
</tr>
<tr>
<td>2</td>
<td>21</td>
<td>2</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>55</td>
<td>3</td>
<td>49</td>
</tr>
<tr>
<td>4</td>
<td>80</td>
<td>3</td>
<td>70</td>
</tr>
<tr>
<td>5</td>
<td>54</td>
<td>3</td>
<td>56</td>
</tr>
</tbody>
</table>

Table 4-1. Iterative inference of DREAM4 100-gene gold-standard GRN using TRaCE and TRaCE+ ($w_{cut} = 0$ and $w_{cut} = 0.3$): Number of iterations and maximum number of genes in KO experiments.
As a further comparison, we also generated \textit{in silico} data for the complete set of double-gene KOs, a total of 4,950 KO experiments. We used this dataset to update the ensemble bounds initially constructed using single-gene KOs. As shown in Table 4-2, only a small fraction of the double-gene KO experiments were useful for verifying uncertain edges, and a number of uncertain edges still remained after the ensemble bound update. Fig. 4-7 gives the TPRs, TDs and JDs of the ensemble bounds. Compared to TRaCE, the bounds update considering edge signs in TRaCE+ led to better TPRs ($p = 0.02$ for $w_{cut} = 0$, and $p = 0.016$ for $w_{cut} = 0.3$). However, the differences in the TDs and JDs between TRaCE and TRaCE+ were not significant.

<table>
<thead>
<tr>
<th>Network</th>
<th>TRaCE</th>
<th>TRaCE+ ($w_{cut} = 0$)</th>
<th>TRaCE+ ($w_{cut} = 0.3$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Informative experiments</td>
<td>Uncertain Edges</td>
<td>Informative experiments</td>
</tr>
<tr>
<td>1</td>
<td>36</td>
<td>15</td>
<td>24</td>
</tr>
<tr>
<td>2</td>
<td>27</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>3</td>
<td>61</td>
<td>37</td>
<td>52</td>
</tr>
<tr>
<td>4</td>
<td>49</td>
<td>81</td>
<td>43</td>
</tr>
<tr>
<td>5</td>
<td>46</td>
<td>37</td>
<td>44</td>
</tr>
</tbody>
</table>

\textbf{Table 4-2.} Ensemble bound update using double-gene KOs in TRaCE and TRaCE+ ($w_{cut} = 0$ and $w_{cut} = 0.3$): Number of informative double-gene KOs for verifying uncertain edges and number of remaining uncertain edges after ensemble bound update.
Figure 4-7. Comparison of ensemble bounds updates using the complete double-gene KO data in TRaCE and TRaCE+ ($w_{cut} = 0$ and $w_{cut} = 0.3$). (a) True Positive Rate (TPR), (b) Total Distance (TD), (c) Jaccard distance (JD) between $G^U$ and reference network, and (d) Jaccard distance (JD) between $G^L$ and reference network.

Finally, we performed the iterative procedure using TRaCE and using TRaCE+ with $w_{cut} = 0$ and $w_{cut} = 0.3$, where we implemented multiplexed REDUCE to generate a large number of KO experiments. We again fixed the maximum number of genes in the KO experiments at each iteration, beginning with 2 and incrementing this limit by 1 when multiplexed REDUCE could not generate any feasible KO experiments. For all gold-standard networks, the iterations generated a unique GRN (i.e. the ensemble bounds converged). Fig. 4-8 summarizes the quality of the ensemble bounds according to TPRs, TDs and JDs. The results closely resembled those from the iterations without multiplexing. Again, employing TRaCE+ led to fewer total KO experiments than using TRaCE ($p = 0.011$ for $w_{cut} = 0$ and $p = 0.006$ for $w_{cut} = 0.3$). In
comparison to the results without multiplexing in Table 4-1, Table 4-3 shows that multiplexing could reduce the number of iterations tremendously.

![Comparison of Iterative GRN Inference](image)

**Figure 4-8.** Comparison of iterative GRN inference with multiplexing assay using TRaCE and TRaCE+ (\(w_{cut} = 0\) and \(w_{cut} = 0.3\)). (a) True positive rate (TPR), (b) total distance (TD), (c) Jaccard distance (JD), and (d) total number of KO experiments.

<table>
<thead>
<tr>
<th>Net</th>
<th>TRaCE</th>
<th>TRaCE+</th>
<th>TRaCE+(w)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iterations</td>
<td>Max KO</td>
<td>Iterations</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>14</td>
<td>3</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
<td>3</td>
<td>12</td>
</tr>
</tbody>
</table>

**Table 4-3.** Iterative inference of DREAM4 100-gene gold-standard GRNs with multiplexing assay using TRaCE and using TRaCE+ (\(w_{cut} = 0\) and \(w_{cut} = 0.3\)): Number of iterations and maximum number of genes in KO experiments.
4.4 Discussion

In this chapter, we developed a new ensemble inference algorithm called TRaCE+ for the identification of GRN structures in the form of a signed digraph. Unlike the traditional GRN inference, TRaCE+ produces upper and lower bounds of an ensemble of signed digraphs, describing the family of GRNs that are consistent with the gene accessibility relationships established by the input transcriptional expression profiles. Specifically, these bounds define the set of uncertain gene regulatory edges that could not be verified by the available data. The outputs of TRaCE+ are directly compatible with our recent optimal design of gene KO experiments called REDUCE and the accompanying iterative GRN procedure [79]. As shown in the case study using 100-gene gold-standard GRNs from DREAM 4 in silico network inference challenge, by iterating TRaCE+, REDUCE and performing optimized gene KO experiments, one can overcome the underdetermined issue of GRN inference and obtain a unique GRN in a relatively small number of iterations (especially when using multiplexing assay). Like TRaCE, a drawback of TRaCE+ is that the procedure requires at the minimum the complete set of single gene KO data, which could become prohibitive for large-scale GRNs. However, we expect that accelerating progress in high-throughput gene editing technology (e.g., CRISPR-Cas9) and RNA sequencing will soon make such a requirement non-limiting.

The consideration of regulatory signs in TRaCE+ represents a significant advance over TRaCE, as the mode of the gene regulations (activation/repression) is very often an important aspect in the applications of GRN. Here, we adapted LTR to get around the issue in obtaining transitive reductions for GRNs with negative cycles. As demonstrated in the case studies, by taking the edge signs into account, TRaCE+ can extract more information from the data than TRaCE. As a result, the numbers of uncertain edges in the ensemble from TRaCE+ were consistently lower than those from TRaCE using the same set of KO data (single-gene KOs).
Furthermore, in the iterative inference, employing TRaCE+ led to significantly fewer total gene KO experiments to reach convergence than using TRaCE. While we used edge weights only for constructing the ensemble lower bound from the initial upper bound, these weights could also serve as a measure of confidence (likelihood) for the existence of an edge (as done in a previous method called TRANSWESD [50]).

There exist many reasons for errors to happen during the ensemble bounds construction and updates, including noise and bias in expression data as well as (type-I and type-II) errors in the statistical tests. Three types of errors can appear in the ensemble bounds from TRaCE+. False negative (FN) errors involve regulatory edges in the reference GRN that do not appear in the upper bound \( G^U \). Meanwhile, false positive (FP) errors pertain to regulatory edges in the lower bound \( G^L \) that do not belong to the true network. Finally, incorrect sign (IS) errors refer to edges in the reference GRN that have the opposite signs in the upper bound. Among the three types of errors, our experience from the case studies showed that FNs were the most common errors while IS errors were the least common, typically affecting less than 1% of the edges in the reference networks. We further noted that the majority (>80%) of FN errors in the case studies were associated with fan-in motifs, where several genes regulated a common target gene. Here, knocking-out only one of the regulators might not cause any differential expression of the target gene due to compensation by the other regulator(s).

Once occurred, FP and FN errors could not be corrected during the iterative GRN inference since these errors affected edges that were not uncertain. In the second case study, the large majority of the errors in the inferred GRN were already present in the initial ensemble bounds constructed using single-gene KO data. Nevertheless, new FP and FN errors could also appear and accumulate over the iterations. By modifying the parameters in TRaCE+, including \( z_{cutoff} \), \( z_{threshold} \), \( w_{cut} \) and \( \alpha \), we can adjust the frequency of FPs and FNs. Lowering \( z_{cutoff} \) and \( z_{threshold} \) has the effect of reducing FN errors, but comes at the cost of higher FP errors and uncertain
edges. We previously showed that $z_{\text{cutoff}} = 3$ and $z_{\text{threshold}} = 2$ provide a good balance among FNs, FPs and uncertain edges [85]. Meanwhile, increasing $w_{\text{cut}}$ could reduce the number of uncertain edges, but also cause more FPs. On the other hand, lowering the parameter $\alpha$ in the ensemble bound update should reduce FPs at the trade-off of increasing FNs. In the ensemble bounds of $E.\ coli$ and DREAM 4 GRNs from single-gene KO data, the frequency of FNs ranged between 24% - 56% ($E.\ coli$: 44%), while the frequency of FPs varied between 2.8% - 13% ($E.\ coli$: 8.1%) when using $w_{\text{cut}} = 0$. These frequencies were reported as a fraction of the number of edges in the reference GRN. Increasing $w_{\text{cut}}$ to 0.3 led to more FPs, especially for $E.\ coli$ GRN (from 8.1% to 26%). The increase in FPs by using $w_{\text{cut}} = 0.3$ among DREAM 4 GRNs was however quite modest (2.9% - 14.5%).
Chapter 5 Conclusions

Despite tremendous efforts from multiple disciplines, inferring GRNs from gene expression data remains extremely challenging. The effort thus far focused on developing algorithms for GRN inference. However, there is a dearth of theoretical studies into the informativeness of the data and experiments used for GRN inference, a critical factor that determines whether or not the inference is even mathematically possible. This thesis presented the first theoretical framework for analyzing the informativeness of data from gene KO experiments in inferring GRNs. Specifically, the framework and algorithms developed in this thesis would allow the estimation of the uncertainty in a GRN inference in terms of ensemble upper and lower bounds. Moreover, the design of experiment algorithm presented herein can provide the optimal KO experiments for reducing this uncertainty. Overall, the theoretical framework and algorithms in this thesis represented a rigorous and effective strategy for resolving the challenge in inferring GRNs.

The difficulty in GRN inference originates from the underdetermined nature of the inference problem as the data do not contain sufficient information to identify all causal interactions among the genes [63, 68]. Consequently, there exist many indistinguishable solutions, i.e. an ensemble of GRNs consistent with the data. Therefore, in this thesis, we have adopted an ensemble inference strategy and developed new frameworks and algorithms called TRaCE and TRaCE+ for the creation of an ensemble of GRN digraphs and signed digraphs, respectively. In particular, TRaCE [85] and TRaCE+ produce the lower and upper bounds of the ensemble. The ensemble represents the uncertainty associated with differentiating direct and indirect regulations using steady-state gene expression data of gene KO experiments. TRaCE+ significantly expanded the capability of TRaCE, enabling the inference of the mode of the gene regulations (activation/repression). By considering the signs of the regulatory edges, TRaCE+
could extract more information from gene KO data than TRaCE, and as a result, reduce the number of uncertain edges.

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 cannot be fully inferred even when error-free data from the 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 down-ranking method [48], the best overall performer in the challenge. However, the down-ranking method is not designed to handle data from multi-gene KO experiments. Meanwhile, TRaCE outperformed GENIE3 [42] and TIGRESS [45] 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.

To address this issue, we further developed REDUCE [79], an algorithm that designs optimal knockout experiments for reducing the uncertainty in the network, as presented in Chapter 3. In particular, based on the upper and lower bounds of an ensemble from TRaCE or TRaCE+, REDUCE provides the KO experiments that could verify the maximum number of uncertain regulatory interactions. The case studies demonstrated that iterative inference can successfully infer the true GRN from error-free data. The iterative inference can also resolve GRN inferability and produce a single GRN using simulated noisy expression data from the designed experiments. In this case, TRaCE+ required much fewer KO experiments to identify a unique GRN than TRaCE, because of its ability to utilize information within the edge signs.

Nevertheless, there were some errors in the predicted networks. False negative errors were more dominant than false positive errors. In particular, most false negative errors were associated with fan-in motifs where two or more genes regulate a target gene. It is
fundamentally challenging to infer such motifs from steady-state gene knock out data because perturbing one of the regulators may cause no effect on the target gene due to compensation by the other gene regulator(s). One could potentially address this issue of false negative by considering other types of data, for example time series gene expression data and transcription factor binding sites.

Most of the errors in the predicted bounds were already present in the bounds estimated by TRaCE or TRaCE+. These errors indicated that TRaCE and TRaCE+ partially succumbed to the difficult challenge of GRN inference from realistic expression data. Additionally, TRaCE and TRaCE+ are limited by their requirement of all single gene KO experiment for estimating the initial bounds. Nevertheless, if the TFs are known, then only KO experiments of TFs are sufficient to estimate the bounds. We expect current developments in high-throughput gene editing (e.g. CRISPR/Cas9) will soon provide the technology to overcome these challenges. Future developments may also lead to the integration of several rapidly growing technologies within the framework of ensemble inference and design of experiment. For example, the progress in the automation of next generation sequencing and multiplexed gene editing tools [100] can be combined with the iterative inference strategy based on TRaCE+ and REDUCE permitting an automated inference of gene regulatory networks.

Future research may also expand the inferability analysis of GRNs beyond the scope of this thesis. For instance, inferability of gene regulatory networks from time-series gene-expression data was out of the scope of this thesis. Analysis of GRN inferability from time-series data and optimal design of time-course experiments for GRNI are likely to open new avenues in identifying GRNs. We hope that the adoption of ensemble based inferability analysis and design of experiments will facilitate future methodologies to overcome the challenges of GRN inference.
Appendix A

A1 Implementation of TRaCE
A1.1 Example of ConTREx

An example of the application of the ConTREx procedure is shown in Fig. A1. The directed edges indicated by blue arrows in $\bar{G}_0$ are the set of indirect regulations.

Figure A1 An example of ConTREx of a simple GRN. 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.
A1.2 Construction of ensemble involving directed cycles

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 \( C \rightarrow A \) belongs to the lower bound and \( A \rightarrow C \) is not in the upper bound. In this case, the ensemble comprises the graphs shown in Fig. A2 (a)-(d).

![Figure A2](image)

**Figure A2** Example of an ensemble involving GRN with cycle.

A1.3 Preprocessing of \( \mathcal{G} \)

In general, accessibility matrices inferred from noisy measurements will have FP and FN errors. To correct errors in \( \mathcal{G} \), we have made the assumption that errors rarely occur for the same edge among input accessibility matrices. Hence, an edge that appears in a majority of the input matrices (beyond a certain threshold) was added to the \( \mathcal{G} \), or was otherwise removed from \( \mathcal{G} \). This procedure is implemented as follows:

\[
A = \text{Acc}(\mathcal{G}) + \sum_i \text{Acc}(\mathcal{G}_{i})
\]

\[
\text{Acc}_{i,j}(\mathcal{G}) = \begin{cases} 
1 & \text{if } A_{i,j} \geq \text{threshold} \\
0 & \text{otherwise}
\end{cases}
\]

Here, \( \mathcal{G} \) denotes the corrected \( \mathcal{G} \). In our experience, we could obtain good results with threshold \( = 0.65 \times N_{\text{acc}} \), where \( N_{\text{acc}} \) is the number of input accessibility matrices. In
general, the performance of TRaCE with error correction was not sensitivity for threshold values in the range between $0.5 \times N_{acc}$ and $0.8 \times N_{acc}$.

A1.4 Filter Algorithms

A1.4.1 Filter for $\tilde{G}^L$

The filter algorithm below is designed to prevent the propagation of FP edges in any $G_{[i]}$ to the lower bound $\tilde{G}^L$. The pseudocode for the union with filter $\bigcup$ is given below:

$$\text{Adj}(\tilde{G}^L) = \text{Adj}(\tilde{G}_o^L)$$

$$A = \text{Acc}(\tilde{G}_o^L)$$

FOR every knockout experiment $k$

$$A_{\text{test}} = \sum_{i \in V_{ko}} A_i \otimes A'$$

$$\text{Adj}(\tilde{G}^L) = \text{Adj}(\tilde{G}^L) + A_{\text{test}} \cdot \text{Adj}(G_{v_{ko}}^L)$$

END FOR

where $\otimes$ denotes the outer product and $\cdot$ denotes the Hadamard multiplication (element wise multiplication). Note that the elements of the matrices are either 0 or 1, and the addition operation $+$ denotes a Boolean sum.

A1.4.2 Filter for $\tilde{G}^U$

The following filter is created to avoid propagating FN edges in any $G_{[i]}$ to $\tilde{G}^U$. The pseudocode for the intersection with filter $\bigcap$ is given below:

$$\text{Adj}(\tilde{G}^U) = \text{Acc}(\tilde{G}_o^U)$$

$$A = \text{Acc}(\tilde{G}_o^U)$$

FOR each knockout experiment $k$

$$A_{\text{test}} = \sum_{i \in V_{ko}} A_i \otimes A'$$

$$\text{Adj}(\tilde{G}^U) = \text{Adj}(\tilde{G}^U) \cdot A_{\text{test}} \cdot \text{Adj}(G_{v_{ko}}^U)$$

END FOR
A1.5 Consistency check

The set of inconsistent edges can be readily identified by the set difference $\tilde{G}^L - \tilde{G}^U$. An inconsistent edge arises due to conflicting evidence about the existence or non-existence of an edge. This means that for some input accessibility matrices, this particular edge is a member of both $\tilde{G}$ and $G$, while for other input matrices, the edge is absent from $\tilde{G}$ and $G$. To resolve inconsistent edges, we take a vote among input matrices that are in support for or against the existence of each of these edges. When an inconsistent edge receives a majority positive vote (supporting its existence), the upper bound $\tilde{G}^U$ is updated to include the edge. Otherwise, this edge is removed from $\tilde{G}^L$. In case of a tie vote, the edge is removed from $\tilde{G}^L$ and added to $\tilde{G}^U$ such that $\tilde{G}^L \subset \tilde{G}^U$. The pseudocode for the consistency check algorithm (CC) is given below:

FOR each inconsistent edge $(i,j)$

$$T_{\text{vote}} = 0$$

$$F_{\text{vote}} = 0$$

FOR each knockout experiment $k$

IF $\text{Adj}(\tilde{G}_{v_{ko}})_{i,j} = 1$ AND $\text{Adj}(G_{v_{ko}})_{i,j} = 1$ THEN $T_{\text{vote}} = T_{\text{vote}} + 1$

ELSEIF $\text{Adj}(\tilde{G}_{v_{ko}})_{i,j} = 0$ AND $\text{Adj}(G_{v_{ko}})_{i,j} = 0$ THEN $F_{\text{vote}} = F_{\text{vote}} + 1$

END IF

END FOR

IF $T_{\text{vote}} > F_{\text{vote}}$ THEN $\text{Adj}(\tilde{G}^U)_{i,j} = 1$

ELSEIF $T_{\text{vote}} < F_{\text{vote}}$ THEN $\text{Adj}(\tilde{G}^L)_{i,j} = 0$

ELSE $\text{Adj}(\tilde{G}^U)_{i,j} = 1$ AND $\text{Adj}(\tilde{G}^L)_{i,j} = 0$

END IF

END FOR
A1.6 Type A errors

As discussed in the text, some Type A errors due to FN errors in the input accessibility matrices are transmitted to the lower and upper bound. While it is intuitive that FP errors in any input matrices could lead to erroneous edges in the lower bound (see an example in Fig. A4), FNs can also give rise to the same kind of errors. An example of this scenario is illustrated in Fig. A3(a)-(e). Here, we focus on the edge $A \rightarrow D$ (blue arrow). Based on the accessibility matrix of $G_\emptyset$ in Fig. A3(b), the edge $A \rightarrow D$ is deemed testable using the accessibility matrix of the GRN with a KO gene $B$. In other words, this edge maybe verifiable when incorporating $G_{\emptyset}^B$ and its ConTREx, $\tilde{G}_{\emptyset}^B$ (see Fig. A3(c)-(d)). Let us consider an FN error associated with a missing edge $C \rightarrow D$ in the measured $\tilde{G}_{\emptyset}^B$, denoted by $G_{\emptyset}^M$. Because of this error, $A \rightarrow D$ edge which should have been removed in the ConTREx, now erroneously appears in $G_{\emptyset}^M$, as shown in Fig. A3 (e). Since $A \rightarrow D$ is a testable edge, the $A \rightarrow D$ error in $G_{\emptyset}^M$ will pass through the filter. In general, an erroneous edge between two nodes can appear in the output of ConTREx due to an FN error, when this error breaks all indirect path(s) between the two nodes. In addition, if such an error occurs in the input matrix involving KO of genes on any path of these two nodes, the erroneous edge will be deemed testable and therefore will pass through the filter algorithm.

There are two circumstances that an FP error can compensate Type A errors originated from an FN in the above example. The first is when the FP errors lead to the creation of cycle(s) to which the Type A error above is incident. For the example above, an FP involving $C \rightarrow A$ in $G_{\emptyset}^M$ will form a directed cycle between nodes $A$ and $C$ (see Fig. A3(f)) and the ConTREx of this input is shown in Fig. A3(g). In this case, the edge $A \rightarrow D$ is removed as this edge is incident to a cycle. Alternatively, an FP can create a new indirect path between the nodes associated with the Type A error edge above. In the simple example, when the FN in $G_{\emptyset}^M$ affecting $C \rightarrow D$ edge is accompanied with an FP of the edge $C \rightarrow E$ (see Fig. A3(h)), the
edge $A \rightarrow D$ will be pruned during ConTREx and thus will not appear in $G_{M}^{M}$. Finally, the difference between Type A error in $\tilde{G}^{c}$ induced by FP and FN is that the FP error needs to occur for the affected edge (see Fig. A4), while the number of scenarios for which this error occurs by way of FN is more numerous. For this reason, there are more Type A errors due to FN that escape the correction algorithm than those due to FP.

**Figure A3** Type A errors due to an FN. (a) $G_{\varnothing}$; (b) $\tilde{G}_{\varnothing}$; (c) $\tilde{G}_{\{B\}}$; (d) $G_{\{B\}}$; (e) $\tilde{G}_{\{B\}}^{M}$ with an FN at $C \rightarrow D$. In this case, $G_{\{B\}}^{M} = \tilde{G}_{\{B\}}^{M}$. (f) $\tilde{G}_{\{B\}}^{M}$ with an FN at $C \rightarrow D$ and an FP at $C \rightarrow A$, and (g) its ConTREx reduction. (h) $\tilde{G}_{\{B\}}^{M}$ with an FN at $C \rightarrow D$ and an FP at $C \rightarrow E$ and (i) its ConTREx reduction.
**Figure A4** Type A error due to an FP. $G_\emptyset$ and $\overline{G}_\emptyset$ are shown in Fig. A3 (a) and (b), respectively. (a) $\overline{G}_{(c)}$. In this case, $G_{(c)} = \overline{G}_{(c)}$. (b) $\overline{G}_{(c)}^M$ with an FP at $B \rightarrow D$. Here, $G_{(c)}^M = \overline{G}_{(c)}^M$.

### A1.7 Structural Hamming Distance

<table>
<thead>
<tr>
<th>FPR</th>
<th>FNR</th>
<th>Before Correction</th>
<th>After Correction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$D(G_\emptyset, \overline{G}^L)$</td>
<td>$D(\overline{G}^L, G_\emptyset)$</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>63.5</td>
<td>54.6</td>
</tr>
<tr>
<td>0.0</td>
<td>0.1</td>
<td>189.4</td>
<td>85.9</td>
</tr>
<tr>
<td>0.0</td>
<td>0.2</td>
<td>191.7</td>
<td>110.8</td>
</tr>
<tr>
<td>0.1</td>
<td>0.0</td>
<td>63.8</td>
<td>1490.5</td>
</tr>
<tr>
<td>0.1</td>
<td>0.1</td>
<td>188.5</td>
<td>1510</td>
</tr>
<tr>
<td>0.1</td>
<td>0.2</td>
<td>191.6</td>
<td>1525.9</td>
</tr>
<tr>
<td>0.2</td>
<td>0.0</td>
<td>63.8</td>
<td>2422.2</td>
</tr>
<tr>
<td>0.2</td>
<td>0.1</td>
<td>189.8</td>
<td>2440.0</td>
</tr>
<tr>
<td>0.2</td>
<td>0.2</td>
<td>191.5</td>
<td>2469.2</td>
</tr>
</tbody>
</table>

**Table A1** Performance of TRaCE on inference of *E. coli* subnetworks (n=100 genes). The reported values represent the average over 50 subnetworks. Let $D(A, B)$ 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 [101].
<table>
<thead>
<tr>
<th>FPR</th>
<th>FNR</th>
<th>Before Correction</th>
<th>After Correction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$D(G_2, \tilde{G}_U)$</td>
<td>$D(\tilde{G}_L, G_2)$</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>2132</td>
<td>1193</td>
</tr>
<tr>
<td>0.0</td>
<td>0.1</td>
<td>3661</td>
<td>2160</td>
</tr>
<tr>
<td>0.0</td>
<td>0.2</td>
<td>3752</td>
<td>2671</td>
</tr>
<tr>
<td>0.1</td>
<td>0.0</td>
<td>2134</td>
<td>36408</td>
</tr>
<tr>
<td>0.1</td>
<td>0.1</td>
<td>3676</td>
<td>36922</td>
</tr>
<tr>
<td>0.1</td>
<td>0.2</td>
<td>3749</td>
<td>37171</td>
</tr>
<tr>
<td>0.2</td>
<td>0.0</td>
<td>2134</td>
<td>63975</td>
</tr>
<tr>
<td>0.2</td>
<td>0.1</td>
<td>3680</td>
<td>64348</td>
</tr>
<tr>
<td>0.2</td>
<td>0.2</td>
<td>3746</td>
<td>64701</td>
</tr>
</tbody>
</table>

**Table A2** Performance of TRaCE on inference of *E. coli* GRN. Let $D(A, B)$ of any two digraphs $A$ and $B$ denote the structural Hamming distance (SHD) between them.
Appendix B

B1. Optimization Algorithm in REDUCE

As mentioned in the Chapter 3, the optimization of gene KO experiment in REDUCE is performed separately for each set of edge separatoids $S_q$ ($q = 1, 2, 3$). For each $S_q$, we employ genetic algorithm (GA) to find the optimal sets $V_{KO}$ and $E_{T,q}(V_{KO})$. Below we provide the pseudo-code for the implementation of GA in REDUCE.

B1.1 Genetic algorithm for optimizing $V_{KO}$ given $S_q$

Input: $S_q$

Output: $V_{KO}$

Parameters: $pop_size$ (population size, default 100), $num_gen$ (number of generations, default 50)

1) Set the initial population of $V_{KO}$ to $S_q$ and set the generation count to 0
2) While generation count is less than or equal to $num_gen$
   a) For each $V_{KO}$ in the population, compute the number of edges in $E_{T,q}(V_{KO})$ (see Section B1.2)
   b) Sort the population of $V_{KO}$ based on the number of edges in $E_{T,q}(V_{KO})$
   c) Select the top $pop_size$ $V_{KO}$ from the population as parents
   d) Mate the parent population to generate $pop_size$ number of offsprings (see Section B1.3)
   e) Mutate the offspring population (see Section B1.4)
   f) Combine the parent and offspring population of $V_{KO}$
   g) Increase generation count by 1
B1.2 Calculation of the cardinality of $E_{T,q}(V_{KO})$

Input: $S_q, V_{KO}$

Output: the number of edges in $E_{T,q}(V_{KO})$

Parameters: $max\_ko$ (maximum number of genes in $V_{KO}$), $essential\_set$ (the set of genes to be excluded from $V_{KO}$)

1) If the number of genes in $V_{KO}$ is larger than $max\_ko$, then set output to 0.
2) If $V_{KO}$ contains any gene in $essential\_set$, then set output to 0.
3) Set count to 0
4) For each $(i, j)$ where $S_q(i, j) \in S_q$
   a) If $(i \notin V_{KO} \& j \notin V_{KO} \& S_q(i, j) \subseteq V_{KO})$ then increase count by 1
5) Return count as output

B1.3 Parent mating algorithm

Input: Parent population of $V_{KO}, pop\_size$

Output: Offspring population of $V_{KO}$

Parameter: $num\_block$ (number of division blocks during crossover, default 6. This should be an even number.)

1) For count = 1 until $pop\_size$
   a) Randomly select two different $V_{KO}$ from the parent population
   b) Convert each parental $V_{KO}$ to a binary string of length $n$ (number of genes), where the $i$-th element is set to 1 when gene $i \in V_{KO}$
   c) Divide each parental string into $num\_block$ number of equal length substrings
   d) Randomly select $num\_block/2$ non-overlapping substring from each parent
   e) Combine the selected substring in step 4 to form an offspring
   f) Convert the offspring binary string to give offspring $V_{KO}$
B1.4 Offspring mutation algorithm

Input: Offspring population of $V_{KO}$

Output: Mutated offspring population of $V_{KO}$

Parameter: $mut\_rate$ (mutation rate, default 0.1)

1) Convert each offspring $V_{KO}$ to a binary string

2) For each $V_{KO}$ string
   a) For every element in a $V_{KO}$ string
      i) Take a random number between 0 and 1 from a uniform distribution
      ii) If the random number $\leq mut\_rate$, then flip the corresponding value of the element
          (from 1 to 0, or from 0 to 1)
   b) Convert the string back to $V_{KO}$

3) Return the population of mutated offspring $V_{KO}$

B2. Iterative Inference Procedure with Multiplexing

When multiplex assay (e.g. RNA sequencing) is available, we can analyze a large number of samples in parallel. Fig. B1 below describes the modified iterative network inference, which allow the use of multiplex technology. In contrast to the original iterative procedure described in the main text, in each iteration we perform REDUCE in a loop without ensemble bound update to generate a series of $\{V^i_{KO,l}\}$, corresponding to optimal gene KO combinations with non-increasing $N(E_{T,q}(V^i_{KO,l}))$. This step is shown in the inner loop in Fig. B1. The loop is performed until the set of uncertain edges that have not been accounted among the optimal designs thus far is exhausted or until no feasible solution can be found or until a desired number of KO experiments are generated. If only a subset of the generated series $\{V^i_{KO,l}\}$ can be
handled in a given iteration, then one should perform the KO experiments prescribed by \( \{ V_{KO,l} \} \) in the order of \( l (l = 1, 2, \ldots, L) \).

**Figure B1** Modified Iterative Network Inference for Multiplex Assay.

We applied the modified iterative procedure to the five 100-gene gold standard networks from DREAM4 network inference challenge, in the manner described in Section 3.3.3 of the main text. Again, we implemented the iterations with increasing cardinality of \( V_{KO} \), starting with \( V_{KO} \) containing only 1 gene (i.e. double KO experiments). We performed REDUCE in a loop until no feasible solution could be found. Subsequently, we used GeneNetWeaver to generate the data for all experiments prescribed by \( \{ V_{KO,l} \} \) and \( \{ E_{T,q}(V_{KO,l}) \} \), emulating the use of multiplex assay. The ensemble bounds update was done as described in the main text using the generated data. We carried out the above steps until \( G^U \) and \( G^L \) converged. Similar to the application of the original iterative procedure, the modified procedure produced a unique network using up to 3 gene KOs. Again, for network 2, we only needed 2 gene KOs. As illustrated in Fig. B2, the identified GRNs were of similar accuracy in comparison to those
from the original procedure. As expected, the modified procedure with multiplexing was able to reduce the number of iterations (~5 fold decrease), but at the cost of a slight increase in the total number of KO experiments (see Table B1).

![Figure B2](image)

**Figure B2** Comparison of Multiplexed Iterative Procedure to Original Procedure and Ancestor-Descendant (AD) Pair on DREAM 4 100-gene networks: (a) number of uncertain edge verifications and (b) total network distance. (c) Jaccard distance of $G^U$ and $G_{\emptyset}$. (d) Jaccard distance of $G^L$ and $G_{\emptyset}$.

<table>
<thead>
<tr>
<th>Network</th>
<th>Number of Iterations using DOE</th>
<th>Number of KO Experiments using DOE</th>
<th>Number of Iterations using Multiplex DOE</th>
<th>Number of KO Experiments Multiplex DOE</th>
<th>Number of KO Experiments using AD pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>34</td>
<td>48</td>
<td>7</td>
<td>46</td>
<td>41</td>
</tr>
<tr>
<td>2</td>
<td>21</td>
<td>22</td>
<td>5</td>
<td>25</td>
<td>22</td>
</tr>
<tr>
<td>3</td>
<td>55</td>
<td>106</td>
<td>10</td>
<td>113</td>
<td>57</td>
</tr>
<tr>
<td>4</td>
<td>80</td>
<td>120</td>
<td>14</td>
<td>107</td>
<td>64</td>
</tr>
<tr>
<td>5</td>
<td>54</td>
<td>77</td>
<td>11</td>
<td>100</td>
<td>57</td>
</tr>
</tbody>
</table>

**Table B1** Number of iterations and total KO experiments for the inference of 100-gene DREAM 4 GRNs
B3. Ensemble Bounds Update under Ideal Conditions

We employ the following procedure for updating the ensemble bounds $G^{U,k}$ and $G^{L,k}$ from the $k$-th iteration to the $k+1$-th iteration under ideal conditions.

1) Given $G^{U,k}$ and $G^{L,k}$, use REDUCE to find $V_{K0}$

2) Construct $G_{V_{K0}}$ from $G_0$

3) Assign $A$ as an $n \times n$ zero matrix ($n$: number of genes)

4) For $i$ from 1 to $n$

   a) if $i \in I^*$, then set the $i$-th row of $A$ equal to the $i$-th row of Acc($G_{V_{K0}}$) (this step corresponds to perturbing gene $i$ in the background of $V_{K0}$ knock-out. Under ideal conditions, we do not have any error in determining the genes that direct and indirectly regulated by gene $i$ in the digraph $G_{V_{K0}}$.)

   b) if $i \notin I^*$, then set the $i$-th row of $A$ equal to the $i$-th row of Acc($G^{U,k}$)

5) Perform ConTREx on the matrix $A$ and set the output as $A_t$

6) Set $G^{U,k+1}$ equal to $G^{U,k}$ and $G^{L,k+1}$ equal to $G^{L,k}$

7) For each uncertain edge $(i, j) \in E_{T,q} (V_{K0}^*)$

   a) if $A(i, j) = 0$, then remove the edge $(i, j)$ from $G^{U,k+1}$ (the uncertain edge $(i, j)$ is verified as non-existent)

   b) if $A(i, j) = 1$, then add the edge $(i, j)$ to $G^{L,k+1}$ (the uncertain edge $(i, j)$ is verified as existent)
Bibliography: