TORSTEN GROSS

NETWORK INFERENCE FROM PERTURBATION DATA:

ROBUSTNESS, IDENTIFIABILITY AND EXPERIMENTAL DESIGN

NETWORK INFERENCE FROM PERTURBATION DATA:
ROBUSTNESS, IDENTIFIABILITY AND EXPERIMENTAL DESIGN

DISSERTATION

zur Erlangung des akademischen Grades
Doctor rerum naturalium (Dr. rer. nat.)

von

M. Sc. Torsten Groß

eingereicht an der
Lebenswissenschaftlichen Fakultät der Humboldt-Universität zu Berlin

Präsidentin der Humboldt-Universität zu Berlin
Prof. Dr.-Ing. Dr. Sabine Kunst

Dekan der Lebenswissenschaftlichen Fakultät
Prof. Dr. Bernhard Grimm

Gutachter

1. Prof. Dr. Nils Blüthgen
2. Prof. Dr. Ing. Julio Saez-Rodriguez
3. Dr. Ralf Steuer

Tag der mündlichen Prüfung: 12. November 2020

Torsten Groß: *Network Inference from Perturbation Data: Robustness, Identifiability and Experimental Design,* June 2020

ABSTRACT

'Omics' technologies offer detailed insights into cellular composition. Yet, they rarely characterize interactions between the components of a biological system. To fill this gap, a wide array of network reconstruction methods has been developed over the past twenty years. Amongst them are algorithms that derive networks from data that describes a system's response to targeted perturbations. This information allows reconstruction methods to deduce causal interaction chains. In this way, they can reveal functional mechanisms in gene regulation, signal transduction, intra-cellular communication and many other cellular processes. Nevertheless, the problem of reverse engineering of biological networks remains essentially unsolved because inferred networks are often based on inapt assumptions, lack interpretability as well as a rigorous description of identifiability. This thesis attempts to overcome these shortcomings.

First, it presents a novel inference method which is based on a simple response logic. The underlying assumptions are so mild that the approach is suitable for a wide range of applications while also outperforming existing methods in standard benchmark data sets. Being implemented within a powerful Answer Set Programming framework, the response logic approach can easily incorporate prior network knowledge and then reveal all networks that conform to the given data. This provides an explicit display of identifiability of individual network links. These qualities were critical for the derivation of plausible network hypotheses from RPPA perturbation data, describing MAPK and PI3K signalling pathways in an adenocarcinoma cell line. The inferred networks could explain distinct sensitivities of different PI3K mutants towards targeted inhibitors. The flexibility and clear interpretability of the response logic approach makes it a versatile and useful framework to gain mechanistic insights not only in signal transduction but in various biological systems.

A second study shows that the identifiability of interaction strengths in linear response networks can be described by an intuitive maximum-flow problem. This analytical result not only allows to devise identifiable effective network models in underdetermined settings but also to optimize experimental design, that is to choose the most effective perturbation targets. Based on the maximum-flow approach, an algorithm was designed that determines the sequence of perturbations, which maximizes the number of uniquely inferable interaction strengths. Benchmarked on a database of human pathways, it achieved full network identifiability on average with less than a third of the perturbations that are needed in a random experimental design. Moreover, allowing for perturbation combinations further reduced this fraction to less than one quarter. As perturbation experiments are often challenging and costly, these improvements can be crucial for a comprehensive characterization of biological networks.

Finally, this thesis presents mathematical advances in Modular Response Analysis (MRA), which is a popular network inference framework that quantifies interaction strengths between network components from perturbation data. In practical applications of MRA, it is important to be able to incorporate prior network knowledge and to allow for multi-target perturbations. In this general setting, the inference of MRA network parameters becomes a hard, non-linear optimization problem, which currently limits the size of inferable networks to the low tens. Yet, it is here shown that under a certain independence assumption this optimization problem can be formulated as a total least squares problem, whose solution is derived analytically and can be robustly evaluated with negligible computational effort. However, with increasing levels of measurement errors the independence assumption breaks down and the total least squares solution becomes imprecise. Nevertheless, it still resides in the vicinity of the global optimum and thus provide an excellent initial condition for a subsequent iterative optimization. In a benchmark on synthetic perturbation experiments on human pathways, this approach drastically improved the computational performance compared to the current standard procedure. This could be an essential step to enhance MRA's capacity to model bigger networks and to handle the next generation of large-scale perturbation data.

ZUSAMMENFASSUNG

Moderne Hochdurchsatzverfahren der Molekularbiologie liefern ein detailliertes Bild zellulärer Zusammensetzung. Allerdings können sie typischerweise keine Aussagen über Interaktionen zwischen den Komponenten eines biologischen Systems treffen. Deshalb wurde in den letzten 20 Jahren ein breites Sortiment an Netzwerk-Rekonstruktionsmethoden entwickelt. Darunter befindet sich eine Klasse von Algorithmen, die Perturbationsdaten analysiert. Solche Daten beschreiben wie ein biologisches System auf gezielte Störungen an einzelnen Komponenten reagiert. Diese Information erlaubt es den Rekonstruktionsmethoden Rückschlüsse über kausale Interaktionsketten zu ziehen. Auf diese Weise konnten funktionelle Mechanismen in der Genregulierung, in der Signal Transduktion, in intra-zellulärer Kommunikation und vielen anderen zellulären Prozessen aufgedeckt werden. Dennoch bleibt das Problem der Netzwerkinferenz im Kern ungelöst. Die Rekonstruktion der Netzwerke basiert häufig auf ungeeigneten Annahmen, oft ist die Frage der Identifizierbarkeit einzelner Netzwerkkanten ungeklärt und die rekonstruierten Netzwerke sind schwer interpretierbar. Das Ziel der Dissertation ist es diese Probleme anzugehen.

Zunächst beschreibt sie eine neue Netzwerk-Rekonstruktionsmethode, die auf einer einfachen Annahme von Perturbationsausbreitung basiert. Damit ist die Response Logic Methode in verschiedensten Zusammenhängen anwendbar und übertrifft andere Methoden in Standard-Benchmarks. Der Algorithmus wurde in einem Answer Set Programming Framework implementiert. Das erlaubt eine einfache Integration von Vorkenntnissen über das zu inferierende Netzwerk. Außerdem ermöglicht es die Bestimmung aller Netzwerke, die den vorgegebenen Daten genügen. Dies zeigt die Identifizierbarkeit jeder einzelnen Netzwerkkante explizit auf. Aufgrund dieser Eigenschaften konnte die Response Logic Methode plausible Netzwerkhypothesen aus RPPA Daten über MAPK und PI3K Signalkaskaden in einer Adenokarzinom-Zellline generieren. Die inferierten Netzwerke erlauben es die unterschiedlichen Sensitivitäten von PI3K-Mutanten gegenüber verschiedener Inhibitoren überzeugend zu erklären. Ihre Flexibilität und leichte Interpretierbarkeit machen die Response Logic Methode zu einem wirkungsvollem Werkzeug zur Rekonstruktion von Netzwerken in verschiedensten biologischen Prozessen.

Ein zweites Projekt untersuchte die Identifizierbarkeit von Interaktionsstärken zwischen Komponenten in Netzwerken, für die lineare Perturbationseffekte angenommen werden können. Hierbei konnte gezeigt werden, dass sich die Frage nach Identifizierbarkeit auf ein Max-Flow Problem abbilden lässt. Dieses analytische Resultat erlaubt es effektive, identifizierbare Netzwerk-Modelle für ursprünglich unterdeterminierte Inferenzprobleme zu bestimmen. Weiterhin ist es damit möglich das experimentelle Design zu optimieren. Basierend auf der Max-Flow Formulierung wurde ein Algorithmus entwickelt, der Sequenzen von Perturbationen bestimmt, die die Anzahl an identifizierbaren Interaktionsstärken maximieren. Dieser wurde auf einer Reihe von bekannten regulatorischen Netzwerken getestet. Im Vergleich zu zufällig generierten Perturbationssequenzen, konnte die durchschnittliche Anzahl der für volle Identifizierbarkeit notwendigen Perturbationen dabei auf unter ein Drittel gesenkt werden. In dem Fall, dass Perturbationen auch kombiniert werden können, reduziert sich dieser Anteil sogar auf unter ein Viertel. Weil Perturbationsexperimente oft sehr aufwändig und kostspielig sind, kann diese Optimierung ein entscheidender Schritt zur vollständigen Charakterisierung biologischer Netzwerke sein.

Schließlich beschreibt die Dissertation eine mathematische Weiterentwicklung der Modular Response Analysis (MRA). Dies ist eine populäre Methode zur Quantifizierung von Interaktionsstärken zwischen Netzwerkkomponenten aus Perturbationsdaten. In deren praktischen Anwendung ist es häufig notwendig, Vorwissen über die Netzwerkstruktur mit einbauen zu können, sowie Perturbationen mit multiplen Zielen zu berücksichtigen. In diesem allgemeinen Fall muss zur Bestimmung der Interaktionsstärken ein aufwändiges, nicht-lineares Optimierungsproblem gelöst werden, was die maximale Größe inferierbarer Netzwerke auf deutlich unter 50 beschränkt. In diesem Zusammenhang kann hier gezeigt werden, dass sich das Problem unter einer bestimmten Unabhängigkeitsannahme als orthogonale Regression darstellen lässt. Deren Lösung konnte analytisch bestimmt werden und lässt sich mit vernachlässigbaren nummerischen Aufwand auswerten. Allerdings wird die Unabhängigkeitsannahme mit wachsenden Messfehlern ungültig und das Ergebnis der orthogonalen Regression ungenau. Jedoch verbleibt die Lösung weiterhin in der Nähe des globalen Optimums und stellt damit einen idealen Anfangswert für eine anschließende iterative Optimierung dar. In Testläufen auf synthetische Perturbationsdaten zeigt dieser Ansatz eine dramatische Steigerung der nummerischen Effizienz. Dies könnte eine maßgebliche Verbesserung von MRA sein, die es ermöglicht größere Netzwerke zu rekonstruieren und somit die neuesten Technologien zur Durchführung von Hochdurchsatz-Perturbationensexperimenten auszunutzen.

CONTENTS

1 PREFACE

This cumulative dissertation presents two publications (Gross et al. 2019; Gross et al. 2020) that resulted from my doctoral studies on network inference. Here, these are summarized and embedded in a larger scientific context. An extensive introduction in Chapter 2 portrays the historical developments that led to a demand for the reverse engineering of biomolecular networks and reviews existing approaches and applications. This reveals the current shortcomings that motivated my research. Chapters 3 and 4 present the two publications and discuss how my studies helped to address these challenges. In addition, Chapter 5 provides some previously unpublished mathematical analysis to improve an established network reconstruction method. While these results are preliminary, they fit well into the general context of the thesis and therefore make an appropriate addition. The thesis concludes with a critical review of the obtained results and provides an outlook on open question.

In all my projects, I carried out all mathematical analyses, developed the algorithms and wrote the manuscripts. Matthew J. Wongchenko and Yibing Yan provided the experimental data for the response logic project and Nils Blüthgen edited the manuscripts and provided and excellent supervision for the entirety my work. Thank you Nils for all the inspiration, your consistent support and for creating an intellectual environment that stimulates a lively exchange of ideas. It has been a tremendous experience working with you.

I also owe my scientific accomplishments to the support of many more people. I am indebted to Bertram for all of his invaluable explanations and insights, as well as for his patience and kindness. Thank you Florian, Johannes and Mattias for all the witty conversations. I am grateful to Manuela and Mathurin for our inspiring meetings. And without Katinka, I would not have discovered the power of Answer Set Programming. I also want to thank my graduate schools Computational Systems Biology (Research Training Group GRK1772) and *Comp*Cancer (RTG2424) for their generous support. These programs allowed me to connect to an international community of scientists and broadened my horizon. Thank you Edda, Marylu, Cordelia and (once again) Manuela for creating such a fun learning environment. I am also very grateful to all members of my dissertation committee for the effort invested in the evaluation of this thesis. Finally, none of this would have been possible without the wholehearted encouragement by Naïma, my friends, and my dear family.

2 INTRODUCTION

This thesis aims to contribute to the field of network inference. Network inference is a process that derives a network models from observations of the state of a system. The need for network inference methods in biology arose with the onset of the 'Omics' revolution at the beginning of the 2000s. Within a few years, a series of technological innovations enabled the experimental observation of a wide array of molecular cell components at an unprecedented scale. Yet, the ensuing wealth of data did not per se deliver tangible insights into the workings of the cell because it does not reveal the connections between cell components. Network inference methods attempt to fill this gap and explain the system's behaviour in terms of the interaction of its components. This perspective on biological complexity can lead to biological insights and reliable predictions.

The first part of the introduction describes the scientific and historical context that called for reverse engineering of biological networks. This clarifies the motivation behind the various types of inference methods, which are reviewed in its second part. Finally, such overview allows to discuss some of the current challenges of network reconstruction, which initiated the work that is presented in later chapters.

2.1 THE HISTORICAL CONTEXT OF BIOLOGICAL NETWORK INFERENCE

Network models in an era of low-throughput data

Mathematical network models were formulated long before the availability of 'Omics' technologies. These pre-millennial achievements are exhaustively reviewed elsewhere (Bailey 1998; Green 2016; Buchman 2002; Wolkenhauer 2001), but to point out the pivotal role of 'Omics' data for the analysis of biological networks, it is worthwhile to mention some early key developments.

The 1940s marked the beginning of the commercial manufacturing of antibiotics, which established the field of biochemical engineering. This gave a strong impetus towards a formalized descriptions of metabolic circuits in microorganism, as they became crucial for the industrial production of vaccines, insulin, biofuels and many other compounds (Bailey et al. 1986). An important theoretical achievement in this context is Metabolic Control Analysis (Kacser et al. 1973; Heinrich et al. 1974), a mathematical framework that allows to quantify the extend to which enzymes control the flux and concentration of metabolites in a metabolic pathway. Later, biochemistry started to be viewed from a cybernetic perspective as well. This allowed to formulate reaction networks without needing to specify their kinetic parameters. The idea is that evolution meticulously refined biochemical networks towards a specific function and thereby constrained their kinetic parameters. Plausible parameter values can thus be retrieved by solving optimization problems for the formulated model. For example, the maximization of cell mass production in a reaction network describing microbial growth allowed to quantify the allocation of critical resources to various key proteins (Dhurjati et al. 1985). Similar ideas amounted to the development of Flux Balance Analysis (Fell et al. 1986). Kinetic models also became popular to describe signalling pathways (Lauffenburger et al. 1996), for which computer simulations could reveal emergent properties, such as "integration of signals across multiple time scales, generation of distinct outputs depending on input strength and duration, and self-sustaining feedback loops" (Bhalla et al. 1999).

Parallel to this research on reaction kinetics, there also was a development of logic models for the description of cellular circuits (Abou-Jaoudé et al. 2016). In his studies on generic (random) logical networks (Kauffman 1969; Glass et al. 1973), Stuart Kauffman provided theoretical results on how fundamental properties such as cell cycle duration or cell type diversity are linked to the size and structure of these networks. In contrast, René Thomas constructed specific logic networks to model lysis and lysogeny of the lambda phage (Thomas 1973; Thomas et al. 1976). Eventually, the logic formalism was increasingly refined and led to a wealth of studies (Kauffman 1993; Thomas et al. 1990).

These examples can broadly be categorized as bottom up approaches. That is, they assemble network models from a priori known interactions. A good network model then captures the interplay of all components appropriately and can, for example, predict the state of the entire system in a variety of environmental conditions or perturbations. The approach is especially effective when interactions are well characterized, as in metabolic pathways, where stoichiometry and enzyme kinetics govern the described reaction network. In contrast, the complexity of gene regulatory networks or signalling pathways generally precludes their description by a bottom up approach. Here, an interaction, for example between genes, represent a multitude of context-dependent biochemical and biophysical processes that cannot be easily cast into an applicable kinetic description. This is why many regulatory network models described before 2000 were either conceptional (such as Kauffman's random networks) or restricted to small, well-controllable systems (such as Thomas' lambda phage network). An alternative to overcome the problems of such reductionist descriptions of regulatory interactions is a top-down approach. This is the (inverse) idea to characterize interactions from observations of the global state of the system. However, such a reverse engineering approach relies on comprehensive data sets, which required technological innovations to overcome the often laborious and low-throughput nature of experimental techniques of the time.

The 'Omics' revolution

A continuously evolving branch of science has no clear starting point. Nevertheless, one could declare June 26, 2000 as the onset of the 'Omics' revolution. On that day, U.S. President Bill Clinton and the British Prime Minister Tony Blair jointly announced the completion of the first survey of the entire human genome. It was the end of a dogged scientific race between the publicly funded Human Genome Project, headed by Francis Collins, and the private company Celera Genomics founded by Craig Venter. The Human Genome Project was initiated in 1990 and aimed to sequence the human genome within 15 years. It was eight years later that Craig Venter set out to do the same, yet in three years only. The resulting shock fully manifested when Collins was told by Venter that in order to coordinate the efforts of the two projects "you can do mouse" (Shreeve 2007). In the end, the contestants tied and their landmark findings were published simultaneously (Lander et al. 2001; Venter et al. 2001). This event marked the beginning for series of technical innovations that brought about an unprecedented wealth of biological data. Since then, DNA sequencing costs first decreased at an exponential rate, and with the advent of next-generation sequencing (NGS) technologies (Slatko et al. 2018) around 2008, even at a super-exponential rate. In consequence, the cost per human genome has now reached the 1000 $ mark (Wetterstrand 2020).

Yet, the name 'Omics' was derived from the fact that technological progress not only improved the capacity to decipher genomes but to make a large variety of cellular components experimentally accessible. One of the particularly important developments was that it became increasingly easy to quantify RNA levels. A crucial step here was the invention of microarrays (Schena et al. 1995), which allowed to simultaneously assay thousands of transcripts at low costs. NGS technologies further revolutionized transcriptomics, as RNA levels could then be determined through cDNA sequencing on a massive scale, using RNA-Seq (Weber 2015). Compared to microarrays, this not only improved dynamic range and sensitivity but as RNA-Seq no longer relied on a predefined set of complementary oligonucleotides, it could also be used to detect transcription initiation sites, sense and antisense transcripts, alternative splicing events, and gene fusion (Vailati-Riboni et al. 2017).

Also high-throughput epigenomic measurements became available. An example is the use of chromatin immunoprecipitation (ChIP) to investigate binding between proteins and DNA. The combination with Microarrays (ChIP-chip) or NSG (ChIP-seq) then allowed to map histone-methylations over entire genomes (Barski et al. 2007) or to identify transcription factor binding sites (Johnson et al. 2007). Other techniques were developed to assay chromatin accessibility (e.g. ATAC-seq), chromatin interaction (e.g. Hi-C), DNA-methylation (Kurdyukov et al. 2016), and other DNA modifications (Stricker et al. 2017).

The 'Omics' revolution encompasses the proteome as well. In particular, it was the development of mass-spectroscopy that manifested high-throughput proteomics, which made the technology an indispensable tool for molecular and cellular biology (Aebersold et al. 2003). It's applications can be broadly divided into three major areas (Cox et al. 2011). There is expression proteomics, which aims to quantify the amount of proteins in a sample, there is the identification of post-translational modifications, and there is mapping of protein interactions (typically by a pull-down assay of a bait protein with its binding partners followed by mass-spectrometric analysis). Even though mass-spectroscopy remains a highly elaborate technology in comparison to other 'Omics' approaches, in the sense that is confined to a few specialized laboratories, it brought about a highly diverse range of biological insights, particularly due to its inherent specificity and sensitivity (Aebersold et al. 2016). A first complete model (yeast) proteome was presented in 2008 (Godoy et al. 2008) and first drafts of the human proteome followed in 2014 (Kim et al. 2014; Wilhelm et al. 2014). Concerning post-translational modifications, high-throughput phosphoproteomics has identified more than 230 000 phosphosites on 13 000 proteins in human (Vlastaridis et al. 2017) and suggests that many of them seem to be involved in cellular regulation (Sharma et al. 2014).

Besides technological developments that enabled high-throughput measurements of various types of cell components, recent years also saw the transition from bulk measurements to single cell experiments. The background to this development was that cellular heterogeneity had been recognized for decades (Rubin 1990; Elsasser 1984), which called the significance of measuring population averages into question. While cell ensemble measurements are reasonable when the cell-to-cell variance is simply due to noise (Elowitz et al. 2002; Ozbudak et al. 2002; Newman et al. 2006), they might not represent the biological state of any cell at all, if the ensemble is composed of distinct subpopulations (Ferrell et al. 1998) (consult Altschuler et al. 2010 for an extensive discussion). This triggered a range of innovations that opened 'Omics' technologies to the single-cell level. The first single-cell RNA-seq data was generated in 2009 (Tang et al. 2009), and followed by many more single-cell 'Omics' approaches (Stuart et al. 2019) to e.g.

reveal genome sequence (Navin et al. 2011), chromatin accessibility (Buenrostro et al. 2015), DNA methylation (Luo et al. 2017), or protein levels using mass-cytometry (Bandura et al. 2009). More recently, even more comprehensive experiments are realized by single-cell multimodal omics, where e.g. non-destructive assays allow simultaneous genome sequencing and transcriptome profiling of the same single cells (Zhu et al. 2020). The costs of single cell experiments decrease continuously which makes such studies prevalent today (Linnarsson et al. 2016).

The availability of 'Omics' technologies inspired the foundation of various large consortia that tried to systematically apply them on a big scale. Amongst them is ENCODE (ENCODE Project Consortium 2012), which started in 2003, and is the successor of the already mentioned Human Genome Projects. Motivated by the observation that protein coding genes only account for approximately 1.5% of DNA in the human genome, this ongoing international collaboration set out to explore the role of the remaining DNA and to compile a comprehensive list of its functional elements. By applying various assays to study transcription (RNA-seq), DNA binding (ChiP-seq), DNA accessibility (ATAC-seq), and others, the goal is to determine which DNA elements act at the protein and RNA levels, control cells and circumstances in which a gene is active. This lead to an assignment of biochemical functions for 80% of the genome, a claim that provoked substantial criticism (Doolittle 2013; Eddy 2013).

Another major (public) data collection effort that was launched after 'Omics' technologies became available is The Human Protein Atlas (Uhlén et al. 2015). This ongoing project started in 2003 and applies antibody-based imaging, mass spectrometry-based proteomics, and transcriptomics to map human proteins in cells, tissues and organs.

Furthermore, The Cancer Genome Atlas, running from 2006 to 2018, compiled genetic mutations (amongst others) in 33 cancer types from 11000 tissue samples (Ding et al. 2018). Some of the key findings from this massive data set were that cancers could be grouped by (pan-tissue) molecular characteristics rather than their tissue of origin (Hoadley et al. 2018), and that cancer is a signalling disease with 89% of tumours having at least one significant alteration in 10 key signalling pathways (Sanchez-Vega et al. 2018).

Another noteworthy mention is the more recent launch of the Human Cell Atlas Project in 2016. Here, the focus lies on using high-throughput single-cell molecular profiling to map all cell types in the human body (Regev et al. 2017). The aim of the first project phase is to profile 30-100 million cells from major tissues of healthy individuals.

20 years of systems biology

'Omics' technologies provide the data that enables a previously inaccessible description of biological networks from a top-down approach, using network inference methods. But network inference is only a part of a much larger, 'Omics'-triggered reformation of quantitative biology. To understand how network reconstruction fits into this new scientific context, this section reviews some of its developments.

The 'Omics' revolution established of an era of unprecedented abundance of biological data. Yet, it turned out that identification and quantification of cellular components alone did not provide meaningful insights into the workings of a cell. What was needed to extract knowledge from ever more powerful experimental techniques was a radical shift in research methodology. This transformation became known as systems biology. The field quickly manifested by the founding of the Institute for Systems Biology in Seattle, Washington and the Systems Biology Institute in Tokyo both in 2000. The same year, the International Conference on Systems Biology (ICSB) was launched. Shortly thereafter, perspectives on the matter were written (Ideker et al. 2001; Kitano 2002) and dedicated journals appeared (Molecular Systems Biology and PLOS Computational Biology in 2005, BMC Systems Biology in 2007). The field expanded rapidly. Within a few years, the number of new articles per year that are indexed in PubMed with a "systems biology" label grew into the thousands (Chuang et al. 2010), and the ICSB that started as a meeting with little more than 20 speakers in 2000 became a full-fledged conference with many parallel sessions, featuring more than 300 talks and workshops in 2019.

But even though the term systems biology is omnipresent today, its specific meaning is still under debate (Green 2016). The apparent consensus may be that systems biology addresses phenomena that are in some sense complex and thus rely on a mathematical or computational formalism for their analysis. Yuri Lazebnik illustrates this idea in his light-hearted contemplation on whether a biologist can fix a radio (Lazebnik 2002). Furthermore, systems biology pursues a wholistic approach to study emergent properties that arise from the complex interplay of the system's components. Philip Warren Anderson, who sadly passed away while I was writing this chapter, coherently explained this concept in his seminal paper "More Is Different" (Anderson 1972). There he argued that a strictly reductionist point of view, which attempts to explain an observed phenomena by breaking it down into ever smaller entities, would fail to constructively describe a complex system. Rather an effective description is based on a scientific hierarchy, where at each level emergent properties are derived from basic principles which in turn serve as the basic principles in a next higher-level description of the system. In practice, these concepts have become widely internalized in the molecular biology community, and often quite subliminally so. It could be regarded as an outcome of 20 years of systems biology that we can sensibly work with high-level concepts, such as protein function or information flow, without needing to detail their biochemical underpinnings. But to not delve into a philosophical discourse about the nature of systems biology, I will resort to a mostly ostensive description.

Research in systems biology is not confined to any particular species, biological mechanism, nor any length or time scale. To give an impression of this diversity and of the systems biology approach, I want to point out a few of the more prominent works of the field. One that attracted a great amount of attention was the discovery of network motifs (Milo et al. 2002; Shen-Orr et al. 2002; Milo et al. 2004; Alon 2007). These motifs are small subnetworks (with typically three to five nodes) that occur in a given network significantly more often than in randomly rewired control networks. This overabundance is thought to manifest evolutionary design principles in biological or synthetic systems. Prominent examples are the feed-forward loop and bi-fan structure that was found to be characteristic for gene regulatory networks. Hopes are that motifs allow to characterize entire network superfamilies by more simpler principles. Yet, the excitement was not unanimously shared. It was argued that motifs simply occur due to specific constraints of the underlying network, such as local clustering effects (which is not taken into account by the background model) (Artzy-Randrup et al. 2004). Furthermore, it was shown that a network motif can exhibit opposing behaviour depending on its parametrization. This suggests that it might be impossible to conclude about biological function of the motifs (Ingram et al. 2006).

Networks generally play a central role in systems biology. They were used to chart genetic links between human diseases and genes (Goh et al. 2007). In yeast, large-scale proteomics studies revealed a protein interaction network (Krogan et al. 2006). Similarly, a set of yeast double mutants including up to 6000 genes allowed to map a genetic interaction network based on synthetic lethality with nearly 1 million interactions (Tong et al. 2004; Costanzo et al. 2010; Costanzo et al. 2016). Such type of networks then form the basis of computational models of either cellular processes, for example metabolism in E. Coli (Orth et al. 2011), or even whole-cell models for simple organisms such as bacterium Mycoplasma genitalium (Karr et al. 2012). Such cell models had already been proposed by Francis Crick in 1973 (Crick 1973). As experimental and computational technology advanced, they seemed within reach and were deemed “the ultimate goal” (Carrera et al. 2015) of systems biology and a “grand challenge of the 21st century” (Tomita 2001), because they would allow to predict complex phenotypes and perturbation responses, as well as to optimize the design of future experiments. However, they also face profound criticism. Some argue that whole cell models will always remain utterly incomplete in light of the astronomical number of cellular interactions (Noble 2012). Furthermore, their inherent lack of abstraction or simplification might fail to give them any explanatory power (Krohs et al. 2007). Such questions are extensively discussed for the Mycoplasma genitalium model in (Gross 2017).

In contrast to such computational efforts, systems biology also includes more theoretical works, such as a study on phenotype switching in clonal populations (Kussell et al. 2005). Here, the analysis of a simple growth equation showed that a phenotypical adaptation to a changing environment that is triggered by stochastic phenotype-switching mechanisms can be favourable to one relying on active sensing. Another investigation on the optimal design of the signalling network of bacterial chemotaxis (Kollmann et al. 2005) revealed that the naturally evolved pathway structure is more robust to gene expression noise than multiple alternative topologies. And as a final example, it was suggested that many biological systems might exhibit self-organized criticality (Mora et al. 2011). This means that certain characteristic properties of the system, for example the activity of retinal ganglion cells, obeys a power law, which could yield an optimal capacity for stimulus representation, or information storage and transmission (Shew et al. 2013). However, others claim that such criticality might solely arise due to external fluctuations rather than from a self-organized fine-tuning of parameters (Schwab et al. 2014).

With time, the field of systems biology further diversified and systemic approaches also entered other areas of research. Today, there is Systems Medicine (Auffray et al. 2009; Apweiler et al. 2018), Systems Genetics (Civelek et al. 2014), or Systems Immunology (Davis et al. 2017) to name but a few of the biological "systems sciences". In fact, the term is now so abundant that it no longer describes a distinct and novel way of doing science. 20 years of systems biology showed that genes do not act in isolation but are embedded in a multi-layered and modular regulatory system (Noble 2008; Hartwell et al. 1999; Kashtan et al. 2005). It also made clear that the elucidation of this astounding complexity requires an integration of experimental, computational and mathematical efforts (Aderem 2005). The success of having established these ideas might complete the epoch of systems biology.

2.2 REVERSE ENGINEERING OF BIOLOGICAL NETWORKS

The previous section showed that a central idea of systems biology is to derive effective descriptions of cellular complexity in terms of networks of interacting modules. These descriptions rely on 'omics' technologies to observe the entirety of involved cellular components.

Yet, the 'Omics' approach per se only offers "complete but physiologically uninterpreted data sets" (Krohs et al. 2007). Some even claim that the ever increasing precision in the identification and quantification of cell components can hamper an effective understanding of cellular processes (Stern 2019), unless the refinements in the experimental protocols are met with advances in mathematical approaches to data analysis (Bizzarri et al. 2019). Currently, the development of such adequate analysis methodology seems to be lagging behind. Alluding to Lazebnik's reflections on how a biologist would fix a radio (Lazebnik 2002) (mentioned in the last section), Jonas and Kording asked whether a neuroscientist could understand a microprocessor (Jonas et al. 2017). The microprocessor was chosen because it can be seen as a hugely simplified model system of a brain, for which data in any level of detail can be obtained. At the same time, being a man-made device it provides a known truth against which analysis results can be tested. Yet, the sobering realization was that even when detailed datasets are available, current methods from neuroscience fail to capture the inner logic of the device and cannot derive functional insights. I would argue that this situation applies to other branches of biology as well. Take as example that even in well-studied eukaryotic model organism such as budding yeast, 20% of the proteins lack any informative description of their biological role (Wood et al. 2019). This is without saying that a simple protein assignment to a biological process for the other 80% does not come anywhere near to allow understanding a protein's function in a given physiological context. The list of fundamental problems that are unsolved despite an abundance of available data is long (Dev 2015). This explains the ongoing focus of the systems biology community to develop new mathematical and computational methods (Polychronidou et al. 2017).

Amongst them is a class of algorithms to reverse engineer biological networks from 'Omics' data. Its goal is to identify or even quantify (pairwise) interactions between the components of a biological system from experimental observations of the system's state. The hope is that this elucidates the underlying processes and mechanisms, and thereby ultimately provides a functional understanding of the system.

In some settings, network information arises directly from experimental data. For example, there is an impressive array of different experimental techniques to analysis protein-protein interactions (Titeca et al. 2019). Similarly, the previously mentioned double-mutant epistasis screens in yeast (Costanzo et al. 2010; Costanzo et al. 2016) quantify genetic interaction strengths directly from measurements of cell viability or culture size. As it was observed that ~73% of Yeast gene are non-essential, the idea is that synthetic lethal genes can indicate processes that buffer each other and thereby display functional relationship (Tong et al. 2004). Also various other measures of genetic interactions from phenotype measurements were defined (Drees et al. 2005).

However, these are exceptions and usually there are no experimental protocols to directly obtain a useful network description. This can have technical reasons, e.g. that double mutant libraries are not available, but more importantly, this is due to a more conceptual limitation. Experimental approaches to characterize network interactions often provide biochemical information that fails to capture the nature of the interactions that the network is supposed to describe. For example, ChIP assays can indicate that a certain transcription factor binds to a promoter of a gene. But this information does not adequately describe gene regulation because a binding event might not be functional or not sufficient with respect to transcription initiation. An alternative strategy is therefore to rather identify appropriate readouts for the relevant components (e.g. mRNA counts), and to derive effective networks from experimental observations of their behaviour under various conditions, in response to perturbations, or over time. In this way, the inferred network becomes an interpretable model for the observed biological phenomenon in a defined biological context. 'Omics' technologies deliver the type of data that is suitable for this approach. Thereby, they initiated a substantial amount of research on the reverse engineering of biological networks.

Some of the first highly cited network inference methods were described around the turn of the millennium (Liang et al. 1998; Friedman et al. 2000; Yeung et al. 2002; Ideker et al. 2002; Friedman 2004) and followed by a continuous stream of new publications, whose number was found to double every two years (Stolovitzky et al. 2009). Today, the literature about the reverse engineering of biological networks includes thousands (Jurman et al. 2019) of research articles about new methods and applications, various books (Lingeman et al. 2012; de la Fuente 2014; Haibe-Kains et al. 2015; Sanguinetti et al. 2019), and more than forty review articles (Natale et al. 2017). Amongst the latter, Markowetz and Spang 2007 provides an excellent overview of the different mathematical and algorithmic approaches, and Natale et al. 2017 presents a comprehensive discussion about applications. The aim of the following sections is thus to only give an overview of some basic concepts and typical approaches, which then allows to evaluate the current state of the field and identify open challenges that inspired the works of this thesis.

Methods

A network is defined as a set of nodes and edges connecting some node pairs. Edges can be directed or undirected. Furthermore, they can be weighted (they are associated with some scalar value) or unweighted.

In the latter case, they merely describe the topology of the network. If on the other hand, there is a weight associated to an edge, it mostly describes one of two properties. Either, it represents the confidence that the data supports the existence of the edge, or it quantifies a certain strength of interaction between the two connected nodes.

Next, we can distinguish observational, time-course and interventional data. Measuring time-courses or observing the system's response to targeted perturbations is often challenging for biomolecular systems. Therefore, observational data, consisting of samples of the unperturbed system state in various conditions, is much more abundant and the majority of inference methods is designed to interpret this kind of data. One of their major strategies is to characterize network edges based on measures of association between pairs of network nodes, such as (Pearson) correlation, for example used in WGCNA (Langfelder et al. 2008). Yet such an approach will not properly distinguish between associations from direct interactions and those that arise due to secondary interactions. To also account for such indirect node interactions, others applied partial correlation scores (Schäfer et al. 2005). Still, correlation coefficients indicate the strength of linear relationships, which rarely arise in a biological system. To address this issue, others applied information theoretic scores such as mutual information, which was for example implemented in ARACNe (Basso et al. 2005). However, the estimation of mutual information from finite datasets is non-trivial (Steuer et al. 2002). To account for non-linearities while also avoiding finite sample effects, more recent work (Ghanbari et al. 2019) has therefore suggested to apply distance correlation (Székely et al. 2007) as a measure of association. Alternatively, interactions can be quantified by a maximum entropy approach (Stein et al. 2015; De Martino et al. 2018). This approach fits parameters of a probability distribution such that the distribution captures the means and covariances of the data while maximizing its entropy to avoid any unjustified bias. The parameters can be interpreted as pair-wise association measures between network nodes. An important feature of this model is that it can be fit to discrete data (it then becomes a Potts model), for example to DNA sequence alignments to reveal evolutionary couplings between pairs of nucleotides (Weigt et al. 2009; Marks et al. 2011).

Ultimately, to obtain a (non fully-connected) topology, each of these methods needs to define a threshold to distinguish important from negligible association scores. All the mentioned association measures are symmetric and thus the inferred networks are undirected. They can therefore not be interpreted in a causal sense. Rather edges are thought to combine functionally associated nodes. These could for example represent co-expression gene modules, or, as often claimed, even gene regulatory networks, although such an interpretation is questionable as discussed further below.

A different type of strategy is to infer Bayesian networks from observational data. A Bayesian network is a probabilistic model with directed edges that represent the conditional dependence of its variables on each other. It implies a joint probability distribution with the interpretation that the data was sampled from it. Using Bayesian inference (Needham et al. 2006), its parameters can be tuned as to maximize its likelihood to the data. However, such maximization relies on the specification of a specific network structure. While established algorithms exist (Spirtes et al. 2000), the search for network structures with a maximum likelihood in the super-exponential space of possible networks is challenging and remains an active area of research (Ghanbari et al. 2015). Bayesian network inference methods (Friedman et al. 2000; Friedman et al. 2003) have been particularly popular because they provide a natural way to integrate prior network knowledge. Another particularly astonishing feature is that these methods infer directed networks from purely observational data. However, the directions of the edges are not necessarily meaningful (Verma et al. 1990). That is because different Bayesian network structures can give rise to identical joint distributions, a phenomenon known as Markov equivalence. Thus, a Bayesian network does not generally provide a unique causal structure. However, Bayesian model averaging can help with causal discovery (Koller et al. 2009).

A Bayesian network cannot contain cycles (it is a directed acyclic graph, DAG). This is a considerable limitation as feedbacks are pervasive in biology (Thomas et al. 1990). When time-course data is available, dynamic Bayesian networks (Dean et al. 1989) can provide a solution to this problem. In these probabilistic models, a variable represents a specific biological entity at a given time point. In this way, a path in a DAG can lead back to the same entity at a later time point and thereby model feedback loops (Hill et al. 2012).

Another popular approach to derive causal models from time-course data is to describe the observed dynamics as a system of differential equations. Typically, it is assumed that trajectories remain close to steady state, which justifies a linearisation of the system. This then allows to fit model parameters, which can be interpreted as interaction strengths. Often, this leads to an underdetermined optimization problem which requires regularization strategies. A popular example of such a method is the Inferelator (Bonneau et al. 2006).

Other approaches on time-course data do not rely on an explicit dynamic model. This includes methods that are based on convergent cross mapping (Sugihara et al. 2012), Granger causality (Zou et al. 2009), or transfer entropy (Runge et al. 2012).

Many of these methods can more reliably determine directed interactions, when the observed time-courses track the response to an intervention at the system (Pearl 2009). However, many experimental techniques are destructive so that it is often impossible or prohibitively laborious to perform measurements at various time points. It is thus more common to only observe system once per perturbation, typically after transient adaptations have terminated and the system has relaxed into a new equilibrium state. The methods developed in this thesis work within this setting.

Perturbations can either target a single or a few network nodes, as is the case for knockouts, gene overexpression, inhibitors, and ligands, or they can be multi-factorial (Jansen 2003), e.g. when comparing populations with different mutational backgrounds. Some of the previously mentioned approaches that infer networks from observational data can be adapted to include interventional data, see (Markowetz and Spang 2007). Notably, Bayesian networks can introduce what is known as ideal interventions (Pe'er et al. 2001; Pearl 2009). These interventions account for perturbations by collapsing according probability distributions to a point mass. Then, some Markov equivalent network structures might no longer provide the same maximum likelihood and thus become distinguishable. In this way perturbations help to determine the direction of causality.

Alternatively, if each single network node can be experimentally perturbed, a much more direct way to infer a directed network is to simply draw edges from a perturbed node to all nodes which showed a significant response to the according perturbation. Clearly, the resulting disruption network (Rung et al. 2002) fails to distinguish between direct and indirect effects. To overcome this challenge, one could hypothesise that the direct interactions are captured by the transitive reduction of the disruption network (Wagner 2001). The transitive reduction (Aho et al. 1972) is the directed network with the fewest edges that still contains a path from a perturbed node to all nodes that showed a response. Yet, also this approach requires a fully perturbed network, is sensitive to noise, and does not allow for an integration of prior network knowledge. More importantly though, there is no fundamental reason to believe that the transitive reduction is biologically justified (de la Fuente, Brazhnik, et al. 2002). In fact, these and other shortcomings were part of the motivation for the development of the response logic approach (Gross et al. 2019) that is introduced in the next chapter. There, I will further discuss similar methods and the more general idea of inferring boolean network models.

In any case, these methods solely consider an edge to be present or absent. Yet, such purely topological information might not always be enough to determine the system's behaviour, as shown for examples in simple synthetic gene networks (Guet et al. 2002), or in the applications described further below. In these scenarios it is crucial to quantify interaction strengths, that is, to infer weighted directed networks. While this can be achieved with Bayesian networks, the involved estimation of probability distributions may require many data samples per perturbation. In contrast, another approach that can infer weighted directed networks from smaller data sets relies on theoretical work in Metabolic Control Analysis (Kacser et al. 1973; Heinrich et al. 1974) and was termed Modular Response Analysis (MRA) (Bruggeman, Westerhoff, et al. 2002). Here, the idea is, again, that the underlying system can be represented by a system of ordinary differential equations. The measurements of the perturbed system are considered as its perturbed steady states and allow to compute the unknown entries of the system's Jacobian matrix, which can be interpreted as interaction strengths. Many extensions to this original idea have been developed (Santra et al. 2018) and will be reviewed in Chapter 4. This thesis also makes contributions to the field of MRA by studying network identifiability and experimental design, as discussed in Chapter 4, and ways to improve optimization of MRA models, which is described in Chapter 5.

This concludes the overview on the most common approaches to network inference. Many more aspects have been discussed in the literature. Amongst them are hidden (unobservable) nodes, which are for example addressed by nested effects models (Markowetz, Kostka, et al. 2007), learning from heterogeneous data sources (Hecker et al. 2009; Chiquet et al. 2019), temporarily evolving networks (Parikh et al. 2011), the recent surge of methods to handle single cell data sets (Aibar et al. 2017; Matsumoto et al. 2017; Todorov et al. 2019; Wang et al. 2019), and many more. Additional information about these developments can be found in more specialised reviews (Sanguinetti et al. 2019).

Evaluation

How can we judge if a network inference method generates useful results? There are several aspects to this question. One approach is to evaluate the performance of a network inference method by the accuracy of its prediction. This would require some gold standard networks to compare against. I can see three reasons why these are hard to come by. For one, they would have to be generated by some complementary experimental method which is often unavailable and also prone to be incomplete and erroneous. Furthermore, it is challenging to ensure a meaningful overlap between a putative gold standard network and the data that serves as input for the inference. For example, it is problematic to benchmark transcription factor networks inferred from transcriptome data on transcription factor binding sites obtained from a ChIP assay. The complication is that binding of a transcription factor at the promoter region of another transcription factor is an indication but no proof of regulation. So even in the absence of experimental error, the networks are not fully comparable. A third reason why a comparison to a gold standard network can be problematic is that every method is based on different assumptions and comes with a different understanding of how the inferred network is to be interpreted. Some methods might be tailored towards a detailed deciphering of the system's biochemical underpinnings, while others attempt to trace the flow of information. How can one then define a single gold standard network to compare these different network concepts?

At the expense of biological relevance, these difficulties can be circumvented to some extent by an evaluation on synthetic, simulated networks. This was the strategy in the first public network inference challenges conducted by the Dialogue on Reverse-Engineering Assessment and Methods (DREAM) project (Stolovitzky et al. 2007). DREAM challenges have since become the de facto standard benchmark of reverse-engineering methods. Participating teams make predictions on provided data sets that are then scored against a held-back gold standard in order to rank the applied methods. Yet, the first years of the challenge led to a perplexing outcome. In retrospect of DREAM3, the organizers conclude that

> However, for the majority of inference methods the precision of the predictions was rather low .... In addition, a surprisingly large number of methods (11 out of the 29) produced network predictions that were, on average, not significantly better than random guessing .... This is a sobering result for the efficacy of the network-inference community. (Marbach et al. 2010)

This underperformance can have many reasons, for example that some participants simply were not very experienced in the development of such methods. Likewise, it might not necessarily reflect upon a poor state of the field for the reasons discussed above. However, there also is a general technical problem that could explain poor results, which might not always have been fully appreciated. It is the issue of identifiability.

Network inference tends to act within a massively underdetermined setting, as typically the number of nodes far exceeds that of the (independent) samples (Natale et al. 2017; Bonneau et al. 2006). In this case, the data is insufficient to confidently infer all edges of the underlying network (Szederkényi et al. 2011). In practice, many methods approach this problem by regularization in one form or the other. Inferring a transitive reduction (Wagner 2001), as discussed above, is one approach that picks a particular network (the sparsest) within a set of equivalent alternatives. Another example is the ARACNe (Basso et al. 2005), which applies the data processing inequality to develop a heuristic that strives to remove putative indirect interactions. Similarly, L1 regularization is used in the optimization of the Inferelator (Bonneau et al. 2006) in order to prevent overfitting. Whenever the goal is to construct performative classifiers or even predictive models, such types of regularization can be an essential tool. However, they require caution when the goal is to interpret the regularized network model itself. The reason is simply that a maximization of sparsity that is implied by regularization might not correctly represent biological networks. Admittedly, ecological, gene-regulatory, metabolic and other biological interaction networks have been reported to be sparse (Busiello et al. 2017), which was, for example, explained by evolutionary selection towards robustness to network perturbations (Leclerc 2008). However, the topological space of sparse network structures that can equally well explain a certain data set can potentially be highly diverse. Committing to a single network amongst them would then likely yield a poor prediction. Additionally, a maximization of sparsity might miss the biological truth, as functional constraints can necessitate additional network edges but might not be reflected by the considered data. For example, recall from the previous section that the most abundant gene regulatory network motif is thought to be a feedforward loop (Milo et al. 2002). Its suspected function is to activate the output node only under a prolonged stimulation and to quickly deactivate it when the stimulation stops (Shen-Orr et al. 2002). Yet from the perspective of a transient reduction, the additional feedforward edge in the circuit is dispensable and would be removed.

To avoid such misjudgements, a reconstruction method should not uncritically commit to a single solution network but rather be explicit about which parts of the network are uniquely determinable by the data, and where rivalling network hypothesis exist (Altay et al. 2010). While such an identifiability analysis is a standard procedure in biological modelling (Chis et al. 2011; Bellman et al. 1970), it has been largely underappreciated within the realm of network reconstruction. Notable exceptions include methods that take into account the uncertainty about inferred parameters by inferring entire parameter distributions instead of just single values (Molinelli et al. 2013), or those that not only infer a single model but are rather explicit about non-identifiabilities by considering an entire ensemble of networks (Kuepfer et al. 2007) which describes the data equally well (Szederkényi et al. 2011; Ud-Dean et al. 2014).

Gaining a rigorous understanding about identifiability of network parameters is crucial for a meaningful network inference but has not yet been sufficiently addressed. It is thus a central focus in the methods that were developed in this thesis. The response logic approach (Gross et al. 2019) presented in the next chapter, returns a comprehensive ensemble of data-conforming networks. And our study on identifiability within the framework of MRA, presented in Chapter 4, maps the question of parameter identifiability onto an intuitive maximum-flow problem, which additionally allows to optimize the experimental design to maximally reduce non-identifiability.

Beyond these more technical aspects of network inference, it is worth thinking about how exactly such methods can aid to understand biology. This seems especially important as the development of new algorithms often seems technically motivated and as the field has become very self-referential. Likewise, it is not convincing to only report a method's superior DREAM scores, when it remains unclear whether it can reveal useful insights from novel data sets. Such shortcomings lead to the unfortunate situation that many new methods will not find applications beyond the decorative examples provided in their publication. To that effect, there should be an increased effort to identify the biological system, data set or use case that a novel method can be applied to. Obviously, it is impossible to define general, quantitative criteria that allow to judge this applicability. Instead, one can point out a number of examples for which reverse engineering expanded our understanding of biology, as done next.

Applications

Network inference methods are applied in various different areas. They serve to derive gene regulatory models from transcriptomics data, but they were also used to describe metabolic pathways, neural circuits, signal transduction, interaction of species, disease networks and many other types of interaction. The following paragraphs therefore do not aim for a comprehensive description of the applications of biological network reverse-engineering, as attempted in Natale et al. 2017, but rather discuss a variety of different examples that represent qualitatively different ideas in which inferred networks can be useful.

Many reverse-engineering methods were developed to be applied on large-scale 'omics', in particular transcriptomics data, in order to infer gene regulatory networks, specifically transcription factor interactions. Examples include the elucidation of MYC targets in human B cells (Basso et al. 2005), or a large-scale study in in Escherichia coli (Faith et al. 2007), which lead to the discovery of many novel transcription factor interactions and a regulatory interaction involved in metabolic control of iron transport, which could be confirmed by follow-up experiments. These studies computed association scores from transcriptional profiles under various different conditions, that is, from observational data. But as discussed before, this only allows to infer undirected networks, which cannot distinguish whether a certain transcription factor is upstream or downstream of its network neighbours. Yet, the lack of such network logics hampers the design of predictive models. In addition, a downside of association score based network predictions is that they possibly tend to overestimate the number of network links. That is because, quite commonly, the variation of transcriptional profiles under various conditions is not explainable from within the network itself. For example, it is imaginable that in certain conditions cells grow in environments, which contain a receptor ligand that drives the transcription of a set of genes that exert no regulatory control amongst each other. Therefore, although they are disconnected, genes in this set will vary concertedly across conditions and appear as highly associated. Thus, even methods that take great care in rejecting indirect links, will erroneously suggest at least some links between them. This is the classic problem that correlation does not imply causation. It thus requires caution to interpret association scores as indicators for regulatory interaction. Often, it might be more adequate to view sets of connected nodes as functionally associated modules. While this point of view diminishes the potential to understand cell regulation and derive predictive models, it can still be useful, for example to predict protein function based on annotated function of neighbouring proteins (Sharan et al. 2007).

Beyond the scope of molecular cell biology, the idea to derive undirected networks from activity or abundance profiles found applications in an astounding diversity of topics. Such networks were used to describe species interactions on the microscopic (Menon et al. 2018) as well as the macroscopic scale (Volkov et al. 2009), they allowed to describe organization of coding in neural populations (De Martino et al. 2018), they represent co-evolved protein residues that allow to predict protein contacts and ultimately protein folding (Weigt et al. 2009; Marks et al. 2011), and were used in many more contexts (De Martino et al. 2018; Stein et al. 2015).

In contrast, to be able to make causal statements and to derive directed networks, many studies relied on perturbation data. In particular, many perturbation experiments have been carried out to decipher various signalling pathways, their cross-talk and context-specificity, with a focus on cancer. Dysregulation of signalling is a driving mechanism in a majority of cancer types (Sever et al. 2015) and a detailed understanding of these functional changes forms the basis of targeted therapy (Gerber 2008). In this context, an important contribution of perturbation studies is to reveal the special role of feedback loops in signalling. For example, Klinger et al. 2013 measured changes in phosphorylation of kinases within the epidermal growth factor receptor (EGFR) signalling network in colorectal cancer cell lines upon perturbations by small molecule inhibitors and growth factors. They could then apply MRA to quantify a signalling model which unveiled a negative feedback from kinase ERK to EGFR. The model could thus mechanistically explain how an inhibition of MEK, the upstream kinase of ERK, increases the activity of EGFR and thereby also that of its other downstream target, AKT. As the AKT pathway is associated with cell survival, this finding undermines the benefit of MEK inhibition for eradication of colorectal cancer cells. Instead, it lead to the hypothesis that cell growth is effectively blocked by a combined inhibition of EGFR and MEK, which could be confirmed in a xenograft model.

While this study constructed a pure signalling model, others explicitly included variables to describe phenotypes. Perturbation experiments on a BRAF-mutant melanoma cancer cell line (Molinelli et al. 2013) lead to a model that not only identified novel signalling interactions but can also systematically simulate the effect of inhibitions of different kinases, beyond the ones observed experimentally. This allowed for a prediction of efficacious drug targets, such as PLK1 whose pivotal role in cell viability could be confirmed by follow-up experiments.

But network inference methods can also address the inverse question of determining the direct targets of a compound. The difficulty is that due to secondary effects, perturbations will affect not only their direct targets but all of their downstream nodes. To this end, perturbation experiments on a nine-gene subnetwork of the SOS pathway in Escherichia coli (Gardner et al. 2003) enabled a network inference method to distinguish these first and second order effects. Thus, network reverse-engineering can also identify the mode of action of uncharacterised pharmacological compounds.

Yet another type of network inference applications is to use the inferred networks as input to downstream analyse. For example, a toxicity study on human embryonic stem cells (Yamane et al. 2016) reconstructed a Bayesian network from qRT-PCR data on ten genes at four time points after administration of five different doses for each of 22 chemicals. The resulting weighted network edges were than added as features to a support vector machine. This improved its accuracy as a classifier of toxicity categories.

Outlook

A recent benchmark compared seven popular network inference methods on a large body of transcriptome data from mouse embryonic stem cells (Meisig et al. 2018). The disheartening result was that the reconstructed networks varied widely and that their features did not reflect upon the data but could rather be traced back to the design of the algorithm. Apparently, also twenty years after it became a trending research topic, reverse engineering of biological networks remains an unsolved problem. So what can be done about that?

A significant hurdle on our way to a better understanding of biological networks is the seemingly prevalent idea that there is a single, true network that underlies the data, and that this network can be reconstructed if only our mathematical and computational abilities become sufficiently developed. This view fails to appreciate that a network is just a mathematical concept to simplify biological complexity at a certain level of abstraction. The spectrum of what networks can represent ranges from specific biochemical reactions to effective descriptions of information flow without specific mechanistic underpinning (Gardner et al. 2005). It is crucial to make a deliberate choice about this level of granularity to be able to interpret an inferred network and generate biological hypotheses from it. Merely listing a "hairball" of decontextualized interactions (Lander 2010) is not inherently more fruitful than to report the original data (Natale et al. 2017). Yet, making such an informed choice relies on a good understanding of what the data represents and what it can allow to learn. The importance of biological context seems to be underappreciated by the reverse-engineering community. An algorithm cannot blindly create knowledge and the automation of science (King et al. 2009) is still a dream. Fortunately, there are methods that are embedded in biological context and incorporate various forms of biological knowledge (Siahpirani et al. 2019). One particularly interesting idea in this direction is that of network contextualization (Liu et al. 2019). Here, instead of reconstructing a network de novo, the algorithm identifies from an ensemble of known pathways those that best explain a set of expression profiles.

As biological systems can be viewed from different perspectives, there generally is no single, best inference method for a certain data set. The challenge is that, in the absence of solid gold standards (see above), methods cannot easily be evaluated. Rather an algorithm has to be chosen based on an assertion of how well its inherent assumptions conform to the biological context. However this is notoriously difficult, as these assumptions are typically expressed in mathematical or algorithmic terms that cannot be directly compared to biological knowledge. Who can say, whether the data processing inequality applied in the ARACNe (Basso et al. 2005) is more adequate to eliminate indirect links in an inferred transcription network in mouse embryonic stem cells than removing them by partial correlations? This kind of problem was also noted in the DREAM challenges, in so far as "sophisticated methods that would in theory be expected to perform better . . . , were more strongly affected by inaccurate prior assumptions in practice" (Marbach et al. 2010). This suggests that the reverse engineering community should increase their efforts to clarify the context in which their methods work. This can be achieved by tailoring methods towards specific applications, e.g. inferring signalling networks from phosphoproteomic data (Invergo et al. 2018), or by simplifying a method's assumptions, which was a central motivation in the development of the response logic approach, which will be presented in the next chapter.

3 ROBUST NETWORK INFERENCE USING RESPONSE LOGIC

This chapter introduces a method called response logic approach, which infers directed networks from perturbation data to derive mechanistic insights. Yet as discussed in the preceding chapter, these networks are not necessarily the only valid representation of a system's interconnections. Rather, the derived interactions reflect the biological assumptions and the viewpoint of the reconstruction method. Therefore, different methods will infer different networks that can potentially be useful to answer different questions about the biological system. The problem is that many methods offer no clear interpretation of their results because their inherent assumptions are either not explicitly formulated or have no tangible biological meaning. Thus, it is often difficult to assess whether a result is adequate for the data set at hand. This challenge was the main motivation in the development of the response logic approach. It was designed to deliver easily interpretable networks by being based on very simple assumptions that apply to a wide array of biological contexts.

An established way to summarize biological complexity into understandable patterns of interactions are logical models (also called boolean networks). These models are useful if the components of the biological system can naturally be described by a discrete number of states (typically two). These states can represent, for example, whether a gene is expressed or not, a kinase is active or inactive, or a metabolite is present or absent. Nodes in boolean networks represent boolean functions that indicate if a specific component changes its state depending on the state of its input nodes. Such state transition rules can be applied synchronously (simultaneously over all components) or asynchronously (sequentially over components) to define a step-wise trajectory that is thought to represent the development of the modelled system over time.

Due to the involved binarization, logic models are a crude simplification of biological complexity. But often they constitutes the adequate level of abstraction to capture the main features of the system. This is why boolean networks have a long tradition in quantitative biology and were used in a large variety of applications (Abou-Jaoudé et al. 2016; Morris et al. 2010). Accordingly, there is large body of methods to reverse-engineer boolean networks (Hickman et al. 2009). Generally, this involves both, the inference of the topology and the assignment of boolean functions to the node. Even for small networks such a task constitutes a combinatorial search space of stupendous complexity (Tatsuya 2018). Therefore, most methods require an input of comprehensive time-course data (Berestovsky et al. 2013) and rely on heuristics such as limited indegrees (Liang et al. 1998). However, many systems cannot be experimentally observed over the course of time. Rather, the available data might only describe the altered state of the system some time after the application of one of various perturbations. This makes for an even harder and more underdetermined computational challenge so that methods that rely on such perturbation data typically incorporate a prior knowledge network. To account for context, this input network can then get optimized such that its perturbed steady states maximally conform to the experimentally observed responses (Saez-Rodriguez et al. 2009). But also this approach depends on ad hoc regularization strategies to reduce model complexity.

This shows that even though boolean networks provide a simple description of functional relationships between system components, their inference is challenging. To avoid this obstacle, the response logic approach is based on an even simpler idea. In contrast to the complexity of a logic model, the response logic simply states that

> a perturbation at a node is propagated along all outgoing edges to the set of connected nodes, and these responding nodes will in turn propagate the signal and so forth. Consequently, a perturbation of a node causes a response at all nodes to which it is connected by a path, and no response at all others. (Gross et al. 2019)

Thus in contrast to boolean networks, the response logic no longer defines any dynamic trajectories but only steady states, and all nodes represent OR gates. Despite of its simplicity, this model represent a common way to reason about cause and effect in biological networks. At the same time, it renders the inference computationally tractable.

The aim of the method is to infer all topologies that best describe the experimental perturbation data given the response logic. Without having to define any parameters or heuristics, this difficult combinatorial search problem can be solved with a powerful logic programming approach, called Answer Set Programming (ASP) (Gebser et al. 2014). ASP has been applied to answer a wide range of questions in bioinformatics (Dal Palù et al. 2018), and led to the development of various data analysis tools (Gebser et al. 2010). In particular, it has been used to infer Petri nets (Ostrowski et al. 2011) and boolean networks (Guziolowski et al. 2013; Videla et al. 2015). But once again, these approaches aim to identify the boolean gates that act on the nodes of a prior knowledge network. In contrast, when such prior knowledge is not available or incomplete, the response logic approach can be applied to reconstruct it, making it a complementary strategy. Clearly, to crude OR-logic of the inferred network might not always be an adequate description. Yet, our publication demonstrates the validity of the response logic in a variety of settings and shows, for example, how it can generate mechanistic hypotheses about signalling pathways that explain drug sensitivities of a colon cancer cell line.

In comparison to existing network reconstruction methods, the response logic approach has a few distinct features. An inferred network is intuitively interpretable: an edge indicates that its target node will show a response to a perturbation if the source node is perturbed. This also entails a very obvious relationship between data and inferred networks. We can easily understand why a certain data point either rules out or necessitates the existence of a certain edge. And this also has the crucial advantage that it becomes apparent when the response logic is not a good model for the biological system. Moreover, the response logic approach is unbiased in the sense that it will return all networks that conform to the given data. It might seem inconvenient, having to deal with such an ensemble of solutions. But this display of non-identifiability can convey the valuable message of where the connectivity of the system can be disentangled and where not, given the current data set. Furthermore, the logic programming framework makes it possible to incorporate additional information and network constraints in a very natural way, which is demonstrated in the publication.

Previous work has discussed approaches that are similar to the response logic. This includes disruption networks (Rung et al. 2002). While the response logic postulates that a node will show a response if it is reachable from a perturbed node by a path of any length, the disruption network considers reachability only between directly connected nodes. To describe all the observed responses under this assumption thus requires a densely connected network, also known as transitive closure. The counterpart to this idea is a description of the response data in terms of a transitive reduction (Wagner 2001). Here, nodes can be reached also via indirect paths and the transitive reduction is the network that describes the experimentally observed reachability relationships with as few edges as possible (Aho et al. 1972). This approach thus reflects the believe that biological networks should be sparse. Further refinements to the original method also accommodate for certain network cycles, positive and negative regulation, and double-perturbation data (Tringe et al. 2004). Combining these two opposing ideas, the TRaCE algorithm (Ud-Dean et al. 2014) examines both the transitive closure and the transitive reduction. It then considers any edge that appears in the closure but not in the reduction as non-identifiable and can thereby quantify the degree of identifiability of different classes of biological networks.

Yet in contrast to the response logic approach, these methods require a fully perturbed network, are sensitive to noise, and do not allow for an incorporation of prior network knowledge. Furthermore, they only identify singular (extreme) networks to describe the perturbation data. The response logic approach on the other hand provides the entire ensemble of conforming networks (including cyclic networks) which can then be further filtered to identify the most appropriate network description of the data.

In summary, the logic response approach is a network inference methods that is based on a simple assumption, which represents the way in which we often reason about the interactions within biological networks. This opens the approach for a wide range of applications. It retrieves networks without any implicit bias (such as through regularization). Rather, all the inference rules are explicit and simple, which allows for a coherent interpretation of the reconstructed networks.

3.1 PUBLICATION

Robust network inference using response logic

Torsten Gross, Matthew J Wongchenko, Yibing Yan, Nils Blüthgen
Bioinformatics, Volume 35, Issue 14, July 2019, Pages i634–i642
`https://doi.org/10.1093/bioinformatics/btz326`

This article was published in the proceedings of the ISMB ECCB 2019 in Basel. I presented it at the conference and a video recording of the talk can be accessed under the following URL.

Robust network inference using response logic - Torsten Gross - NetBio - ISMB ECCB 2019, *ISCB*, Aug 20, 2019, Accessed on: May. 15, 2020. [Video file].
Available: `https://youtu.be/tQhbhERf4rw?t=474`

A Python/Answer Set Programming implementation of the response logic approach can be accessed at:
`github.com/GrossTor/response-logic`

Bioinformatics, 35, 2019, i634–i642
doi: 10.1093/bioinformatics/btz326
ISMB/ECCB 2019

Robust network inference using response logic

Torsten Gross[1,2,3], Matthew J. Wongchenko[4], Yibing Yan[4] and Nils Blüthgen[1,2,3,*]

[1]Institut für Pathologie, Charité—Universitätsmedizin 10117 Berlin, [2]IRI Life Sciences, Humboldt Universität zu Berlin, 10115 Berlin, [3]Berlin Institute of Health, 10178 Berlin, Germany and [4]Oncology Biomarker Development, Genentech Inc., South San Francisco, CA 94080, USA

*To whom correspondence should be addressed.

Abstract

Motivation: A major challenge in molecular and cellular biology is to map out the regulatory networks of cells. As regulatory interactions can typically not be directly observed experimentally, various computational methods have been proposed to disentangling direct and indirect effects. Most of these rely on assumptions that are rarely met or cannot be adapted to a given context.

Results: We present a network inference method that is based on a simple response logic with minimal presumptions. It requires that we can experimentally observe whether or not some of the system's components respond to perturbations of some other components, and then identifies the directed networks that most accurately account for the observed propagation of the signal. To cope with the intractable number of possible networks, we developed a logic programming approach that can infer networks of hundreds of nodes, while being robust to noisy, heterogeneous or missing data. This allows to directly integrate prior network knowledge and additional constraints such as sparsity. We systematically benchmark our method on KEGG pathways, and show that it outperforms existing approaches in DREAM3 and DREAM4 challenges. Applied to a novel perturbation dataset on PI3K and MAPK pathways in isogenic models of a colon cancer cell line, it generates plausible network hypotheses that explain distinct sensitivities toward various targeted inhibitors due to different PI3K mutants.

Availability and implementation: A Python/Answer Set Programming implementation can be accessed at github.com/GrossTor/response-logic. Data and analysis scripts are available at github.com/GrossTor/response-logic-projects.

Contact: nils.bluethgen@charite.de

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Complex molecular networks control virtually all aspects of cellular physiology as they transduce signals and regulate the expression and activity of genes. Understanding those molecular networks requires an appropriate simplification of the stupefying complexity that we find in cells. A very successful and common abstraction in molecular cell biology is to define effective modules and map out their interactions (Ideker and Nussinov, 2017). But even though new experimental techniques can reveal and quantify countless cellular components in ever increasing level of detail, they typically cannot identify the relationships between them. This is why for more than two decades various methods were developed to infer gene regulatory networks, signalling pathways and genotype–phenotype maps (De Smet and Marchal, 2010). These methods vary widely in their notion of network (e.g. directed versus undirected, weighted versus unweighted links), their mathematical methodology (e.g. statistical measures versus model-based parameter fits) or their goals (e.g. interaction discovery versus network property characterization versus perturbation response prediction) (Basso *et al.*, 2005; de la Fuente *et al.*, 2004; Ghanbari *et al.*, 2015; Kholodenko *et al.*, 2002; Klamt *et al.*, 2006; Molinelli *et al.*, 2013; Natale *et al.*, 2017). Not surprisingly, different methods produce radically different results on same datasets (Marbach *et al.*, 2010; Meisig and Blüthgen, 2018). This makes for an intricate choice of method and guarantees a certain degree of arbitrariness in interpreting the inferred networks.

One major goal of network inference for signalling and regulatory networks is to derive directed networks, that is, to infer information about causal relations within the studied system. This differs profoundly from the inference of undirected associations between node pairs, such as by correlation, as it requires to trace the flow of

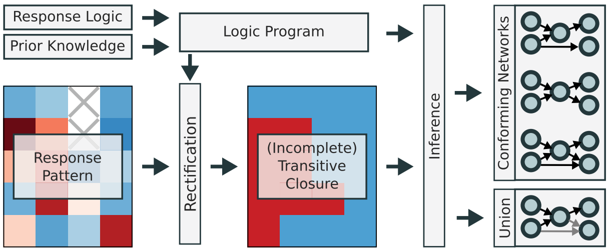

Fig. 1. The steps of the response logic approach. The response logic and all additional prior network knowledge are formulated as a logic program. It is first used to rectify the experimentally determined response pattern, and second, takes the resulting (potentially incomplete) transitive closure as input to infer either all individual conforming networks or the union thereof

information through the network. A popular approach is to use time-series data, for which methods like convergent cross mapping (Čenys *et al.*, 1991; Sugihara *et al.*, 2012) or Granger causality (Granger, 1969) can distinguish correlation from causation, given sufficiently dense samples. But most often, experimental protocols or excessive expenditures preclude the observation of suitable temporal trajectories for many contexts in molecular biology. Thus, a complementary approach is to observe the system's responses, for instance the steady-state response, to a set of localized perturbations (Bruggeman *et al.*, 2002; Sachs *et al.*, 2005; Wagner, 2001a). Depending on the specific system, these perturbations could, for example, be gene knockouts or kinase inhibitions. However, existing methods for such data rely on context-specific assumptions whose validity is hard or impossible to assess in practice, which makes it very difficult to interpret their results. Facing this challenge, we asked whether we could derive a more generally applicable scheme for the inference of directed networks—a method that is based on a principle which is accurate enough for most contexts while also sufficiently simple to allow for an intuitive understanding of how the network structure was resolved. Furthermore, we noticed that even though most network inference problems are embedded within very well-studied contexts, the vast majority of reverse-engineering methods predicts networks *de novo*. Therefore, we additionally aimed for a method that could readily incorporate prior knowledge about presence or absence of certain links or about other known network properties. This resulted in what we call the response logic approach.

In the following, we describe the response logic approach in more detail and then benchmark it by (i) assessing the performance using synthetic data derived from KEGG pathways (Kanehisa *et al.*, 2017), and (ii) comparing its performance to competing methods using community-wide inference challenges (Dialogue for Reverse Engineering Assessments and Methods, DREAM) (Stolovitzky *et al.*, 2007). Finally, we use the approach to study RAS/MAPK/PI3K signalling in a colon cancer cell line, and predict differences in the signalling network topology due to different PI3K mutants, that manifests in differential sensitivity of a colon cancer cell to various targeted drugs.

2 Materials and Methods

We developed a method to infer directed network structures from perturbation data that we term response logic (see Fig. 1). As an input, this method requires binary information about which nodes in the network respond to which perturbation, together with a rank of confidence of each data point. We refer to this set of experimental observations as the response pattern. Given this information, the response logic approach infers networks that agree to the following simple rule: a perturbation at a node is propagated along all outgoing edges to the set of connected nodes, and these responding nodes will in turn propagate the signal and so forth. Consequently, a perturbation of a node causes a response at all nodes to which it is connected by a path, and no response at all others. The information about which node can be reached from which other nodes is known as the network's transitive closure. Thus, the central assumption of our response logic approach is that experimentally observed responses are in agreement with the transitive closure. This assumption then leads to the inverse problem of identifying the networks whose transitive closure actually matches the response pattern.

The algorithm to infer these networks consists of two main steps. Using a logic programming approach, it first modifies the experimental response pattern to match a transitive closure (rectification step) and then infers either all individual networks that comply to the given data or the union over all those conforming networks. We will describe the different steps in the following sections.

2.1 Rectifying the response pattern

The response logic approach interprets the measured response pattern as a noisy, incomplete transitive closure. But because of misclassification, a response pattern might not match any actual (incomplete) transitive closure. Consider for example a three-node network in which all nodes are observed to respond to a perturbation at node one. This implies two paths, from nodes one to two and nodes one to three. Therefore, if a perturbation at node two causes a response at node one, node three is expected to respond as well. But assume that this response at node three was not observed (misclassification). Then, there is no directed network with a transitive closure that would match this response pattern. We expect that such misclassification occurs rather often when working with experimental data because of experimental noise or because the system under consideration does not fully comply with the assumptions of the response logic. Thus, it is necessary to identify the most relevant subset of the response pattern that forms an (incomplete) transitive closure which can then be used to infer networks.

Our rectification algorithm requires to rank the observations of the response pattern from most to least confident. Typically, such confidence levels are readily available since the response pattern is often derived from a binarization of continuous experimental readouts, in which case a confidence score could be the distance to the binarization threshold, or a score of statistical significance. The algorithm then iterates the elements of the response pattern from high to low confidence, and at each step, determines whether the so far collected elements form a transitive closure and also conform with additional constraints from prior knowledge. This is done using a logic program (see below), which determines if there is any network that is compatible with these elements of a transitive closure. If the new element is compatible, it is added to the collection of conforming data and otherwise discarded. The more data points enter the collection the more restrictions apply to the remaining elements of the response pattern. As high confidence observations are taken into account first they are thus less likely to be discarded, ensuring that we extract the most relevant subset of the response pattern that indeed forms an incomplete transitive closure that is in line with additional constraints. If confidence levels of different data points cannot be easily distinguished, it is recommendable to repeat the response logic analysis for alternative rankings and inspect how this impacts the set of compatible networks.

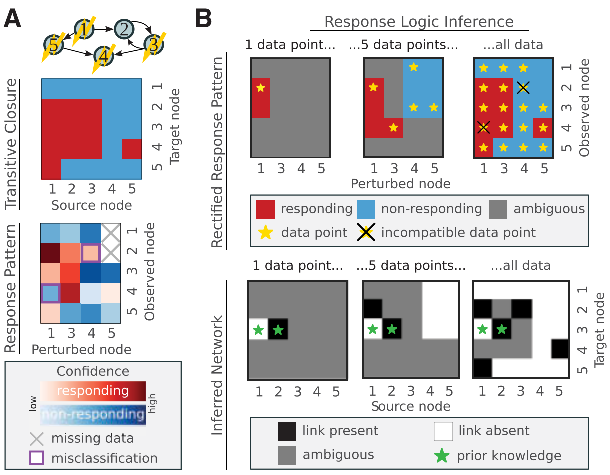

Fig. 2. Response logic inference of a toy network. (**A**) From top to bottom: an example network of five nodes, where flashes indicate which nodes were perturbed; the full transitive closure; the response pattern that captures parts of the network's transitive closure, with missing or misclassified data and confidence scores. (**B**) Three instants during data rectification: data points are added sequentially from high to low confidence (stars in top row), and increasingly constrain the inferred network and the (rectified) response pattern (red and blue fields in top row). Bottom row shows the inferred network at the given instant during rectification

Figure 2 demonstrates this scheme for a toy network of five nodes, for which we assume that four nodes were perturbed (indicated as flashes in Fig. 2A, top). The resulting response pattern then consists of a five-by-four matrix, and we assume that two data points are missing, and two elements of the response pattern do not match the transitive closure (compare heat maps in Fig. 2A). In Figure 2B, we exemplify how the response pattern is iteratively rectified. We assume that we know that a link from node two to node three exists and that there is no link between node one to node three (green stars). Given this prior knowledge, already the first (highest confidence) data point (yellow star in the leftmost panel) additionally implies that node three also responds to a perturbation of node one. Any subsequent data point that is in conflict with this information will be discarded. The middle panels of Figure 2B show that the five most trusted data points constrain five other elements of the rectified response pattern. Among them are the two misclassified, as well as the two missing data points. Therefore, in this toy example, the high confidence data points automatically correct these false or missing pieces of information. The bottom panel of Figure 2B shows how adding data points increasingly constrains the network structure. Once all data are considered, most of the links (but not all, as discussed later on) are known to be either present or absent from the network. Note, however, that the rectification process does not require to compute the shown union of conforming networks, but only requires to determine if for any network at all, all constraints are satisfiable, which is computationally far less expensive.

2.2 Finding conforming networks with logic programming

Mapping the response pattern to its corresponding set of conforming networks is a substantial computational challenge, as there are $2^{N \cdot N}$ possible directed networks (with N being the number of nodes), making it infeasible to enumerate all networks even for small sizes. We therefore solve the search problem with a logic programming approach, which is a form of declarative programming where the problem is represented via a set of logical rules. We chose to use the logic modelling language Answer Set Programming (ASP) (Baral, 2003), as implemented in the Potsdam Answer Set Solving Collection (Gebser *et al.*, 2011). For ASP solving, we apply the *clingo* (Gebser *et al.*, 2014) system.

ASPs generate-define-test pattern (Lifschitz, 2002) allows for a convenient encoding of the response logic, which is detailed in Supplementary Material S1. In short, we generate the collection of answer sets, consisting of all possible network structures, then define auxiliary predicates, in our case the networks' transitive closure, and then test whether this transitive closure agrees with the data and also whether the tested network complies to all other heuristic constraints. Then the ASP solver, *clingo*, allows to enumerate all conforming networks. Note that the computational effort needed to identify a conforming network heavily depends on network size and the provided heuristic constraints. But overall, the logic programming approach infers networks of up to 100 nodes within seconds, without any parallelization.

The previously discussed data rectification sequentially checks the satisfiability of every data point and could therefore become a performance bottleneck for large systems. However, because this process only requires to decide whether any network at all is in agreement with the latest data, instead of having to provide the entire set of conforming networks, we can solve a much simpler logic program, which is detailed in Supplementary Material S1. It drastically improves performance because it does not require to define an answer set for each possible network structure.

2.3 Identifiability and heuristic constraints

While every directed network has a single transitive closure, a transitive closure can often be mapped to many different networks, even more so if the transitive closure is only partially known. Thus, we can usually not infer a unique directed network from a rectified response pattern alone. For example, any feedback loop creates a strongly connected network component, that is, a set of nodes for which any pair is connected by a path. Therefore the response pattern is independent on how exactly the nodes are connected to each other. Similarly, the response pattern does not change with any additional feed-forward loops that cuts short an existing path. To resolve such structures we need to resort to additional constraints that are derived from contextual knowledge about the studied system. A crucial advantage of the response logic approach is that it can easily integrate various kinds of such constraints. Here, we want to exemplify this and introduce those constraints that are used in the applications shown further below.

Rarely will we analyze networks that have never been studied before. Therefore one can use prior knowledge to constrain networks, such as by requiring the presence of well-established links in the network, or by excluding links that are physically not feasible (such as interactions between molecules located in different compartments). This information can directly be integrated into the logic program by defining the presence or absence of links as additional constraints. In addition, the logic program can also accommodate more subtle constraints, such as to enforce bounds on the numbers of incoming and outgoing edges of (groups of) nodes, see the implementation in Supplementary Material S1. This allows, for example, to encode the information that a module of nodes signals to other parts of the network without having to explicitly state which of the module's nodes has the outward link. The same idea holds for a module that is known to receive at least one input to any subset of its components. Note that these types of constraints directly limit the space of possible networks and in turn that of the transitive closures. They will thus influence how the response pattern can be rectified and must be taken into consideration during the process.

But even these additional constraint might not sufficiently limit the number of conforming networks to consider them individually. Alternatively, an extension of the logic program, described in Supplementary Material S1, allows to efficiently find the union of all answer sets. This union reveals which links (or missing links) appear in all solutions and which are ambiguous. The latter is particularly informative to either guide the choice of additional perturbation experiments or to reveal effective strategies on how to further filter the set of solutions.

One widely used strategy in this regard is to require an overall sparse architecture (Wagner, 2001b). We would thus want to identify the conforming networks with the fewest links. However, naïvely parsing all network solutions will be infeasible when the set of solutions is large. To overcome this problem we developed an algorithm that sequentially removes as many ambiguous links as possible, without violating any constraint. To do so, after every link removal the pruned network is tested for satisfiability. If it complies to the given constraints, the link remains removed and the procedure continues. Otherwise the link is considered necessary and the procedure continues without the removal of the link. This leads to what will be referred to as the *sparsified network*. Yet, such scheme is only reasonable if the order by which links are removed, reflects to some extent a knowledge about which links are more likely to be absent in the underlying network, and should therefore be tested for removal first. However if such information is not available, one can use yet another approach to filter for sparse networks, termed the *parsimony constraint*. This constraint asks whether a link from a conforming network can be removed without it changing the network's transitive closure. If that is the case, the network is considered non-sparse and is removed from the solution set. The specific encoding is found in Supplementary Material S1. While this procedure does not generally single out a unique solution as before (multiple networks can be parsimonious), it was nevertheless observed to drastically reduce the solution space.

Taken together, a response pattern will typically be compatible with a large number of network topologies, but various types of prior network information can be incorporated into the response logic approach to reveal a finer network structure than what would have been possible from the response pattern alone. At the same time, the approach states explicitly whether or not the presence or absence of a link can be inferred from the given data and constraints.

2.4 Implementation and data acquisition

The response logic approach is implemented in Python 3.6 as a package available at github.com/GrossTor/response-logic. Numerical computations, data handling and plotting was done using the SciPy libraries (Jones *et al.*, 2001) and seaborn. Additional functions were taken from the networkx package (Hagberg *et al.*, 2008). Clingo's python API (version 5.2.2) (Gebser *et al.*, 2014) is used to ground and solve the Answer Set Programs.

The repository contains all Answer Set Programs, which are accessed by the main *response.py* module. It includes the *prepare_ASP_program* function to set up a logic program according to the provided data and additional constraints, the *conform_response_pattern* function that rectifies the response pattern, as well as various functions to solve a logic program. Additionally, a projects repository available at github.com/GrossTor/response-logic-projects includes all scripts and data that were used to obtain the results from the following sections.

KEGG data (Kanehisa *et al.*, 2017) was retrieved via the KEGG package within the biopython library. The KEGG pathway maps database was parsed for human pathways and the retrieved KGML files were used to build network representations based on their 'relation elements'.

The data and evaluation scripts for the DREAM3 and DREAM4 challenge was retrieved with the official DREAMTools python package (Cokelaer *et al.*, 2016). Leaderboards were taken from Cokelaer and Costello (2015) and Figure 3 from Marbach *et al.* (2010).

The SW-48 perturbation data were generated using a SW-48 cell line, and two derived clones with mutations in PI3K. Cell lines were obtained from Horizon Discovery. All lines were maintained in RPMI (Invitrogen) with 10% FBS (Invitrogen). Cell growth was assessed using the Cell Titer 96 Aqueous One Solution Cell Proliferation Assay (Promega). Cells were treated with compound 24 h after plating and grown for 72 h. The cell growth was determined by correcting for the cell count at time zero (time of treatment) and plotting data as percent growth relative to vehicle (DMSO)-treated cells. Reverse-phase protein array (RPPA): cells were treated 24 h after plating and incubated with inhibitor (GDC0973, GDC0068, Erlotinib) or solvent control (DMSO) for 1 h, and then stimulated either with EGF, HGF and IGF or with control (BSA) for 30 min. Cells were lysed in T-PER (Thermo), 300 mM NaCl, cOmplete® protease inhibitor (Roche) and Phosphotase Inhibitor Cocktails 2, 3. RPPA measurements were carried out by Theranostics Health. All data can be accessed from the according data folder in the projects repository (response-logic-projects/SW-48_analysis/data/).

3 Results

3.1 Performance assessment on KEGG pathways

We first set out to systematically quantify how misclassification and missing data in the experimentally determined response pattern impacts the quality of the predicted network structure. To this end, we inferred network structures from synthetic datasets. As a relevant and representative collection of test networks, we extracted all 270 human gene regulation and signal transduction networks (maximally containing 100 nodes) from the KEGG pathway database (Kanehisa *et al.*, 2017). For each of these network structures we generated its transitive closure, which we considered as the immaculate response pattern. Then, we repeatedly generated a random confidence pattern, C, where each entry is drawn from a uniform distribution between 0 and 1. To evaluate the effect of missing data, we remove a fraction ϵ_M of data points from the perfect response pattern and to evaluate the effect of measurement error, we also misclassify a fraction ϵ_C of the remaining data points. Missing or misclassified data points were chosen with a probability that was proportional to their confidence score C_{ij}. We then used the resulting response and confidence patterns to infer the sparsified network, as defined in the previous section, via the response logic approach and, comparing it to the original KEGG network, computed precision and recall as performance scores, see Figure 3A.

For each of the 270 KEGG networks the procedure was repeated 50 times for different choices of ϵ_M and ϵ_C, and the mean of the scores is shown in Figure 3B. In the absence of misclassifications ($\epsilon_C = 0$, red and orange dots in Fig. 3B), prediction errors stem exclusively from the previously discussed multitude of conforming network structures.

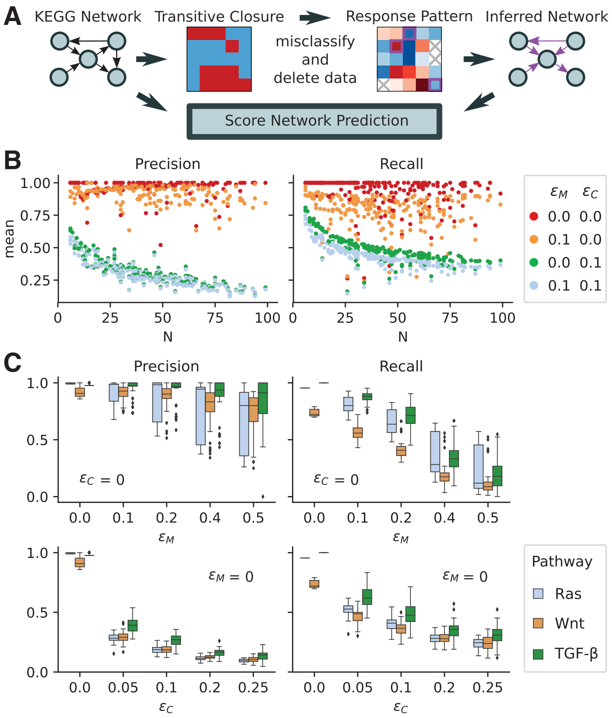

Fig. 3. The performance of the response logic approach on synthetic data generated from 270 human KEGG pathways (Kanehisa *et al.*, 2017). *N* denotes network size, ϵ_M quantifies the fraction of missing and ϵ_C the fraction of misclassified data points. (**A**) Data generation and scoring scheme. (**B**) Each dot per colour represents a different pathway, colours represent different parameters for misclassification (ϵ_C) and missing data (ϵ_M). (**C**) Precision and recall for three particular signalling pathways as a function of the fraction of misclassified or missing data

Interestingly, for a vast set of biological pathways the resulting inference errors are rather mild, and highly accurate predictions can be made independent of network size. However, once misclassifications are present, the predictivity is markedly reduced. Interestingly, this effect increases with growing network size.

We next examined the dependency on missing data and misclassification rates in more detail for the three signalling pathways: RAS, Wnt and TGF-β. We chose to scan the parameters from 0.0 to 0.5 and 0.0 to 0.25 for ϵ_M and ϵ_C, respectively, as a complete loss of information would either occur when all data were missing, $\epsilon_M = 1$, or half of the entries were misclassified, $\epsilon_C = 0.5$ ($\epsilon_C = 1$ would produce an inversion of the response pattern). For all pathways, we found that recall is more affected by missing data than precision (see Fig. 3C). That is, with less data the predicted links remain rather accurate but fewer of them are predicted. We also confirmed our previous finding that misclassification reduces prediction scores much stronger than missing data. Interestingly, even when half the data were discarded, in many instances precision remained still close to one. This suggests that discarding low-confidence data points rather than risking to accept many misclassified data points might be a good strategy to improve predictions. We will re-examine this idea by the end of the next section.

3.2 Response logic approach outperforms competing methods in DREAM challenges

The DREAM (Stolovitzky *et al.*, 2007) provides community-wide reverse-engineering challenges that foster the development of new systems biology models. Particularly, the DREAM3 and DREAM4 *in-silico* challenges (Greenfield *et al.*, 2010; Marbach *et al.*, 2010) assessed the performance of various gene network-inference methods and have since become a standard benchmark to which we can compare the response logic approach. In these two challenges various biologically plausible *in-silico* gene networks of different sizes were simulated under stochastic conditions to emulate realistic transcription dynamics resulting from knockdowns and knockouts of each single gene. Participants were given the resulting time courses, the steady states and the wild-type level of each gene and asked to infer the directed network structure from them. A ranked list of predicted gene pair interactions was then compared against the gold standard from which the area under the precision–recall (PR) and the area under the receiver operating characteristic curve (ROC) are computed, see Supplementary Figures S1 and S2. Comparing these to a null model provides *P*-values for each of the given five networks per network size that then get combined into a single overall score (Stolovitzky *et al.*, 2009).

To infer the DREAM networks with the response logic approach, we generated response patterns from the *in-silico* knockout experiments of these challenges only (not considering knockdown or time-series data). These were computed as follows. When K_{ij} denotes the level of gene *i* after knockout of gene *j*, and the wild-type levels are **w**, we defined the normalized global response matrix, *R*, as

$$R_{ij} = \frac{|K_{ij} - w_i|}{s_i},$$

with s_i being the standard deviation of the knockout levels of gene *i* (row *i* of *K*). We then defined gene *i* to be responding to a knockout of gene *j* if $R_{ij} > 1$. The entries of the associated confidence matrix were defined as a normalized distance of knockout levels to this threshold, see Supplementary Material S2. We then applied our response logic approach to these matrices to infer sparsified networks, as defined earlier. The goal of the DREAM challenge is to provide a list of gene pairs that is ranked by their predicted likelihood to be interacting. We generated it by first listing the predicted interacting and then the non-interacting gene pairs, where within each group, the pair list was ordered according to the associated entries in the global response matrix (interaction $i \to j$ was ranked higher than $k \to l$ if $R_{ij} > R_{kl}$). As comparison, we also created a ranked list by simply ranking gene pairs in the order of the global response matrix, without the grouping that was introduced by the response logic, which we termed ‘naïve approach’.

These ranked lists were then scored using the official DREAMTools package (Cokelaer *et al.*, 2016) (with a minor modification for one network score at DREAM3 $N = 100$, see Supplementary Material S2). Figure 4A shows the results of our method and that of the naïve approach in comparison to the 10 best performing participants at each network size and challenge that were provided with the full (knockout, knockdown, time-course) datasets. Except for the small networks with $N = 10$, where the response logic approach ranks second and third, it outperforms all 29 competitors participating in DREAM3 (Marbach *et al.*, 2010), as well all 29 competitors participating in DREAM4 (Cokelaer and Costello, 2015). Note that the best performers for the small networks ($N = 10$) that scored higher than the response logic (Küffner *et al.*, 2010; Yip *et al.*, 2010) also used the provided time-course data, which we did not use in our response logic approach.

We also observed that the response logic always outperformed the naïve approach, confirming that non-trivial additional knowledge is gained when applying the response logic. Notably however,

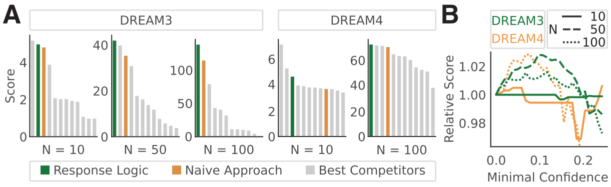

Fig. 4. (**A**) Performance of the response logic approach for the gene-network reverse engineering challenges DREAM3 and DREAM4 (Greenfield *et al.*, 2010; Marbach *et al.*, 2010) (green bars), compared with a 'naïve' scoring approach (orange bars) and the 10 best approaches that took part in the respective challenges (grey bars). Scores are calculated as in the original challenge, with higher scores indicating better performance. (**B**) Relative changes in performance when excluding data points with confidence below a certain threshold. *N*: network size

already the naïve approach scores comparatively well, which let the challenge's organizers to conclude that 'sophisticated methods that would in theory be expected to perform better than the naïve approach described above, were more strongly affected by inaccurate prior assumptions in practice' (Marbach *et al.*, 2010). This observation affirms our initial motivation to design an approach with minimal assumptions on the data.

Finally, the DREAM data also allowed us to test if disregarding low-confidence data points, as suggested by the KEGG pathway analysis, improves predictions. Thus, considering the confidence matrix with scaled entries between 0 and 1 (Supplementary Material S2), we removed data points with confidence scores below a threshold and re-engineered the networks from those smaller datasets. The resulting scores relative to the original scores, which were obtained from the full response patterns, are shown in Figure 4B. With the exception of the $N = 10$ networks, these numerical experiments confirmed that removing low-confidence data effectively improved network inference. Peak performance is reached when approximately 5% of the data are discarded.

In summary, our benchmarks using the DREAM *in-silico* challenges provide a strong indication that the response logic approach is capable of reverse-engineering biological networks. Its simplicity not only makes its results comprehensible but the DREAM challenge showed that they are also more accurate than those of existing methods.

3.3 Reverse engineering MAPK and PI3K signalling in a colon cancer cell line

Having benchmarked the response logic formalism, we next used it to investigate signalling networks in cancer cells. In a first step, we decided to reverse engineer the Ras-mediated signalling network including MAPK and PI3K/AKT signalling in SW-48 colon cancer cells. We performed multiple perturbation experiments using either ligands or inhibitors that targeted EGFR, PDFR, ERK and AKT, and measured changes in phosphorylation using a reverse-phase protein assay (RPPA) platform. Ten of the antibody-based readouts passed a quality control and were relevant to the considered pathways, see details in Supplementary Material S3. Using replicate measurements of both unperturbed and perturbed conditions, we constructed the response pattern as well as the according confidence scores, which are shown in Figure 5A (see Supplementary Material S3 for details).

The RAS signalling network has been well studied, which allowed us to compile a literature network shown in Figure 5C that can be used as a gold standard to measure prediction performance. We then applied our response logic framework to the response

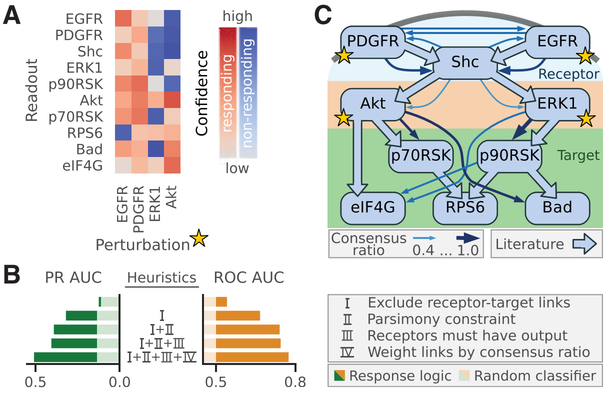

Fig. 5. (**A**) Response pattern of the SW-48 cell line of selected phospho-proteins after perturbations affecting EGFR, PDGFR, ERK1 and AKT. (**B**) Performance of response logic network inference under various (combinations of) heuristics, as explained in text, compared to a random classifier (shaded colours). (**C**) Literature network (filled arrows) and final network prediction (finer arrows, only links with consensus ratio ≥0.4 are shown)

pattern, and evaluated predictions by means of the areas under the ROC-, as well as PR curves, as shown in Figure 5B, see Supplementary Material S3 for details. As it was computationally impossible to enumerate all networks, we determined the union of all conforming networks, as described earlier, and scored links based on whether they are found in all, in some and in no conforming networks. Doing so led to PR and ROC curves that were only marginally better than random (top row in Fig. 5B). The apparent challenge concerning the network inference for this network is the substantial disparity between 10 readouts to only 4 perturbations, making the reverse engineering problem strongly underdetermined. A crucial benefit of the response logic analysis is that it allows for the incorporation of various additional insights about the structure of signalling networks to reduce the space of solutions. We therefore investigated how the inclusion of generic and indirect network knowledge rendered the analysis more meaningful. First, we enforced a hierarchy in the network (heuristic I). Signalling networks typically process signals received on the receptor level through a chain of intermediate kinases, before they are passed on to a set of targets. We therefore disallowed any direct connections between the receptor and the target level (according to the allocation shown in Fig. 5C) (these ruled-out links were obviously not taken into account for the scoring procedure, which explains the different areas under the PR curve for the random classifier in Fig. 5B). Furthermore, kinase interactions are highly specific, resulting in sparse signalling networks. Therefore, we found it reasonable to rid the network of redundant links and apply the parsimony constraint, as defined earlier (heuristic II). Lastly, we required that any node at receptor level must have at least one outgoing link (heuristic III).

Adding these three heuristics, I–III in Figure 5B, considerably improved the performance and reduced the solution space to 666 conforming networks. This makes it possible to enumerate them all and compute for each possible link the fraction of how many times it was present in all conforming networks (consensus ratio). We reasoned that a higher consensus ratio also implies a higher relevance, which we found confirmed when using the consensus ratio, rather than the union of networks to score the predictions (heuristic IV). From these results, we conclude that the response logic is indeed a valid assumption for the MAPK and PI3K pathway activity in the SW-48 cell line and that rather apparent additional information can effectively compensate for the small number of perturbations.

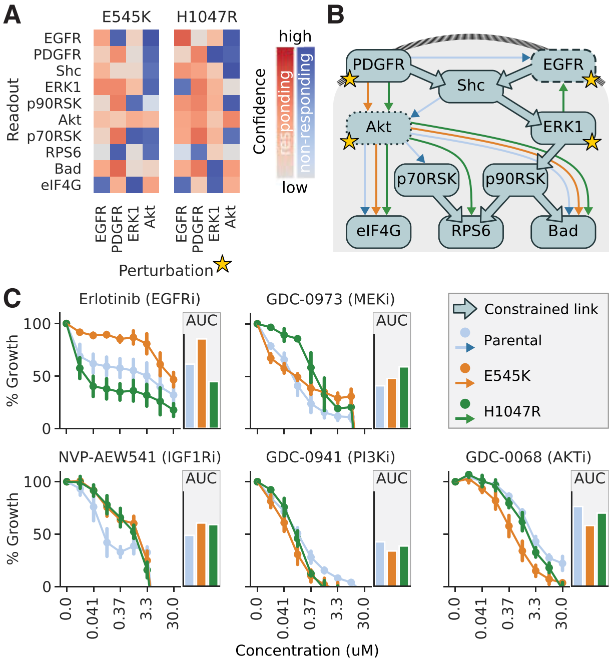

Fig. 6. (**A**) Response pattern for two PI3K-mutant cell lines derived from SW-48, carrying either the E545K or H1047R mutations in PIK3CA, as in Figure 5. (**B**) Mutant-specific networks derived from the response data arrows in blue (parental cell line), orange and green (two mutant cell lines), with links constrained due to literature knowledge shown with large arrows. (**C**) Dose–response curves for different inhibitors targeting the inferred networks in the parental cell line (blue) and the two clones with PI3K mutations (orange and green), and the area under the curve displayed as bar graphs

3.4 Modelling the effects of PI3K mutations in a colon cancer cell line

Having verified the validity of the response logic approach on the SW-48 cell data, we next used it to investigate how different mutations in the PI3K change signalling. To investigate this, we used clones of SW-48, in which two mutations that are commonly found in tumours were integrated, namely PIK3CA^{H1047R} and PIK3CA^{E545K}. We generated data using the same scheme as before, by perturbing the cells with ligands and inhibitors, and measuring the response using RPPA. Considering that the MAPK and PI3K pathways are very well studied, we assumed that the literature network depicted in Figure 5C is valid for all three cell lines, except for those links that could be affected by the different PI3K mutations. Because PI3K is not among the readouts, we model PI3K mutations to potentially affect links from and to its next downstream target, which is AKT. Furthermore, the literature network does not include context-dependent feedbacks in the MAPK pathway (Lake *et al.*, 2016). As we observed mutant-dependent upregulation of EGFR as well as SHC upon MEK inhibition, see Supplementary Figure S4, we considered this option in the inference as well. Therefore, to model the different mutant response patterns shown in Figure 6A, we used a heavily constrained response logic approach in which the presence or absence of network links is governed by the literature network, except for those links going in and out of AKT and those links going into EGFR. Not only did these constraints compensate for the few perturbations but also connect differences in the data to plausible alterations of the network. Furthermore, as the parsimony constraint has proven beneficial in the response logic validation on the parental cell line data, Figure 5, it is used as well (with the constrained literature network, previous heuristics I and III no longer apply, and IV is not relevant as shown next).

This approach resulted in four, one and two conforming networks for the parental, the E545K and the H1047R cell line, respectively. For the two ambiguous cell lines, we decided to isolate the single, biologically most plausible network hypothesis. In the case of the parental cell line, the four conforming networks consist of the combined options of whether or not SHC feeds back to EGFR and whether EGFR signals directly to AKT or via SHC. SHC has been found to be an adapter protein that is recruited to the activated EGFR (but does not activate it) and is essential for the receptor's signal relay (Ravichandran, 2001). We thus chose the parental network hypothesis that excludes the SHC to EGFR and the EGFR to AKT link. The two H1047R networks only differed in whether a feedback to EGFR originates from p90RSK or from ERK. Since the ERK to EGFR feedback is well described in the literature (Lake *et al.*, 2016), we decided for that option. With this, we could compare the mutant-specific network hypotheses, as shown in Figure 6B, which led to two main observations. First, in contrast to the parental cell line, the mutant cell lines do not have a link from the EGFR receptor to the PI3K pathway. And second, the H1047R cell line is the only one bearing a feedback from ERK (or any node) to EGFR.

We next aimed to explore if these different network topologies might explain phenotypic differences between these cell lines. We therefore evaluated the drug response of these cells for different targeted drugs, as shown in Figure 6C. Some differences in drug response can be understood directly from the mutations that have been added to the cell lines: the PI3K and AKT inhibitors seem to be slightly more effective in the PI3K-mutant cell lines, which is not surprising as these cells have constitutively active PI3K signalling. Similarly, inhibition of IGFR was more effective in the wild-type cells, as the mutant cells are more self-sufficient in PI3K signalling and therefore potentially require less IGFR activity. The drug responses to the EGFR inhibitor, and the MEK inhibitor are more complex and can only be interpreted when considering the network rewiring. The PI3K^{H1047} mutant cell line is rather resistant to the MEK inhibitor. This can be understood by the presence of the negative feedback from ERK to EGFR in this cell line, which is known to cause resistance by re-activating ERK and amplifying AKT signalling upon MEK inhibition (Klinger *et al.*, 2013). EGFR inhibition effects the cell line with the E545K mutant less, and the cell line with the H1047R mutant more strongly compared to the parental cell line. Both mutants decouple the EGF-receptor to the AKT pathway, so one would expect that they also show a less pronounced effect upon its inhibition. However, in the H1047R cell line there is a strong ERK-EGFR feedback that generally reduces the MAPK pathway activity, and one can speculate that additional EGFR inhibition can push the MAPK pathway activity to sub-critical levels.

Taken together, the response logic modelling allows to reconstruct networks from complex perturbation data and provides network information that can be interpreted and linked to phenotypic behaviour. This example demonstrates how this approach allows to integrate noisy response data, prior network knowledge and generic signalling constraints to identify hypothesis on changes in networks due to mutations that can subsequently be studied experimentally.

4 Discussion

We developed the response logic approach as a method to infer directed networks from perturbation data. Its central idea is to assume that a perturbation of a node is propagated along the edges and thus causes a response at all nodes to which there is a directed path, starting from the perturbed node. This simple hypothesis is integrated in a logic program that allows to identify the networks whose transitive closure most closely matches that of the experimental data. The power of logic programming, and more generally declarative programming, has enabled its use in a wide range of topics in computational biology (Backofen and Gilbert, 2001; Becker *et al.*, 2018; Bockmayr and Courtois, 2002; Dunn *et al.*, 2014; Videla *et al.*, 2015; Yordanov *et al.*, 2016). In our approach, logic programming provides a way to efficiently scan the vast search space of all directed networks and to easily express and incorporate additional information and prior knowledge about the network.

Many reverse-engineering methods involve tunable parameters, which can drastically affect the results. However, it is often far from obvious how these parameters should be set in a specific context. In contrast, our response logic approach is parameter free and strictly infers the networks that follow from the provided response pattern and any additional constraints provided.

At first glance, it might seem wasteful to reduce the data to a binary information of responding versus non-responding, when many experimental techniques allow to quantify the magnitude of response of the observed components. However, data binarization renders inference more general and robust, and in many settings, technical issues such as measurement error, heterogeneous data sources or various normalization steps, make the interpretation of magnitudes difficult.

The idea to map an experimentally observable response pattern onto a transitive closure has been proposed before. It was hypothesized that the sparsest directed acyclic graph whose transitive closure matches the observed response pattern describes the direct regulatory interactions in gene networks (Wagner, 2001a). Such a graph is also known as the transitive reduction and can be computed efficiently (Aho *et al.*, 1972). This approach was heuristically expanded to also allow for some cycles, and to refine the inferred network by incorporating double mutant perturbations and information about up- and downregulation (Tringe *et al.*, 2004). Yet, this procedure has several shortcomings: it cannot incorporate existing domain knowledge, it cannot handle missing data points, but simply considers an unknown or uncertain response behaviour as non-responding and it only finds a single, most parsimonious, network, which might not necessarily represent the underlying structure.

This last point is a strategy to compensate for the fact that network inference is an inherently underdetermined problem, because the number of independent measurements generally falls short on the number of possible interactions (De Smet and Marchal, 2010). The response logic approach explicitly addresses this problem as it considers the entire ensemble of conforming networks rather than to single out a particular one, based on some fixed and potentially inaccurate assumption. It thereby reveals which parts of the network cannot be inferred from the information provided so far. This important insight can then be used to either guide additional experiments or to systematically reduce the solution space by adding constraints that are most warranted in the given context. We consider this as a crucial advantage over existing approaches, whose inferred networks can generally not be intuitively traced back to the data and thus tend to disguise if and how the inferred network is justified by the data.

But while the response logic is based on a simple and intuitive concept, such simplicity comes at a price. As with any other assumption, it might not actually be representative of the studied system. Major problems might occur due to robustness, or saturation effects, all of which disrupt the presumed flow of signal but are an essential part of various biological systems (Fritsche-Guenther *et al.*, 2011). Also, from a Boolean perspective, the response logic treats nodes exclusively as OR gates, whereas certain systems require a more involved logic (Razzaq *et al.*, 2018). But again, the declarative nature of the ASP encoding allows to account for such effects. One could, for example, rather easily implement a maximal path length over which a perturbation gets attenuated, or explicitly state a Boolean function that governs the signal propagation of a certain node. Another important shortcoming for many questions is that it does not neither assign any weights nor signs to the inferred links. Yet, the inferred network can serve as an input for methods that are devised to quantify link strengths on a given input network (Dorel *et al.*, 2018).

On the other hand, the response logic's simplicity makes it suitable for various different fields of research. Because it is based on a formalization of an intuitive network behaviour, it can infer ecological, infection, or even social or transportation networks. Such generality would even permit to use the response logic to ask the inverse question: given a certain network structure and the observed perturbation responses, can I infer where a perturbation hit the network? This question could be particularly interesting in the analysis of man-made networks, for which the structure is typically known, but not the location of the perturbation. The inverted logic program would then identify where an electric connection malfunctioned, an intruder attacked or a disease originated from. All these possibilities show that the simplicity of the response logic does not limit its applicability.

Acknowledgements

The authors thank Dr Manuela Benary, Mathurin Dorel, Dr Bertram Klinger and Dr Johannes Meisig for their fruitful discussions.

Funding

This work was supported by a grant from the Berlin Institute of Health and by the German Research Foundation (DFG), through GRK1772 and RTG2424.

Conflict of Interest: none declared.

References

Aho,A. *et al.* (1972) The transitive reduction of a directed graph. *SIAM J. Comput.*, **1**, 131–137.

Backofen,R. and Gilbert,D. (2001) Bioinformatics and constraints. *Constraints*, **6**, 141–156.

Baral,C. (2003) *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge University Press, Cambridge.

Basso,K. *et al.* (2005) Reverse engineering of regulatory networks in human B cells. *Nat. Genet.*, **37**, 382–390.

Becker,K. *et al.* (2018) Designing miRNA-based synthetic cell classifier circuits using Answer Set Programming. *Front. Bioeng. Biotechnol.*, **6**, 70.

Bockmayr,A. and Courtois,A. (2002) Using hybrid concurrent constraint programming to model dynamic biological systems. In: Stuckey, P.J. (ed.), *Logic Programming, Lecture Notes in Computer Science*. Springer, Berlin Heidelberg, pp. 85–99.

Bruggeman,F.J. *et al.* (2002) Modular response analysis of cellular regulatory networks. *J. Theoret. Biol.*, **218**, 507–520.

Čenys,A. *et al.* (1991) Estimation of interrelation between chaotic observables. *Physica D*, **52**, 332–337.

Cokelaer,T. and Costello,J. (2015, July) Final leaderboards dream4 - in silico network challenge. https://www.synapse.org/Synapse: syn3049712/wiki/74631 (30 August 2018, date last accessed).

Cokelaer,T. *et al.* (2016) DREAMTools: a python package for scoring collaborative challenges [version 2; referees: 1 approved, 2 approved with reservations]. *F1000Research*, **4**, 1030.

de la Fuente,A. *et al.* (2004) Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics*, **20**, 3565–3574.

De Smet,R. and Marchal,K. (2010) Advantages and limitations of current network inference methods. *Nat. Rev. Microbiol.*, **8**, 717–729.

Dorel,M. *et al.* (2018) Modelling signalling networks from perturbation data. *Bioinformatics*, **34**, 4079–4086.

Dunn,S.-J. *et al.* (2014) Defining an essential transcription factor program for naïve pluripotency. *Science*, **344**, 1156–1160.

Fritsche-Guenther,R. *et al.* (2011) Strong negative feedback from Erk to Raf confers robustness to MAPK signalling. *Mole. Syst. Biol.*, **7**, 489.

Gebser,M. *et al.* (2011) Potassco: the Potsdam answer set solving collection. *AI Commun.*, **24**, 107–124.

Gebser,M. *et al.* (2014) Clingo = ASP + control: preliminary report. *CoRR*, abs/1405.3694.

Ghanbari,M. *et al.* (2015) Reconstruction of gene networks using prior knowledge. *BMC Syst. Biol.*, **9**, 84.

Granger,C.W. (1969) Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, **37**, 424–438.

Greenfield,A. *et al.* (2010) DREAM4: combining genetic and dynamic information to identify biological networks and dynamical models. *PLoS One*, **5**, e13397.

Hagberg,A.A. *et al.* (2008) Exploring network structure, dynamics, and function using networkx. In Varoquaux, G., Vaught, T., and Millman, J. (eds), *Proceedings of the 7th Python in Science Conference, Pasadena, CA*, pp. 11–15.

Ideker,T. and Nussinov,R. (2017) Network approaches and applications in biology. *PLoS Comput. Biol.*, **13**, e1005771.

Jones,E. *et al.* (2001) SciPy: open source scientific tools for Python. http://www.scipy.org/ (25 January 2018, date last accessed).

Kanehisa,M. *et al.* (2017) KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res.*, **45**, D353–D361.

Kholodenko,B.N. *et al.* (2002) Untangling the wires: a strategy to trace functional interactions in signaling and gene networks. *Proc. Natl. Acad. Sci. USA*, **99**, 12841–12846.

Klamt,S. *et al.* (2006) A methodology for the structural and functional analysis of signaling and regulatory networks. *BMC Bioinformatics*, **7**, 56.

Klinger,B. *et al.* (2013) Network quantification of EGFR signaling unveils potential for targeted combination therapy. *Mole. Syst. Biol.*, **9**, 673.

Küffner,R. *et al.* (2010) Petri nets with fuzzy logic (PNFL): reverse engineering and parametrization. *PLoS One*, **5**, e12807.

Lake,D. *et al.* (2016) Negative feedback regulation of the Erk1/2 MAPK pathway. *Cell. Mol. Life Sci.*, **73**, 4397–4413.

Lifschitz,V. (2002) Answer set programming and plan generation. *Artif. Intell.*, **138**, 39–54.

Marbach,D. *et al.* (2010) Revealing strengths and weaknesses of methods for gene network inference. *Proc. Natl. Acad. Sci. USA*, **107**, 6286–6291.

Meisig,J. and Blüthgen,N. (2018) The gene regulatory network of mESC differentiation: a benchmark for reverse engineering methods. *Philos. Trans. R. Soc. B*, **373**, 20170222.

Molinelli,E.J. *et al.* (2013) Perturbation biology: inferring signaling networks in cellular systems. *PLoS Comput. Biol.*, **9**, e1003290.

Natale,J.L. *et al.* (2017) Reverse-engineering biological networks from large data sets. arXiv: 1705.06370 [q-bio].

Ravichandran,K.S. (2001) Signaling via Shc family adapter proteins. *Oncogene*, **20**, 6322.

Razzaq,M. *et al.* (2018) Computational discovery of dynamic cell line specific Boolean networks from multiplex time-course data. *PLoS Comput. Biol.*, **14**, e1006538.

Sachs,K. *et al.* (2005) Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, **308**, 523–529.

Stolovitzky,G. *et al.* (2007) Dialogue on reverse-engineering assessment and methods: the DREAM of high-throughput pathway inference. *Ann. NY Acad. Sci.*, **1115**, 1–22.

Stolovitzky,G. *et al.* (2009) Lessons from the DREAM2 Challenges. *Ann. NY Acad. Sci.*, **1158**, 159–195.

Sugihara,G. *et al.* (2012) Detecting causality in complex ecosystems. *Science*, **338**, 496–500.

Tringe,S.G. *et al.* (2004) Enriching for direct regulatory targets in perturbed gene-expression profiles. *Gen. Biol.*, **5**, R29.

Videla,S. *et al.* (2015) Learning Boolean logic models of signaling networks with ASP. *Theoret. Comp. Sci.*, **599**, 79–101.

Wagner,A. (2001a) How to reconstruct a large genetic network from n gene perturbations in fewer than n2 easy steps. *Bioinformatics*, **17**, 1183–1197.

Wagner,A. (2001b) The yeast protein interaction network evolves rapidly and contains few redundant duplicate genes. *Mol. Biol. Evol.*, **18**, 1283–1292.

Yip,K.Y. *et al.* (2010) Improved reconstruction of in silico gene regulatory networks by integrating knockout and perturbation data. *PLoS One*, **5**, e8121.

Yordanov,B. *et al.* (2016) A method to identify and analyze biological programs through automated reasoning. *NPJ Syst. Biol. Appl.*, **2**, 16010.

Supplementary Material to: Robust network inference using response logic

Torsten Gross[1,2,3], Matthew Wongchenko[4], Yibing Yan[4], and Nils Blüthgen[1,2,3]

[1]Charité - Universitätsmedizin Berlin, Institut für Pathologie, Berlin, Germany
[2]IRI Life Sciences, Humboldt University, Berlin, Germany
[3]Berlin Institute of Health, Berlin, Germany
[4]Genentech Inc., Oncology Biomarker Development, USA

S1 Encoding the response logic in ASP

The response logic is formulated in the framework of Answer Set Programming. We use the syntax of the input language gringo. A straightforward introduction is provided in section 4 of (Videla et al., 2015). The following facts and clauses constitute the logic program.

At the beginning, the program needs to state the facts, i.e. the number of nodes in the network as constant `n`, the set of perturbations as functions `pert(I,OUT)`, where variable `I` names a specific perturbation and variable `OUT` indicates the node(s) targeted by the perturbation (could be multiple), as well as the rectified response pattern as functions `response(I,N)` and `-response(I,N)` (classical negation), where variable `I` names a specific perturbation and variable `N` indicates the node(s) responding or not responding to the perturbation, respectively. If a perturbation is defined to target multiple nodes, it is assumed to cause a response on all nodes that are path-connected to any of the perturbed nodes. Strictly speaking, such a response no longer corresponds to a transitive closure, as it describes a node's reachability from any of multiple nodes. But for the sake of simplicity, we will nevertheless keep the terminology.

The program then follows the generate-define-test pattern, as below. The function `tc(I,IN)` represents the transitive closure of the answer set network.

```
%%%Generate%%%
node(0..n-1).
{edge(OUT,IN)} :- node(IN), node(OUT).

%%%Define%%%
tc(I,IN) :- edge(OUT,IN), pert(I,OUT).
tc(I,IN) :- edge(OUT,IN), tc(I,OUT).

%%%Test%%%
```

```
:- not tc(I,N), response(I,N).
:- tc(I,N), -response(I,N).
```

The last two lines are the integrity constraints that discard any answer set whose transitive closure is in disagreement with the rectified response pattern. ASPs distinction between classical (”-”) and default (”not”) negation allows to only test the integrity for the known entries of the response pattern and to ignore the test for any entries that are not known.

For large networks the search space of the above program becomes very large with negative effects on performance. This is especially problematic during the rectification of the response pattern because in principle every entry of the response pattern must be tested for consistency. We implemented two strategies to speed up the process. First, at each step of the iteration, we check if the previous transitive closure is in agreement with the new data point. If so, we know that there is a conforming network and can accept the data point without an explicit test. Second, if there is no additional constraint (see below) stating the absence of a network edge, we can significantly simplify the logic program. In that case, if the logic program is satisfiable at all, the network with directly links from perturbed node to all its responding nodes, i.e. a network with star topology, constitutes a conforming network. As the rectification process only needs to determine satisfiability, a much more simple logic program avoids generating all possible networks but only tests the star network (which we do not need to state explicitly):

```
response(I,N) :- response(J,N), pert(J,M), J!=I,
                 response(I,M).
response(I,OUT) :- pert(I,IN), edge(IN,OUT).
response(I,N) :- response(J,N), pert(J,M), J!=I, pert(I,M).
```

The idea of this encoding is to define all additional responses that are implied by the star network from the given response pattern (line 1): if node N responds to a perturbation of node M (it is reachable from M) then node N will also respond to any other perturbation for which it is known that it causes a response at node M. This defines the minimal set of responding nodes consistent with the response logic. Any constraints on the absence of edges would imply (longer) non-direct paths that could potentially imply more nodes to respond to certain perturbations. This program does not even need to explicitly state an integrity constraint because an observed non-response given as classical negation `-response(I,N)`, already rules out that `response(I,N)` is contained in the (single) answer set. Furthermore, line 2 adds responses that are implied by edges that are known to exist and the last line is needed in the case where multiple perturbations hit the same node, but a perturbed node itself does not respond.

As discussed at length, we might wish to incorporate additional, external domain knowledge into the logic program. The most direct way of doing so is to simply state the knowledge about presence or absence of e.g. the link from node 2 to node 3 as a fact:

```
edge(2,3).
```

or

```
-edge(2,3).
```

respectively.

To encode bounds on the number of links that enter or leave one or a group of nodes is simple, due to ASPs built-in #count aggregates. If, for example, we were to maximally allow for 3 incoming edges to node 1 we would add the following integrity constraint to the logic program

```
:- #count {OUT : edge(OUT,1) } > 3.
```

Obviously, this statement could easily be adapted to formulate a lower bound, or bounds for groups of nodes.

It is more complex to state the parsimony constraint that filters out any network for which the removal of any link does not change the transitive closure. Essentially, for a given network, we define a set of links, x_edge(OUT,IN) that are subject to removal. Those are all the edges that are part of the network but which are not enforced by external knowledge. Then, the idea is to define all networks with a link removed, determine their transitive closure and compare it to that of the original network. The final integrity constraint removes the answer set if any of the reduced networks does not have a reduced (that is, it has the same) transitive closure as the original one.

```
r_edge(OUT,IN,NOUT,NIN) :- (OUT,IN)!=(NOUT,NIN),
                           edge(OUT,IN), x_edge(NOUT,NIN).
r_tc(I,IN,NOUT,NIN) :- r_edge(OUT,IN,NOUT,NIN), pert(I,OUT).
r_tc(I,IN,NOUT,NIN) :- r_edge(OUT,IN,NOUT,NIN),
                       r_tc(I,OUT,NOUT,NIN).
has_reduced_tc(NOUT,NIN) :- not r_tc(I,IN,NOUT,NIN),
                            tc(I,IN), x_edge(NOUT,NIN).
:- not has_reduced_tc(NOUT,NIN), candidate_edge(NOUT,NIN).
```

As this logic creates a significant amount of additional variables, we observed that the parsimony constraint will suffer performance issues when applied to more than a few tens of nodes.

Finally, a crucial feature of the response logic approach is the ability to generate the union of all conforming networks, which points out the links that can or cannot be uniquely determined. Typically, the number of conforming networks is intractable which precludes to simply enumerate all solutions and then compute their union directly. Rather, we let a logic program find it directly. The union of all answer sets of an ASP program is also termed brave consequences, hence the naming conventions below. In a first step, we need to annotate the presence or absence of an edge explicitly.

```
edge_brave(OUT,IN,1):-edge(OUT,IN).
edge_brave(OUT,IN,0):-not edge(OUT,IN),node(OUT),node(IN).
```

Then, the idea is to repeatedly solve the logic program, while iteratively building up the union of conforming networks. That is, we record for each edge whether the solutions generated so far, found it to be always absent, always present or neither (sometimes present, sometimes absent). To obtain the union over all answer sets, the program is further constrained with every new solve call to only permit networks that are not a subset of the union that was recorded so far. This is encoded by the following integrity constraint.

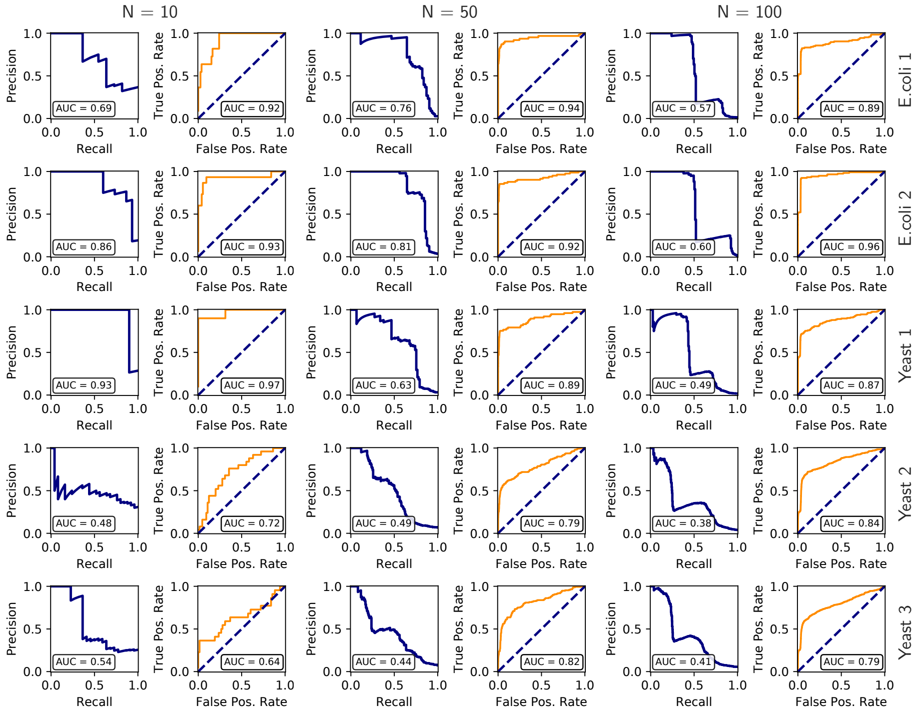

Figure S1: DREAM3 challenge 4: Precision Recall (blue) and ROC (red) curves for the response logic predictions of the five networks.

```
:- not edge_brave(OUT,IN,B) : node(OUT), node(IN), B=0..1,
                              @is_in_union(OUT,IN,B) == 0.
```

The `@is_in_union(OUT,IN,B)` construct is a function called by the solver that will return 1 if the presence or absence (`B` is 1 or 0, respectively) of edge `OUT` to `IN` is already recorded in the union and 0 otherwise. Thus, every solve call adds new elements to the union until there are no more conforming networks that show the presence or absence of a link, beyond what is already represented in the union. This is when the program is no longer satisfiable and the repeated solve calls stop.

S2 Additional information about the inference of DREAM in-silico networks

The rectification of the response pattern requires confidence scores, C. Those are defined as a normalized distance of each entry of the global response matrix R to 1, which was chosen as the response threshold. The normalization is chosen such that confidence levels range from zero, for $R_{ij} = 1$, to one, for R_{ij} taking either the maximal $R^{\max}$ or minimal $R^{\min}$ value of all entries. Formally,

$$C_{ij} = \begin{cases} |R_{ij} - 1| / |R^{\max} - 1| & \text{for } R_{ij} - 1 > 0 \\ |R_{ij} - 1| / |R^{\min} - 1| & \text{for } R_{ij} - 1 < 0. \end{cases}$$

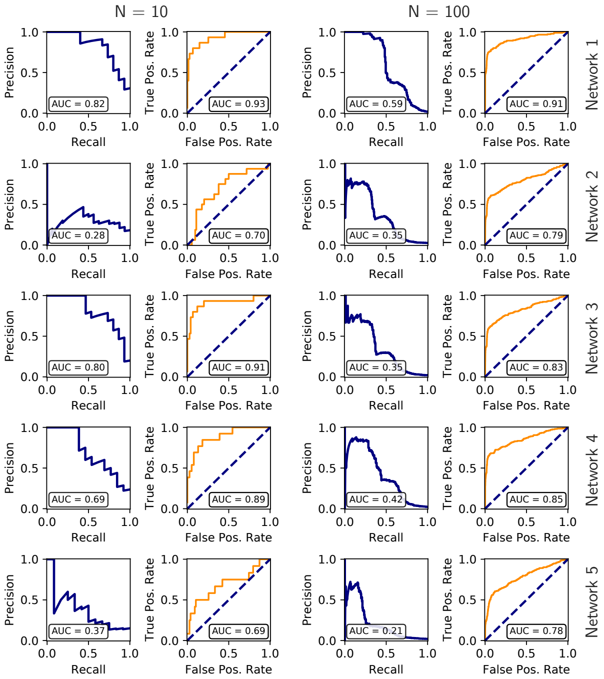

Figure S2: DREAM4 challenge 2: Precision Recall (blue) and ROC (red) curves for the response logic predictions of the five networks.

Both DREAM3 and DREAM4 provided five network challenges per network size. The precision-recall curve and the ROC curve for each network prediction are shown in Figure S1 and Figure S2.

The official DREAMTools package (Cokelaer et al., 2016) was used to compute the final prediction scores that are shown in main text Figure 4A. The scoring is based on p-values for the ROC and precision-recall area under the curves (AUC) that are computed from probability distributions of random network predictions. In the case of the DREAM3, N=100, Yeast3 network the precision-recall AUC of the response logic prediction is larger than that of any of the provided random network predictions. Therefore the determined p-value becomes zero and the overall network score becomes infinity. To obtain a more reasonable score we decided to modify the p-value computation in that case. We identified the largest AUC for which the random probability distribution shows a nonzero probability and simply defined an extended distribution whose probability decreases linearly from this point to zero probability at AUC=1 and computed the p-value based on this approximated distribution. This only concerned one of the five scored networks in the DREAM3, N=100 prediction and the resulting overall score, 139.7, is shown in main text Figure 4A, third panel. Another strategy to cope with the problem is to remove the network altogether and compute the overall score (mean of geometric mean of p-values) only from the remaining four networks. This provides a lower bound for the overall score of 122.9, which is still clearly better than the score of any of the competitors and the naive approach.

S3 Additional information about the analysis of the SW-48 cell line

A reverse phase protein array (RPPA) perturbation experiment was carried out on the SW-48 parental, the SW-48 E545K and SW-48 H1047R PI3K mutant cell lines under full serum conditions. Antibodies were chosen to cover the activity (phosphorylation) of a range of kinases. Cells were perturbed by single small molecule inhibitors, by single growth factors or by a combination of one growth factor and one inhibitor. Measurements were carried out in 8 technical replicates.

In a first step readouts were filtered as follows. First, we removed all readouts whose signal remained within the technical noise level throughout the treatments, as this indicates a malfunctioning antibody. The readout had to be a component or a target of the MAPK or the PI3K pathway. To decrease redundancy, we removed very closely related readouts. This included readouts of functionally related phosphosites on the same kinase, or kinases that showed near-identical qualitative behaviour due to their proximity within the signalling pathway. This left us with 10 different readouts. Concerning the perturbations, we filtered out ineffective, or redundant inhibitors and ligands. This resulted in the data set depicted in Figure S3.

To use this data in the response logic framework we need to convert it to a response pattern with according confidence scores. First, we need to localize the targets of the perturbations. As not all the direct targets of each perturbation were part of the readouts, we replaced those by the ones that were the closest downstream the signalling chain. Concerning HGF stimulation, we observed a strong response by the PDGF receptor. C-Met not being amongst the readouts, we chose the target accordingly. Lacking additional data, it is subject to speculation whether this behaviour could be explained e.g. by receptor co-activation (Xu and Huang, 2010) or impurities of the used ligand. This resulted in the following allocations.

perturbation	EGFRi	MEKi	AKTi	EGF	HGF	IGF
target	EGFR	ERK	AKT	EGFR	PDGFRB	AKT

This makes apparent that the perturbations only have four different targets (PDGFRB, EGFR, ERK, AKT).

Next, we need to determine the response behaviour with respect to perturbations of those four targets. Inhibiting an unstimulated signalling pathway can lead to saturation effects, when additional reduction of kinase activity is not possible. Thus, to faithfully track the inhibition response we decided to investigate inhibition while cells are stimulated, that is to compare ligand + inhibitor to only ligand. To ensure that such a stimulation actually affects the inhibited pathways, we chose EGF stimulation for the EGFR and MEK inhibitors and IGF stimulation for the AKTi inhibitor. The stimulation effects do not suffer from such saturation effects and were thus compared to basal levels (DMSO+PBS) directly. We compiled an overview of the resulting perturbation comparisons in Figure S4.

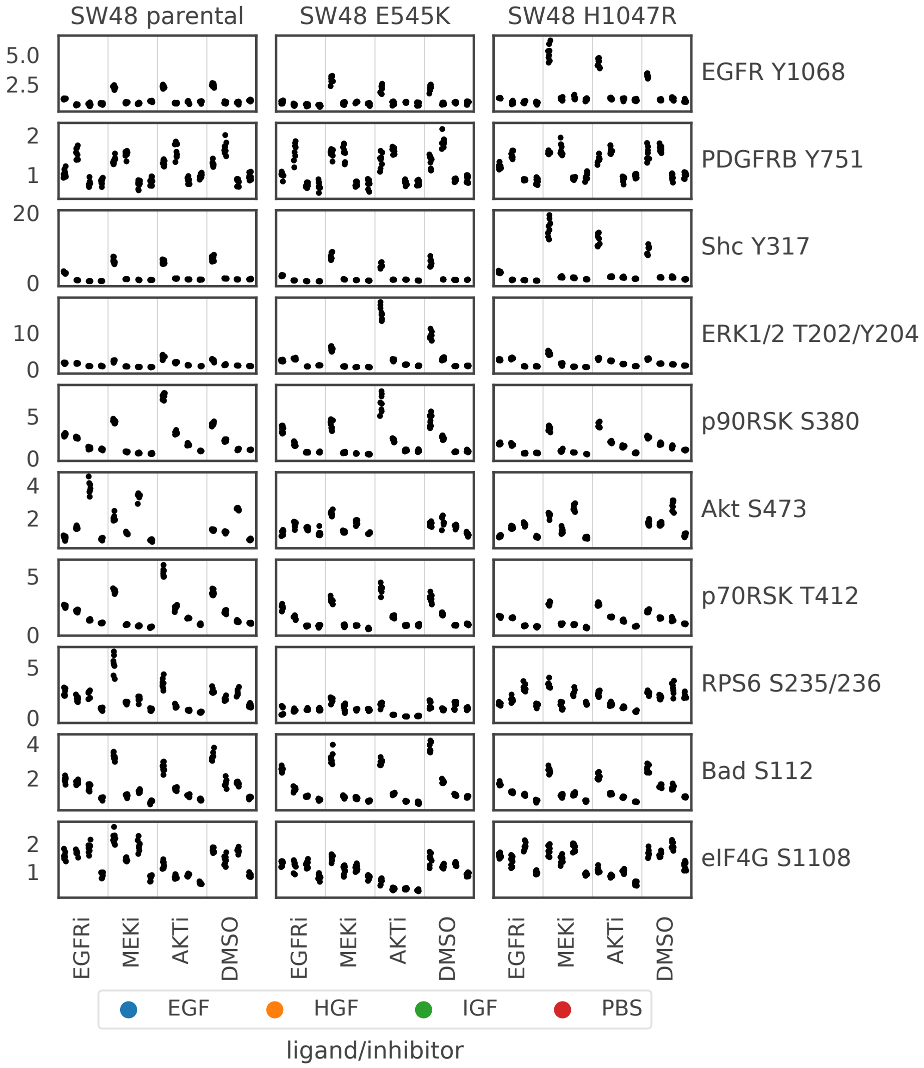

Figure S3: RPPA measurements on SW48 cell lines

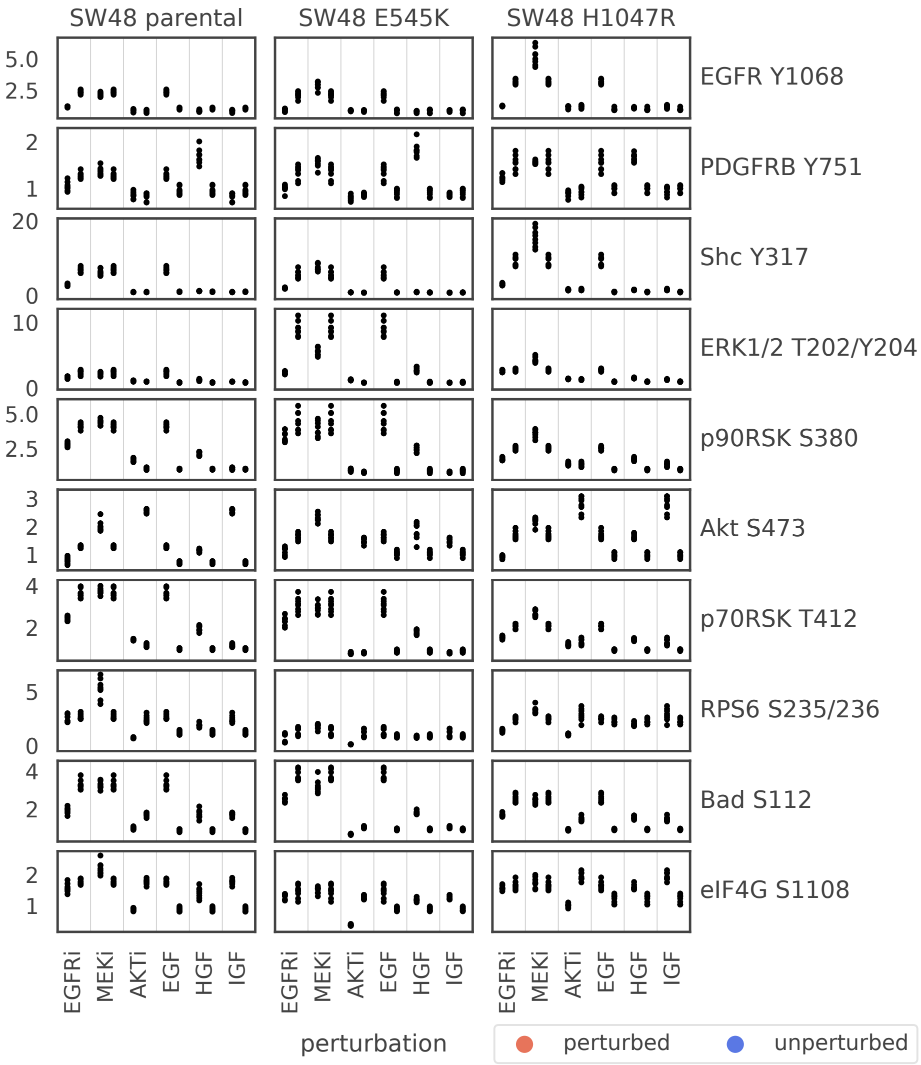

Figure S4: Same data as in Figure S3, but filtered and regrouped to highlight perturbation effects.

To decide whether a perturbation caused a response, we computed p values for each pair of perturbed-unperturbed samples using an unpaired, two-sample t-test. As the data replicates are technical in nature we chose a conservative significance threshold of 0.01. To compute confidence scores we used the same procedure as for the DREAM data, section S2, that is, the confidence scores represent a normalized distance between the p-value and the p-value threshold. Recall, that the six perturbations only have four different targets. To remove redundancy in the response logic sense, we decided to remove the two redundant data points with lower confidence (per cell line and readout). This resulted in the response patterns shown in main text Figure 5A and 6A.

The computation of the area under ROC and precision recall-curves in main text Figure 5A requires to rank the predicted links. However, depending on how the response logic approach is applied, some links can end up with the same score. We therefore, computed the PR AUC and ROC AUC as a mean over 100 AUC values that were generated by randomly reshuffling groups of links with equal score. Repeated computation of these PR AUC and ROC AUC values only varied in the third decimal place.

References

Cokelaer T. et al. (2016) Dreamtools: a python package for scoring collaborative challenges [version 2; referees: 1 approved, 2 approved with reservations]. *F1000Research*, 4.

Videla S. et al. (2015) Learning Boolean logic models of signaling networks with ASP. *Theoretical Computer Science*, 599, 79–101.

Xu A.M. and Huang P.H. (2010) Receptor tyrosine kinase coactivation networks in cancer. *Cancer Research*, 70, 3857–3860.

4 IDENTIFIABILITY AND EXPERIMENTAL DESIGN IN PERTURBATION STUDIES

The simplicity of the response logic offers a crude description of biological mechanics. Often times, it is necessary to obtain a more fine grained picture, which does not only classify network edges as present or absent but associates them with a specific (scalar) interaction strength (Ingram et al. 2006; Guet et al. 2002; Gates et al. 2016). This is useful for example to interpret signalling differences where a specific cellular context solely downregulates a phosphorylation reaction but does not fully eradicate it. A popular method to derive weighted directed graphs from perturbation data is known as Modular Response Analysis (MRA). In short, the method considers perturbation responses as steady states of a system of ordinary differential equations and computes unknown entries of its Jacobian matrix as proxies for interaction strengths. Early works (Bruggeman and Kholodenko 2002; de la Fuente, Brazhnik, et al. 2002; de la Fuente and Mendes 2002; Kholodenko et al. 2002) highlight the connection to metabolic control analysis (Kacser et al. 1973; Heinrich et al. 1974). This explains the naming of MRA, as the molecular description of metabolic control analysis was subsumed into a modular perspective. Modules are thought to involve many cellular components that are interconnected by chemical reactions. However, only few of these components influence other modules and so it is justifiable to describe the system in terms of interaction strengths between modules.

More than one hundred MRA related articles have been published (Bastiaens et al. 2015), including some rediscoveries of the method (Timme 2007; Barzel et al. 2013). While in principle, the approach could be applied to many different biological networks, its main area of application turned out to be the analysis of signalling pathways. This includes the discovery of topological variabilities in the MAPK core network in PC-12 cells which determine the induction of proliferation or differentiation (Santos et al. 2007). Another study elucidated the transcription factor network that is induced by RAS signalling in an ovarian cancer model (Stelniec-Klotz et al. 2012). As described in a previous chapter, MRA allowed to uncover an EGF receptor feedback loop that mediates a cross-talk between the MAPK and the PI3K pathway and conferred resistance to MEK inhibition (Klinger et al. 2013). Similarly, the method described a negative feedback from p70S6K to IRS1 in colorectal cancer cells, which entails resistance to EGFR inhibition (Halasz et al. 2016). MRA was used to analyze signaling networks in breast cancer and could detect rewiring within different cancer types (Speth et al. 2017) and a cross talk between estrogen and retinoic acid receptors (Jimenez-Dominguez et al. 2020). Recently, MRA was also applied to single cell CyTOF data and could elucidate signalling differences between clusters of single cells in mouse intestinal organoids (Brandt et al. 2019).

To keep up with the ever increasing range of applications, the method has been continuously advanced, a development that is extensively reviewed in Santra et al. 2018. For example, it was noted early on that the inferred interaction strengths are sensitive to noise in the response data (Andrec et al. 2005). Amongst other things, this sparked the integration of Monte-Carlo sampling in MRA to obtain confidence intervals for the inferred parameters (Santos et al. 2007). Another practical problem is that it is often difficult to perform comprehensive perturbation experiments. This requires an extension of the original formulations of MRA, which assumed the number of perturbations to equal or exceed the number of nodes in the network. To this end, a Bayesian variable selection algorithm (Santra et al. 2013) or a maximum-likelihood formulation (Klinger et al. 2013) allow to enforce zero interaction strength. Such zero parameters can either be determined from prior network knowledge about missing edges or they are inferred from the response data (Klinger et al. 2018). In addition to incomplete perturbation data, the maximum-likelihood formulation (Klinger et al. 2013) and its successor the stasnet package (Dorel et al. 2018) even allow to infer interaction strengths to and from components that could not be experimentally observed. Another approach formulated MRA as a mixed-integer quadratic programming problem (Bosdriesz et al. 2018), which can simultaneously infer a network from perturbation data on multiple systems. This procedure allows to define which edges are shared or can differ between systems and thus permits to pinpoint, for example, cell line specific differences, even when the data on individual cell lines is sparse.

All of this shows that nearly twenty years after its original formulation, MRA has given rise to a versatile and established reverse-engineering toolbox. Nevertheless, challenges remain. Amongst them are the closely related issues of identifiability and experimental design, which will be the focus of the publication referred to by the end of this chapter. The question is whether a certain interaction strength can be uniquely quantified from a given set of perturbations and a certain amount of prior knowledge about the network. Up to now, this could only be answered within the maximum-likelihood formulation of MRA in terms of a profile likelihood analysis (Raue et al. 2009). However, this iterative procedure needs to frequently evaluate the likelihood function, which can lead to intractable computational complexity, it requires a definition of likelihood thresholds and is thus not guaranteed to be fail-safe, and it is a post hoc approach that only works after experimental data has been collected. Therefore, it could also not address the issue of experimental design, which is the task to (a priori) determine the sequence of perturbations that maximizes the number of identifiable interaction strengths. Intervention experiments are typically laborious and costly, and thus often limited to a handful of perturbations. Thus, improving experimental design is crucial. But even as high-throughput perturbation experiments start to be within reach (Adamson et al. 2016; Datlinger et al. 2017; Dixit et al. 2016; Jaitin et al. 2016; Schraivogel et al. 2020), an efficient reduction of non-identifiability remains important to increase the size of confidently inferred networks. It is therefore surprising how little attention has been paid to the optimization of experimental design, compared to the numerous efforts towards reverse engineering methods. Some of the existing approaches (Birget et al. 2012; Ideker et al. 2000; Lang et al. 2014; Spieth et al. 2004; Steinke et al. 2007; Tegnér et al. 2003) are briefly described in the introduction of Ud-Dean et al. 2016. But most of these either focus on optimizing the inference of boolean networks or rely on time-series data. Up till now, no experimental design strategy for linear response networks such as the ones described by MRA was known.

In this context, our article shows analytically that structural parameter identifiability (Bellman et al. 1970) can be described as a simple maximum flow problem, which only depends on the prior knowledge about the network topology and the targets of the applied perturbations. The maximum flow perspective provides an intuitive understanding of identifiability before data is collected. This makes it possible to explore parameter identifiability for different sequences of perturbations, which can drastically reduce the number of perturbations that is required to determine a set of interaction strengths. This approach is not specific to MRA but holds in principle for all inference methods that are based on linearity assumption similar to that of MRA. Nevertheless, the derivation of the maximum flow formulation yield some insights that could be beneficial for an MRA based network inference. To the best of my knowledge, our article shows for the first time how to reformulate the MRA equations such that the unknown network parameters are determined by a set of linear equation systems, within a general setting of arbitrary prior network knowledge and non-specific perturbations. This formulation not only allowed to derive the maximum-flow conditions but also improves the parameter optimization of MRA models, as shown in Chapter 5.

4.1 PUBLICATION

Identifiability and experimental design in perturbation studies

Torsten Gross, Nils Blüthgen
Bioinformatics, [accepted], 2020.

A Python implementation of the maximum flow approach and the experimental design optimization algorithm can be accessed at:
`github.com/GrossTor/IdentiFlow`

Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

Subject Section

Identifiability and experimental design in perturbation studies

Torsten Gross [1,2,3] **and Nils Blüthgen**[1,2,3]*

[1]Charité - Universitätsmedizin Berlin, Institut für Pathologie, Berlin, Germany, [2]IRI Life Sciences, Humboldt University, Berlin, Germany, and [3]Berlin Institute of Health, Berlin, Germany

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: A common strategy to infer and quantify interactions between components of a biological system is to deduce them from the network's response to targeted perturbations. Such perturbation experiments are often challenging and costly. Therefore, optimising the experimental design is essential to achieve a meaningful characterisation of biological networks. However, it remains difficult to predict which combination of perturbations allows to infer specific interaction strengths in a given network topology. Yet, such a description of identifiability is necessary to select perturbations that maximize the number of inferable parameters.

Results: We show analytically that the identifiability of network parameters can be determined by an intuitive maximum flow problem. Furthermore, we used the theory of matroids to describe identifiability relationships between sets of parameters in order to build identifiable effective network models. Collectively, these results allowed to device strategies for an optimal design of the perturbation experiments. We benchmarked these strategies on a database of human pathways. Remarkably, full network identifiability was achieved with on average less than a third of the perturbations that are needed in a random experimental design. Moreover, we determined perturbation combinations that additionally decreased experimental effort compared to single-target perturbations. In summary, we provide a framework that allows to infer a maximal number of interaction strengths with a minimal number of perturbation experiments.

Availability: IdentiFlow is available at `github.com/GrossTor/IdentiFlow`.

Contact: nils.bluethgen@charite.de

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Rapid technological progress in experimental techniques allows to quantify a multitude of cellular components in ever increasing level of detail. Yet, to gain a mechanistic understanding of the cell requires to map out causal relations between molecular entities. As causality cannot be inferred from observational data alone (Pearl, 2009), a common approach is to observe the system's response to a set of localised perturbations (Sachs *et al.*, 2005) and reconstruct a directed interaction network from such data. Examples for such perturbations are ligands and small molecule inhibitors for the study of signalling pathways, or siRNA knockdowns and CRISPR knockouts of targets in gene regulatory networks.

A recurring idea within the large body of according network inference methods (Marbach *et al.*, 2010) is to conceive the system as ordinary differential equations and describe edges in the directed network by the entries of an inferred Jacobian matrix (Gardner *et al.*, 2003; Bonneau *et al.*, 2006; Tegner *et al.*, 2003; Kholodenko, 2007; Bruggeman *et al.*, 2002; Timme, 2007). Such methods have been successfully applied to describe various types of regulatory networks in different organisms (Ciofani *et al.*, 2012; Arrieta-Ortiz *et al.*, 2015; Lorenz *et al.*, 2009; Klinger *et al.*, 2013; Brandt *et al.*, 2019). They are continuously improved, e.g. to reduce the effect of noise, incorporate heterogeneous data sets, or allow for the analysis of single cell data (Greenfield *et al.*, 2013; Santra *et al.*, 2018; Klinger and Blüthgen, 2018; Santra *et al.*, 2013; Kang *et al.*, 2015; Dorel *et al.*, 2018) and have thus become a standard research tool.

Nevertheless, identifiability (Hengl *et al.*, 2007; Godfrey and DiStefano, 1985) of the inferred network parameters within a specific perturbation setup has not yet been rigorously analysed, even though a limited number of practically feasible perturbations renders many systems underdetermined (De Smet and Marchal, 2010; Meinshausen *et al.*, 2016; Bonneau *et al.*, 2006). Some inference methods do apply different heuristics, such as network sparsity, to justify parameter regularisation (Gardner *et al.*, 2003; Bonneau *et al.*, 2006; Tegner *et al.*, 2003), or numerically analyse identifiability through an exploration of the parameter space using a profile likelihood approach (Raue *et al.*, 2009). Yet, neither approach provides a structural understanding on how parameter identifiability relates to network topology and the targets of the perturbations. However, such structural understanding is required to systematically define identifiable effective network models and to optimize the sequence of applied perturbations. The latter is of particular interest because perturbation experiments are often costly and laborious, which demands to determine the minimal set of perturbations that reveals a maximal number of network parameters. To address these challenges, this work derives analytical results that explain the identifiability of network parameters in terms of simple network properties which allow to optimize the experimental design.

2 Methods

We consider a network of n interacting nodes whose abundances, $\boldsymbol{x}$, evolve in time according to a set of (unknown) differential equations

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{p}). \tag{1}$$

The network can be experimentally manipulated by p different types of perturbations, each represented by one of the p entries of parameter vector $\boldsymbol{p}$. We only consider binary perturbations that can either be switched on or off. Without loss of generality, we define $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{p})$ such that the k-th type of perturbation changes parameter p_k from its unperturbed state $p_k = 0$ to a perturbed state $p_k = 1$.

The main assumption is that after a perturbation the observed system relaxes into stable steady state, $\boldsymbol{\varphi}(\boldsymbol{p})$, of Equation 1. Stability arises when the real parts of the eigenvalue of the n × n Jacobian matrix, $J_{ij}(\boldsymbol{x}, \boldsymbol{p}) = \partial f_i(\boldsymbol{x}, \boldsymbol{p})/\partial x_j$, evaluated at these fixed points, $\boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{p})$, are all negative within the experimentally accessible perturbation space (no bifurcation points). This implies that $J(\boldsymbol{\varphi}(\boldsymbol{p}), \boldsymbol{p})$ is invertible, for which case the implicit function theorem states that $\boldsymbol{\varphi}(\boldsymbol{p})$ is unique and continuously differentiable, and

$$\frac{\partial \varphi_k}{\partial p_l} = -\left[J^{-1} S\right]_{kl}, \tag{2}$$

where n × p Sensitivity matrix entry, $S_{ij} = \partial f_i(\boldsymbol{x}, \boldsymbol{p})/\partial p_j$, quantifies the effect of the j-th perturbation type on node i. Dropping functions' arguments is shorthand for the evaluation at the unperturbed state, $\boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{0})$ and $\boldsymbol{p} = \boldsymbol{0}$.

A linear response approximation

A perturbation experiment consists of q perturbations, each of which involves a single or a combination of perturbation types, represented by binary vector $\boldsymbol{p}$, which forms the columns of the p × q design matrix P. The steady states after each perturbation, $\boldsymbol{\varphi}(\boldsymbol{p})$, are measured and their differences to the unperturbed steady state form the columns of the n × q global response matrix R. Assuming that perturbations are sufficiently mild, the steady state function becomes nearly linear within the relevant parameter domain,

$$\varphi_k(\boldsymbol{p}) - \varphi_k(\boldsymbol{0}) \approx \sum_{l=1}^{\mathrm{p}} \frac{\partial \varphi_k}{\partial p_l}\, p_l. \tag{3}$$

Replacing the partial derivative with the help of Equation 2 and writing the equation for all q perturbations yields

$$R \approx -J^{-1} S\, P. \tag{4}$$

This equation relates the known experimental design matrix, P, and the measured global responses, R, to quantities that we wish to infer: the nodes' interaction strengths, J, and their sensitivity to perturbations, S.

A dynamic system defined by rates $\tilde{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{p}) = W\, \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{p})$, with any full rank n × n matrix W, has the same steady states but different Jacobian and sensitivity matrices, namely $W\, J$ and $W\, S$, as the original system, defined by Equation 1. It is thus impossible to uniquely infer J or S from observations of the global response alone, and prior knowledge in matrices J and S is required to further constrain the problem. In the following, we assume that prior knowledge exists about the network topology, i.e. about zero entries in J, as they correspond to non-existent edges. Likewise, we assume that the targets of the different types of perturbations are known, which implies known zero entries in S for non-targeted nodes. In line with prior studies (Kholodenko, 2007), we also fix the diagonal of the Jacobian matrix

$$J_{ii} = -1.$$

Thus, for the i-th row of J we can define index lists $\bar{\boldsymbol{\mu}}_i$ and $\hat{\boldsymbol{\mu}}_i$ to identify its known and unknown entries. The first indicates missing edges or the self loop and the second edges going into node i. These lists have $|\bar{\boldsymbol{\mu}}_i|$ and $|\hat{\boldsymbol{\mu}}_i|$ entries, respectively, with

$$|\bar{\boldsymbol{\mu}}_i| + |\hat{\boldsymbol{\mu}}_i| = \mathrm{n}. \tag{5}$$

Analogously, for the i-th row of S we define index lists $\bar{\boldsymbol{\nu}}_i$ and $\hat{\boldsymbol{\nu}}_i$, with

$$|\bar{\boldsymbol{\nu}}_i| + |\hat{\boldsymbol{\nu}}_i| = \mathrm{p}, \tag{6}$$

to report its unknown and known entries. These describe the perturbations that do not target or respectively target node i.

We show in Supplementary Material S1 that Equation 4 can be repartitioned to obtain a system of linear equations for each row in J and S, exclusively in the

$$\mathrm{u}_i = |\hat{\boldsymbol{\mu}}_i| + |\hat{\boldsymbol{\nu}}_i|$$

unknown parameters, which we collect in vector $\boldsymbol{x}_i$. Thus, there is a $\mathrm{u}_i \times \mathrm{d}_i$ matrix V_i, such that

$$\boldsymbol{x}_i = V_i \boldsymbol{w} + \tilde{\boldsymbol{x}}_i,\ \forall \boldsymbol{w} \in \mathbb{R}^{\mathrm{d}_i}, \tag{7}$$

where $\tilde{\boldsymbol{x}}_i$ is some specific solution to the equation system. We further show in Supplementary Material S1 that V_i is a basis of the kernel of

$$\Psi_i = \begin{bmatrix} \hat{S}_i \hat{J}_i^{-1} & I_{|\hat{\boldsymbol{\nu}}_i|} \\ \bar{S}_i \hat{J}_i^{-1} & 0_{|\bar{\boldsymbol{\nu}}_i|,|\hat{\boldsymbol{\nu}}_i|} \end{bmatrix}, \tag{8}$$

where $I_{|\hat{\boldsymbol{\nu}}_i|}$ and $0_{|\bar{\boldsymbol{\nu}}_i|,|\hat{\boldsymbol{\nu}}_i|}$ are the identity and zero matrix of annotated dimensionality. The n × $|\hat{\boldsymbol{\mu}}_i|$ matrix $\hat{J}_i^{-1}$ consists of the columns of $(J^{-1})^T$ that are selected by indices in $\boldsymbol{\mu}_i$. Finally, $|\bar{\boldsymbol{\nu}}_i|$ × n matrix $\bar{S}^i$ and $\hat{\boldsymbol{\nu}}_i$ × n matrix $\hat{S}^i$ shall be formed by taking rows of S^T according to indices in $\bar{\boldsymbol{\nu}}_i$ and $\hat{\boldsymbol{\nu}}_i$. These matrix partitionings are demonstrated for a toy example in Supplementary Figure S1. Furthermore, in Supplementary Material S1 we derive the following expression for the solution space dimensionality

$$\mathrm{d}_i = |\hat{\boldsymbol{\mu}}_i| - \mathrm{rank}\left(\bar{S}_i \hat{J}_i^{-1}\right).$$

Identifiability conditions

The system is underdetermined when $\mathrm{d}_i > 0$. But independent of d_i, a parameter is identifiable if the solution space is orthogonal to its according axis direction. This idea can be expressed as algebraic identifiability conditions. Accordingly, we show in Supplementary Material S1 that the unknown interaction strength $J_{i\hat{\mu}_{ij}}$ is identifiable if and only if

$$1 + \mathrm{rank}(\bar{S}_i \hat{J}^{-1}_{i\setminus j}) = \mathrm{rank}(\bar{S}_i \hat{J}^{-1}_i), \qquad (9)$$

where $\hat{J}^{-1}_{i\setminus j}$ is matrix $\hat{J}^{-1}_i$ with the j-th column removed. Furthermore, the unknown sensitivity $S_{i\bar{\nu}_{ij}}$ is identifiable if and only if

$$\mathrm{rank}\left(\begin{bmatrix}\bar{S}_i \\ \hat{\boldsymbol{S}}^j_i\end{bmatrix} \hat{J}^{-1}_i\right) = \mathrm{rank}(\bar{S}_i \hat{J}^{-1}_i), \qquad (10)$$

where $\hat{\boldsymbol{S}}^j_i$ denotes the j-th row of matrix $\hat{S}_i$. However, the ranks depend on the unknown network parameters themselves and can thus not be directly computed. Yet, we can show how a reasonable assumption makes this possible and allows to express the identifiability conditions as an intuitive maximum flow problem.

First, we rewrite the identity $J^{-1}J = I_\mathrm{n}$ as

$$[J^{-1}]_{kl} = \sum_{m \neq l} [J^{-1}]_{km}\, [J]_{ml} - \delta_{kl},$$

with δ_{kl} being the Kronecker delta (recall that $J_{ll} = -1$). We can view this equation as a recurrence relation and repeatedly replace the $[J^{-1}]_{km}$ terms in the sum. The sum contains non-vanishing terms for each edge that leaves node l. Therefore, each replacement leads to the next downstream node, so that eventually one arrives at

$$[J^{-1}]_{kl} = l \rightarrowtail k \;\; [J^{-1}]_{kk}, \text{ with}$$

$$l \rightarrowtail k = \sum_{\boldsymbol{\omega} \,\in\, \Omega_{l\to k}} \; \prod_{m=1}^{|\boldsymbol{\omega}|-1} [J]_{\omega_{m+1}\, \omega_m},$$

where the set $\Omega_{l\to k}$ contains elements, $\boldsymbol{\omega}$, for every path from node l to node k, each of which lists the nodes along that path. Strictly speaking, these elements are walks rather than paths because some nodes will appear multiple times if loops exist between l and k. In fact, with loops, $\Omega_{l\to k}$ contains an infinite number of walks of unbounded lengths. But as the real part of all eigenvalues of J are assumed negative, the associated products of interaction strengths converges to zero with increasing walk length.

To simplify our notation, we want to expand the network by considering perturbations $\bar{\boldsymbol{\nu}}_i$ as additional nodes, each with edges that are directed towards that perturbation's targets. Furthermore, letting the interaction strength associated with these new edges be given by the appropriate entries in S we can rewrite the matrix product

$$\left[\bar{S}_i \hat{J}^{-1}_i\right]_{kl} = \bar{\nu}_{ik} \rightarrowtail \hat{\mu}_{il} \;\; [J^{-1}]_{\hat{\mu}_{il}\, \hat{\mu}_{il}}$$

where $\hat{\mu}_{il}$ and $\bar{\nu}_{il}$ denote the l-th entry in $\hat{\boldsymbol{\mu}}_i$ and $\bar{\boldsymbol{\nu}}_i$, respectively. As every finite-dimensional matrix has a rank decomposition, we can further write

$$\bar{S}_i \hat{J}^{-1}_i = \Upsilon_i\, Y_i, \qquad (11)$$

where $|\bar{\boldsymbol{\nu}}_i| \times \mathrm{rank}(\bar{S}_i \hat{J}^{-1}_i)$ matrix Υ_i and $\mathrm{rank}(\bar{S}_i \hat{J}^{-1}_i) \times |\hat{\boldsymbol{\mu}}_i|$ matrix Y_i have full rank. Finding such a decomposition therefore reveals the rank of $\bar{S}_i \hat{J}^{-1}_i$. To this end, we propose

$$[\Upsilon_i]_{kn} = \bar{\nu}_{ik} \rightarrowtail y_{in}, \text{ and } [Y_i]_{nl} = y_{in} \rightarrowtail \hat{\mu}_{il} \;\; [J^{-1}]_{\hat{\mu}_{il}\, \hat{\mu}_{il}},$$

where y_{in} denotes the n-th component of a certain list of nodes $\boldsymbol{y}_i$. In order for Equation 11 to hold, it must be possible to split each path from any perturbation $\bar{\nu}_{il}$ to any node $\hat{\mu}_{il}$ into a section that leads from the perturbation to a node in $\boldsymbol{y}_i$ and a subsequent section that leads from this node to $\hat{\mu}_{il}$. For an extended graph that includes an additional source node, with outgoing edges to each perturbation in $\bar{\boldsymbol{\nu}}_i$, and an additional sink node, with incoming edges from all nodes in $\hat{\boldsymbol{\mu}}_i$ (see Figure 1B), $\boldsymbol{y}_i$ thus constitutes a vertex cut whose removal disconnects the graph and separates the source and the sink node into distinct connected components. Next, we want to show that if $\boldsymbol{y}_i$ is a minimum vertex cut, the rank of $\bar{S}_i \hat{J}^{-1}_i$ equals the size of $\boldsymbol{y}_i$. Because Equation 11 is a rank decomposition this is equivalent to showing that the according matrices Υ_i and Y_i have full rank. To do so, we apply Menger's theorem (Menger, 1927), which states that the minimal size of $\boldsymbol{y}_i$ equals the maximum number of vertex-disjoint paths from the source to the sink node. This also implies that each of these vertex-disjoint paths goes through a different node of the vertex cut $\boldsymbol{y}_i$. Recall that entries in Υ_i constitute sums over paths from perturbation to vertex cut nodes, so that we could write

$$\Upsilon_i = \bar{\Upsilon}_i + \hat{\Upsilon}_i,$$

where $\bar{\Upsilon}_i$ only contains the vertex-disjoint paths and $\hat{\Upsilon}_i$ the sums over the remaining paths. As each of these vertex disjoint paths ends in a different vertex cut node, any column in $\bar{\Upsilon}_i$ can contain no more than a single non-zero entry. Furthermore, as a consequence of Menger's theorem there are exactly $|\boldsymbol{y}_i|$ non-zero columns. Because these paths are indeed vertex disjoint also no row in $\bar{\Upsilon}_i$ has more than a single non-zero entry. Thus, the non-zero columns are independent, showing that $\bar{\Upsilon}_i$ has full rank. We further assume that adding $\hat{\Upsilon}_i$ does not reduce rank, which also gives Υ_i full rank. In the context of biological networks there are two different scenarios that could lead to a violation of this non-cancellation assumption. The first is that network parameters are perfectly tuned to lie inside a specific algebraic variety (a manifold in parameter space) such that certain columns (or rows) of Υ_i become linearly dependent or zero. This would for example be the case if, for a given vertex disjoint path, there also is an alternative path whose associated product of interaction strengths has the same magnitude as that of the vertex disjoint path but opposite sign, making their sum vanish. However, we consider it implausible for biological networks to be fine-tuned to such a degree that they could achieve such perfect self-compensation of perturbations, and rule out this possibility. A more realistic scenario is that network parameters are zero and thereby lead to zero columns or rows in Υ_i or Y_i, which make these matrices rank deficient. In practice, such zero-parameters can occur, for example, if a perturbation is not effective on (one of) its target(s), or if robustness effects (Fritsche-Guenther *et al.*, 2011) obstruct the propagation of the perturbation signal at a certain link. But essentially, this means that our prior knowledge about the network included practically non-existing links or perturbation targets. If the network topology and perturbation targets are correctly stated and take these effects into consideration, there will be no zero-parameters and therefore the non-cancellation assumption holds. We explore the consequences of incomplete or flawed prior knowledge in Supplementary Material S5.

Having shown Υ_i to be of full rank, the same line of reasoning will demonstrate a full rank for matrix Y_i as well, which implies that indeed

$$\mathrm{rank}(\bar{S}_i \hat{J}^{-1}_i) = |\boldsymbol{y}_i|, \qquad (12)$$

where $\boldsymbol{y}_i$ is a minimum vertex cut between source and sink node. This equation has the crucial benefit that $|\boldsymbol{y}_i|$ does not depend on any unknown parameters and can be computed as the maximum flow from source to sink node with all nodes having unit capacity (Ahuja *et al.*, 1993), as detailed in Figure 1B. A flow is defined as a mapping from a network edge to a positive real number that is smaller than the edge's capacity. Additionally, the sum of flows entering a node must equal the sum of the flows exiting a node, except for the source and the sink nodes. The maximum flow problem is to attribute (permissible) flow values to all edges, such that the sum of flows leaving the source (which is equal to the sum of flows entering the sink) is maximal. In our case however, we did not define edge but node capacities, meaning that the sum of flows passing through any node must not exceed one. Yet, we can express such unit node capacities as unit edge capacities in an extended flow network. It is defined by replacing every node by an *in*- and an *out*-node, where all incoming edges target the *in*-node, all outgoing edges start from the *out*-node, and the *in*-node has an edge to the *out*-node.

This maximum flow problem allows to express the algebraic identifiability conditions 9 and 10 in terms of network properties, providing an intuitive relationship between network topology, perturbation targets and identifiability. Specifically, $J_{i\hat{\mu}_{ij}}$ is identifiable if and only if the removal of the edge from node $\hat{\mu}_{ij}$ to the sink node reduces the maximum flow of the network, see Figure 1C, and $S_{i\hat{\nu}_{ij}}$ is identifiable if the maximum flow does not increase when an additional edges connects the source node with perturbation node $\hat{\nu}_{ij}$, see Figure 1D. In Supplementary Material S5, we simulate a perturbation experiment to numerically verify these findings.

Identifiability relationships

Often, network inference is an underdetermined problem (De Smet and Marchal, 2010; Gross *et al.*, 2019). Thus, to achieve identifiable effective network models, certain parameters have to be set to constant values, such that the remaining parameters become uniquely determinable. This requires an understanding of the identifiability relationships between parameters, i.e. we need to know which parameter becomes identifiable when other parameters are fixed. Supplementary Equation 18 formally relates these relationships to the ranks of certain linear subspaces of the range of V_i^T as defined in Equation 7. It shows that for each network node there is a set of parameters amongst which identifiability relationships can exist. Such a set contains those interaction strengths that quantify the edges, which target the associated node, and the associated node's sensitivities to perturbations. Furthermore, we show in Supplementary Material S2 that the identifiability relationships of such parameter groups can be described as a matroid (Whitney, 1935). Matroids can be defined in terms of their circuits. Here, a circuit is a set of parameters with the property that any of its parameters becomes identifiable after fixing all of the others. Therefore, circuits describe all minimal parameter subsets that could be fixed to obtain an identifiable network.

We enumerated the set of circuits with an incremental polynomial-time algorithm (Boros *et al.*, 2003). This algorithm requires an independence oracle that indicates linear dependence of subsets of columns of V_i^T. Supplementary Material S2 shows that we can construct such an oracle by considering linear dependence within the dual matroid, which amounts to determining

$$\operatorname{rank}\left(\begin{bmatrix}\tilde{\mathcal{P}}_2^T\,\hat{S}_i\\ \bar{S}_i\end{bmatrix}\hat{J}_i^{-1}\,\mathcal{P}_1\right).$$

Matrices $\mathcal{P}_2$ and $\mathcal{P}_1$ are truncated identity matrices defined in Supplementary Equations 19 and 20. Yet, the crucial point of this expression is that it has the same form as the left hand side of Equation 12. We can therefore conveniently determine it by solving a simple maximum-flow problem.

Supplementary Material S2 shows how to transform the circuits into cyclic flats. These provide a more convenient representation of the identifiability relationships, which we clarify at an example in the next section. Finally, certain scenarios constrain the choice of fixable parameters, for example when quantifying multiple isogenic cell lines (Bosdriesz *et al.*, 2018). Supplementary Material S2 describes a greedy algorithm that takes such preferences into consideration.

Experimental design strategies

We assume that we are given a set of p perturbations, each of which targets a different subset of nodes. In the following, we will define different experimental design strategies that suggest different sequences in which these perturbations should be applied. By means of our understanding of identifiability, we can determine ξ_i, the number of identifiable edges after having performed the first i perturbations in such a sequence. Our goal is to find a strategy for which this number of identifiable edges increases fastest. Thus, as a measure of a sequence's optimality, we can define an identifiability area under the curve

$$\frac{1}{\mathrm{p}}\sum_{i=1}^{\mathrm{p}}\frac{\xi_i}{\xi}, \tag{13}$$

where ξ is the number of edges in the network. For any network and perturbation sequence, this score ranges between zero and one.

Consider a directed graph with 2^{p} nodes, each of which represents a different subset of the p perturbations. Each edge in this graph shall connect such a perturbation subset to one of its proper supersets that contains one additional perturbation. Then, we can view perturbation sequences as paths on this graph, starting from the empty perturbation subset. We shall define design strategies as rules that describe which perturbation(s) should be performed next, given the perturbations that have already been applied. These rules thus represent the edges on the graph and will therefore determine which perturbation sequences are associated with the given strategy. To enumerate these perturbation sequences, we implemented a depth-first search. The details of our algorithm are described in Supplementary Material S3. Here, we provide an overview over the different implemented strategies.

An obvious approach to design an optimal strategy is to simply consider all remaining perturbations as next possible perturbations. This *exhaustive* strategy is therefore associated with the entire set of possible perturbation sequences. We are therefore guaranteed to find those sequences amongst them that maximize the identifiability area under the curve. On the downside, this strategy quickly becomes computationally intractable when the set of perturbations becomes large (we analyse computational complexity of the different strategies in Supplementary Material S3). Therefore, we also implemented strategies with more restrictive rules. A *random* strategy will, at each step, randomly choose one of the remaining perturbations. This will thus result in a single random perturbation sequence. A *naive* strategy is based on the notion that perturbations should be more informative if they cause a response at a possibly large number of nodes. Thus, this strategy considers the perturbed nodes for each of the remaining perturbations and computes the number of network nodes to which these are connected to by a path. It then selects those perturbations as possible next perturbations, which maximize this number. In contrast, the *single-target* strategy makes use of the maximum-flow approach, as it selects perturbations that will, first, maximize the number of identifiable edges, and second, minimize the overall dimensionality of the solution space, $\sum_i \mathrm{d}_i$. Finally, the *multi-target* strategy is similar to the single-target approach, except that it not only considers single but combinations of perturbations. That is, we allow any perturbation combination to be considered as a single perturbation experiment, which will then perturb all targets of the combined perturbations. Clearly, this can open an excessively large search space, when the number of possible perturbations is big. We therefore implemented a tractable, step-wise procedure to build up perturbation combinations, which is described in Supplementary Material S3.

As these strategies allow for multiple perturbations to be considered next in the sequence, they are associated not only to a single but to many sequences (which we enumerate by the depth-first search). Amongst them, we can then choose the ones that maximize the optimality score defined

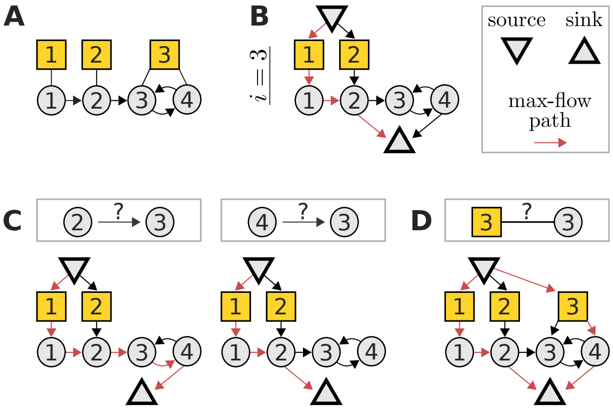

Fig. 1. A maximum flow problem determines the identifiability of interaction strengths and perturbation sensitivities when reconstructing a network from perturbation data. (A) Example network with three perturbations (yellow squares) to illustrate the algorithm. (B) The corresponding flow network to determine the identifiability of the edges into node 3 and the sensitivity of node 3 to perturbations. A path carrying the maximal flow of one is denoted in red (note that it is not unique). (C) The interaction strength between a given node and node 3 is identifiable if and only if the maximum flow is reduced after removing that node's edge to the sink node. In this example, there are alternative max-flow paths that re-establish a unit-flow after removal of the according edges. Thus, the respective interaction strengths are non-identifiable. (D) Similarly, the sensitivity of node 3 to perturbation 3 is identifiable, if and only if the depicted extension of the flow network does not increase the maximum flow. In this example, the maximum flow is increased by one, again revealing non-identifiability. Note that such flow representations provide an intuitive understanding on how alterations in the network or perturbation setting affect identifiability. For example, it is obvious that if the toy model would not contain an edge from node 3 to 4, the edge from 2 to 3 would become identifiable.

in Equation 13. However, for large systems the number of these strategy-associated sequences can become too large to be completely enumerated. We therefore also implemented an approach to randomly sample from this sequence set, as follows. For a given strategy, instead of considering the entire set of possible next perturbations, we only randomly pick a single one. The strategy will then be associated with a single sequence. Every time we repeat this procedure, we randomly sample from the (original) strategy-associated sequences.

3 Results

Identifiability and identifiability relationships

Perturbation experiments are frequently used to infer and quantify interactions in biological networks. But whether a given network edge can indeed be uniquely quantified from experimentally observed perturbation responses depends on the specific targets of the perturbations and the topology of the network. In order to build interpretable network models and guide experimental design, we need to elucidate this identifiability status of the network parameters. Here, we view a biological system as a weighted directed network, and assume that perturbations are sufficiently mild to cause a linear steady state response. This allows to relate the interaction strengths between nodes (i.e. the entries in the Jacobian matrix J) and the sensitivity to perturbations (i.e. the entries in the Sensitivity matrix S) to the measured responses (Equation 4), an approach that is widely known as Modular Response Analysis (Kholodenko, 2007). We derived analytical identifiability conditions (Equations 9 and 10) that describe whether this relation allows to uniquely determine the network parameters for the given network topology and the experimental setting. However, these conditions can not be directly evaluated, as they depend on the (unknown) network parameters themselves. But instead, they can be reformulated as intuitive maximum flow problems, if one disregards singular conditions of self-cancelling perturbations.

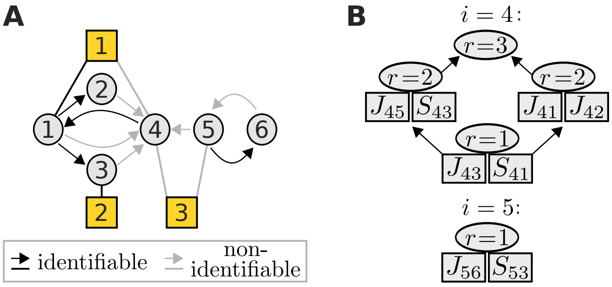

Fig. 2. (A) An example network with three perturbations (yellow squares), where nodes 4 and 5 are associated with non-identifiable parameters (grey). (B) Their identifiability relationships are represented by the lattices of cyclic flats of rank r. Each cyclic flat consists of the annotated elements in addition to elements from its preceding cyclic flats. All parameters of a cyclic flat with rank r become identifiable if at least r independent flat parameters are fixed.

The derivation and details are given in the Methods section but briefly, to determine the identifiability of either the interaction strength from node j to node i, or the sensitivity of node i to perturbation p, the following flow network is considered: The original network is extended by (i) adding a node for each perturbation that does not target node i and connecting it to the respective perturbation's target(s), (ii) adding a "source" node that connects to all those perturbation nodes, and (iii) having all nodes that target node i connect to an additional "sink" node, see Figure 1B. Furthermore, all nodes (except source and sink) and all edges have a flow capacity of one. To reveal identifiability, we need to determine the network's maximum flow from source to sink. This is a classic problem in computer science, which we solve using the Edmonds-Karp algorithm (Dinic, 1970; Edmonds and Karp, 1972) as implemented in the Networkx package (Hagberg *et al.*, 2008). Then, the interaction strength from node j to node i is identifiable if and only if the removal of the edge from node j to the sink node reduces the maximum flow, see Figure 1C. Similarly, node i's sensitivity to perturbation p is identifiable if and only if the maximum flow does not increase after linking the source to an additional node that is in turn connected to all targets of perturbation p, see Figure 1D.

Often, experimental settings do not allow determining all unknown parameters (De Smet and Marchal, 2010; Gross *et al.*, 2019). Nevertheless, they constrain the solution space such that after fixing one or multiple parameters, others become identifiable. We found that such identifiability relationships can be described by matroids, which are combinatorial structures that generalize the notion of linear dependence (see Methods). This is demonstrated for an example perturbation experiment on the network displayed in Figure 2A.

Each node is associated with a set of parameters amongst which identifiability relationships can exist. Such a set contains those interaction strengths, which quantify the edges that target the associated node, and that node's sensitivities to perturbations. Here, nodes 4 and 5 are associated with sets of non-identifiable parameters. For example for node 5, these are J_{56} and S_{53}. We represent the matroid for such a parameter set as a hierarchy (lattice) of cyclic flats, as show in Figure 2B. A cyclic flat is a set of parameters with an associated rank r. It has the property that all of its parameters become identifiable, if amongst them at least r independent parameters are fixed. Parameters are independent if none of them becomes identifiable after fixing the others. For node 5, parameters J_{56} and S_{53} only form a single cyclic flat with $r = 1$, and thus fixing either one parameter makes the other identifiable. The identifiability relationships among the six parameters associated with node 4 are more complex. For example,

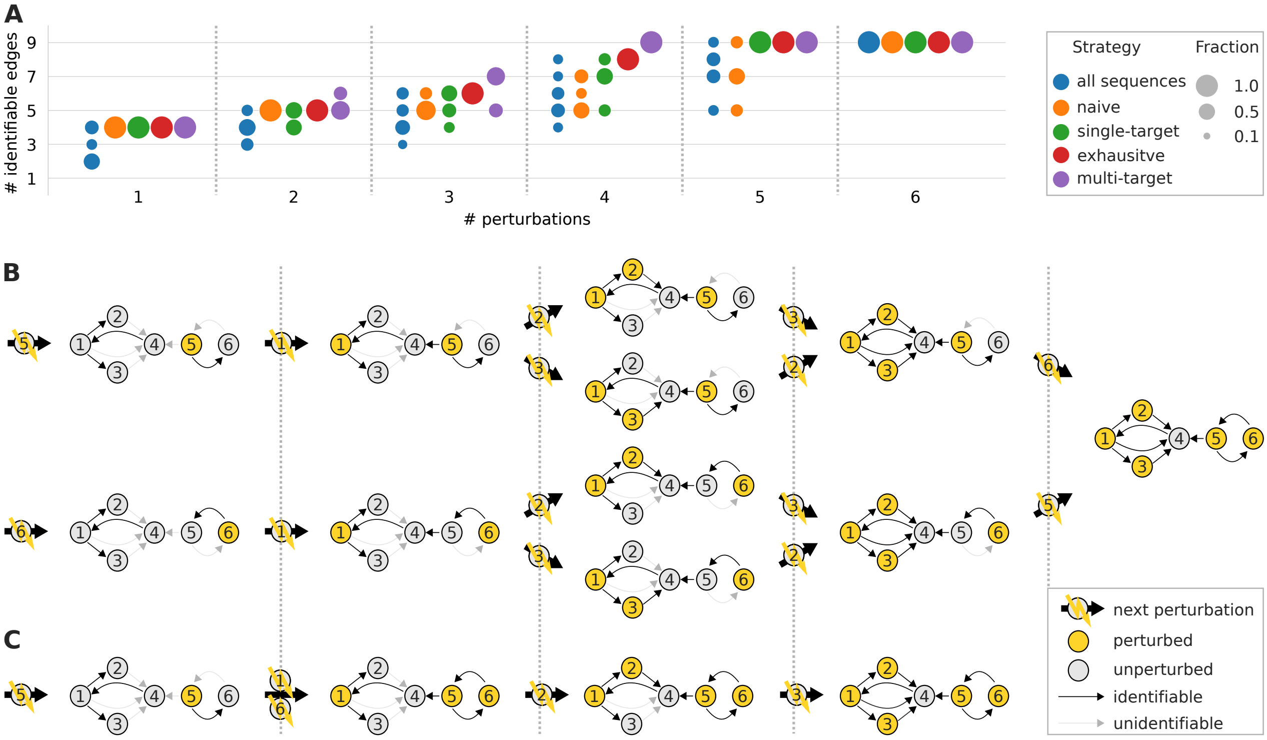

Fig. 3. (A) The same network topology as in Figure 2 was subjected to a set of perturbations that target each node individually. Shown are distributions of numbers of identifiable edges for different experimental design strategies and an increasing number of perturbations. (B) All perturbation sequences associated with the single-target strategy and (C) one sequence associated with the multi-target sequence.

J_{43} and S_{41} form a cyclic flat with $r = 1$ and thus fixing one, fixes the other. Yet together with J_{45} and S_{43}, they form a cyclic flat with $r = 2$, thus fixing e.g. S_{41} and S_{43} will allow unique determination of J_{43} and J_{45}. In contrast, fixing J_{43} and S_{41} does not render any other parameter identifiable because they are not independent. This illustrates how the matroid description allows to generate effective models, i.e. models where a minimum number of parameters has to be set to fixed values to allow for a unique estimation of all other parameters. Importantly, the lattice of cyclic flats can be derived without specifying unknown parameters by solving a sequence of maximum flow problems (see Methods).

Collectively, our results provide a concise framework to algorithmically determine identifiability of network parameters and to construct identifiable effective networks when the experimental setting does not suffice to uniquely determine the original network structure.

Experimental design

Next, we applied our identifiability analysis to optimize experimental design, i.e. to minimize the number of perturbation experiments that is required to uniquely determine a network's interaction strengths. For this, we designed the following strategies to determine an optimal sequence from a set of available perturbations: The *exhaustive* strategy considers all possible sequences and selects the best performing amongst them. As this approach entails a prohibitive computational effort for larger networks, we also designed approaches that select perturbation sequences in a step-wise manner: The *single-target* strategy chooses next perturbations such that they increases the number of identifiable edges most. The *multi-target* strategy is similar to the single-target strategy except that it not only considers a single but any combination of perturbations. In contrast, the *naive* strategy does not use our identifiability analysis. Rather, it chooses perturbations first that cause a response at the largest possible number of nodes (see Methods for details).

We first scrutinised the proposed experimental design strategies on the example network shown in Figure 2. We defined six different types of perturbations, each of which targets a (different) single node, or any combination of such for the multi-target strategy. Figure 3A shows how the number of identifiable edges increases with the number of performed perturbations for each strategy. A single strategy is associated with multiple sequences, as described in Methods. Accordingly, Figure 3A shows the performance distribution over all these sequences. In practice, we would only select the best performing sequence amongst them. Nevertheless, the depicted distributions are informative because for larger networks we can no longer enumerate all but only a (random) subset of conforming sequences, as described in Methods.

When comparing the methods, we found that each strategy's average performance is higher than the average performance of all possible sequences.

Moreover, the "naive" strategy that did not use our framework mostly required all six perturbations to fully identify all parameters, whereas the single-target and exhaustive strategies only needed five, and the multi-target strategy only four perturbations. Figures 3B and C display all perturbation sequences associated with the single-target strategy, and one sequence associated with the multi-target strategy respectively, and illustrate which network edge becomes identifiable at which step in the sequence.

To systematically analyse if and how our approach improves experimental design, we benchmarked the different strategies on all 267 nontrivial human KEGG (Kanehisa *et al.*, 2019) pathways, ranging from 5 to 120 nodes (see Supplementary Material S4 for details). Again, we assumed that perturbations can target (all) single nodes. For each

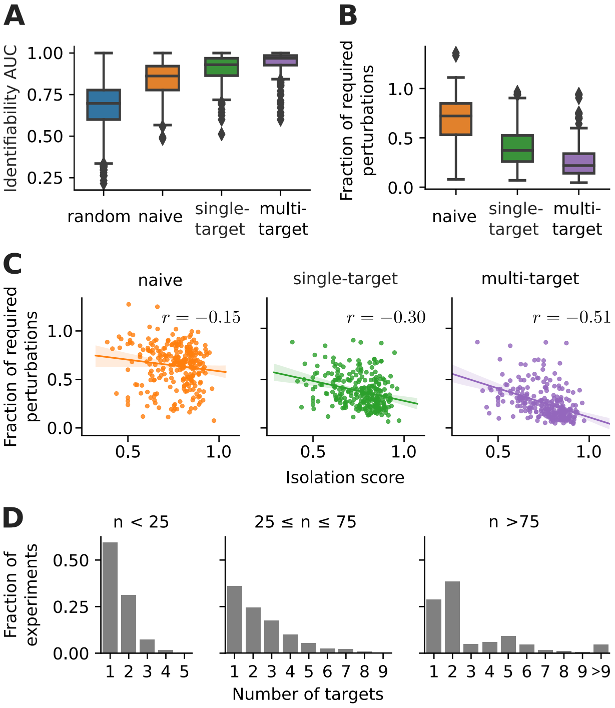

Fig. 4. Performance of different experimental design strategies on 267 human KEGG pathways. (A) Identifiability AUC, defined as area under the number of identified nodes vs. number of perturbation curve, see Eqn. 13, (B) For each network and strategy, the average number of perturbations required for full identifiability is shown relative to the average number required for a random strategy. (C) The fraction of required perturbations correlated against the isolation score of a network (Eqn. 14), r : Spearman's rank correlation. (D) The fraction of multi-target perturbations with a specific number of targets to all multi-target perturbations (experiments) in KEGG networks of the annotated size range.

network, we sampled 10 conforming sequences per strategy (as described in Methods) and compared against the performance of 10 randomly chosen sequences. As a performance measure of each sequence, we considered the number of identifiable edges as a function of the number of perturbations and computed a normalised area under the curve, as defined in Equation 13. Figure 4A shows the result of this benchmark, and confirms the trend already observed for the example in Figure 3A: Compared to choosing perturbations randomly, the naive strategy improved identifiability. Performance was further increased when we applied our single-target strategy, yet the multi-target strategy clearly performed best. An exhaustive enumeration of all sequences is not feasible for all KEGG networks. However, we found for a subset of small networks that there is no performance difference between the exhaustive and the single-target strategy, as shown in Supplementary Figure S5A.

Furthermore, we determined the number of perturbations that is required for full network identifiability (shown in Supplementary Figure S6) and computed the fraction between a given strategy and the random sequences, see Figure 4B. We found that the average number of required perturbations can be reduced to less than one third or even less than a quarter, when using a single-target or multi-target strategy, respectively. To verify that the performance of the multi-target strategy is not only due to its much larger set of perturbation choices, we also measured the performance of random sequences of perturbation combinations, shown in Supplementary Figure S7. While such a multi-random strategy increases the performance compared to the random strategy, it is still inferior to the single- and multi-target approach.

We next investigated which network properties led to a performance increase using our strategies. Intuitively, perturbations might be more informative if their response propagates to large parts of the network. We therefore hypothesised that a careful experimental design is particularly beneficial when networks contain many isolated nodes with little connection to the rest of the network because, in contrast to a random choice, a good strategy could then avoid perturbing such non-informative targets. On the contrary, the sequence of perturbations is irrelevant in the extreme case of a fully connected network. To investigate this hypothesis we defined a network's isolation score as

$$1-\sum_{ij}^{\mathrm{n}}\frac{\pi_{ij}}{\mathrm{n}\,(\mathrm{n}-1)},\ \text{with}\ \pi_{ij}=\begin{cases}1, & \exists\ \text{path}\ i\rightarrow j\\ 0, & \nexists\ \text{path}\ i\rightarrow j\end{cases}. \qquad (14)$$

Figure 4C shows that indeed the isolation score negatively correlates with the previously defined fraction of perturbations required for full network identifiability. Furthermore, we also observed a positive correlation between isolation score and the difference in the identifiability AUC between non-random and random strategies, as shown in Supplementary Figure S5B. This suggests that indeed our experimental design strategies increase their performance with increasing network isolation.

When response signals converge at a node, the individual contribution from each incoming edge can not be distinguished. Thus, the advantage of a multi-target perturbation to potentially track signal propagation through larger parts of the network is counter-balanced if it leads to more convergent signal propagation. This is prevented when the (combined) perturbations target isolated parts of the network. Therefore the strongest correlation in Figure 4C is found for the multi-target strategy because with higher isolation score we can expect to find more such isolated subnetworks. And indeed, Figure 4D shows that the multi-target strategy typically suggest combinations of multiple single target perturbations, especially in larger networks.

In summary, we have developed an algorithmic approach to determine structural identifiability for a given network. This approach allows to derive experimental design strategies that drastically reduce experimental effort in perturbation studies. In particular, the multi-target strategy proved most efficient. Potentially, this finding has practical relevance because in many experimental contexts it easy to combine perturbations, e.g. by multiplexed CRISPR knockouts (Minkenberg *et al.*, 2017).

4 Discussion

We have shown analytically that the identifiability of parameters in linear perturbation networks can be described as a simple maximum flow problem (summarised in Figure 1). All that is required to perform this analysis is an accurate specification of the (directed) network topology and the targets of the perturbations. This includes the consideration of, e.g., robustness, or perturbation off-target effects that are specific to the experimental setup and that can influence the wiring of the network. A failure to do so might break the non-cancellation assumption (discussed in Methods) and thereby lead to flawed identifiability statements, as shown in Supplementary Material S5.

Our intuitive description of identifiability not only explains how to achieve fully identifiable effective network models (Figure 2), but also enables us to optimize the design of perturbation experiments (Figure 3). As a test case, we examined all human KEGG pathways and found that our method typically allows to cut down the number of perturbations required for full identifiability to one fourth compared to choosing perturbation targets randomly (Figure 4). We provide a `python` implementation of our results `github.com/GrossTor/IdentiFlow`, which allows to determine identifiability, perform matroid computations that display identifiability relationships between parameters, and optimize experimental design. The package relies on standard maximum flow algorithms from the Networkx package (Hagberg *et al.*, 2008).

Technically, it would be possible to cope with non-identifiable parameters numerically, as was done previously (Gardner *et al.*, 2003; Bonneau *et al.*, 2006; Tegner *et al.*, 2003; Dorel *et al.*, 2018). Yet, these procedures tend to be computationally expensive, might depend on heuristic thresholds and are thus not guaranteed to work in general, which makes them inadequate tools for experimental design. Even more importantly, the benefit of the maximum flow perspective is that identifiability can be intuitively understood in relation to the network topology and the targets of the perturbations. This means that instead of requiring numerical procedures on a case by case basis, our approach uses intuitively understandable flow networks to link identifiability to the network topology and perturbation set-up. This provides a comprehensive overview on which edges become identifiable under which perturbations. For one, this permits a straightforward optimisation of the experimental design, as shown before. But even in a situation where the set of perturbations is *a priori* fixed because of experimental constraints, our approach concisely reveals which network topologies are in principle amenable to a meaningful analysis. Thereby, it maps out the range of answerable biological questions. For example for the toy network depicted in Figure 1A, we could ask whether node 2 or node 4 activates node 3 more strongly, which would be an important question if the activity of node 3 is associated with a certain phenotype that we try to influence by inhibiting either node 2 or node 4. Figure 1C showed that this is not answerable because both edges are non-identifiable. However, the maximum flow approach makes it obvious that the question could indeed be addressed if there was another edge from node 1 to node 3 (as this creates an additional edge from node 1 to the source node in the flow net in Figure 1B that increases the maximum flow to two).

Our analysis describes the identifiability of parameters in a network model whose steady state changes linearly with the magnitude of a perturbation. But clearly, biological systems generally break linearity assumptions in varying degrees, which bears asking how useful our description is. In principle, we could expand the steady state function Equation 3 to higher orders and attempt to also infer nonlinear rate terms, which are products of different node and perturbation magnitudes. However such products no longer have any meaningful network interpretation, as they cannot be reasonably assigned to any edge. Therefore we argue that the linearity assumption is essential to derive a useful effective network description, if we choose to interpret the biological systems in terms of ordinary differential equations. On the downside, the biological meaning of interaction strengths becomes increasingly obscure the more the system violates the linearity assumption (Prabakaran *et al.*, 2014). Even though our method could still correctly reveal which linear network parameters are uniquely determined by the data, it is questionable how useful this information is, if this value no longer holds a biological meaning. In particular, this could diminish the benefit of a multi-target experimental design strategy, as combined perturbations might push the system into saturation. Hence, even though our maximum-flow approach is independent of the actual measured response data, a strongly non-linear behaviour of the underlying biological system can render it irrelevant. We therefore need to carefully consider when a linear network model is an adequate description.

Importantly, our approach described in this article solely addresses the problem of structural identifiability. In contrast, problems with so-called practical identifiability arise from insufficient quality of experimental data (Raue *et al.*, 2011). Thus, even when the structural identifiability condition for a specific parameter holds, it does not necessarily mean that its value can be reliably estimated. The maximum flow approach can be used before experiments are conducted, and thus is agnostic to information about noise that could potentially render a structurally identifiable parameter practically non-identifiable. Similarly, it cannot cope with missing measurements of a node's steady state response, which is a common challenge in novel single cell perturbation studies (Jaitin *et al.*, 2016; Datlinger *et al.*, 2017). Yet, in these scenarios our approach can provide an experimental strategy to construct a structurally identifiable model. And subsequently, established methods can be used efficiently to handle practical non-identifiability (Raue *et al.*, 2009; Dorel *et al.*, 2018).

Funding

This work has been supported by the Deutsche Forschungsgemeinschaft, RTG2424, CompCancer

References

Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, N.J.

Arrieta-Ortiz, M. L., Hafemeister, C., Bate, A. R., Chu, T., Greenfield, A., Shuster, B., Barry, S. N., Gallitto, M., Liu, B., Kacmarczyk, T., Santoriello, F., Chen, J., Rodrigues, C. D. A., Sato, T., Rudner, D. Z., Driks, A., Bonneau, R., and Eichenberger, P. (2015). An experimentally supported model of the Bacillus subtilis global transcriptional regulatory network. *Mol. Syst. Biol.*, **11**(11), 839.

Bonneau, R., Reiss, D. J., Shannon, P., Facciotti, M., Hood, L., Baliga, N. S., and Thorsson, V. (2006). The Inferelator: An algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome Biol.*, **7**(5), R36.

Boros, E., Elbassioni, K., Gurvich, V., and Khachiyan, L. (2003). Algorithms for Enumerating Circuits in Matroids. In T. Ibaraki, N. Katoh, and H. Ono, editors, *Algorithms and Computation*, Lecture Notes in Computer Science, pages 485–494, Berlin, Heidelberg. Springer.

Bosdriesz, E., Prahallad, A., Klinger, B., Sieber, A., Bosma, A., Bernards, R., Blüthgen, N., and Wessels, L. F. A. (2018). Comparative Network Reconstruction using mixed integer programming. *Bioinformatics*, **34**(17), i997–i1004.

Brandt, R., Sell, T., Lüthen, M., Uhlitz, F., Klinger, B., Riemer, P., Giesecke-Thiel, C., Schulze, S., El-Shimy, I. A., Kunkel, D., Fauler, B., Mielke, T., Mages, N., Herrmann, B. G., Sers, C., Blüthgen, N., and Morkel, M. (2019). Cell type-dependent differential activation of ERK by oncogenic KRAS in colon cancer and intestinal epithelium. *Nat. Commun.*, **10**(1), 2919.

Bruggeman, F. J., Westerhoff, H. V., Hoek, J. B., and Kholodenko, B. N. (2002). Modular response analysis of cellular regulatory networks. *J. Theor. Biol.*, **218**(4), 507–520.

Ciofani, M., Madar, A., Galan, C., Sellars, M., Mace, K., Pauli, F., Agarwal, A., Huang, W., Parkhurst, C. N., Muratet, M., Newberry, K. M., Meadows, S., Greenfield, A., Yang, Y., Jain, P., Kirigin, F. K., Birchmeier, C., Wagner, E. F., Murphy, K. M., Myers, R. M., Bonneau, R., and Littman, D. R. (2012). A validated regulatory network for Th17 cell specification. *Cell*, **151**(2), 289–303.

Datlinger, P., Rendeiro, A. F., Schmidl, C., Krausgruber, T., Traxler, P., Klughammer, J., Schuster, L. C., Kuchler, A., Alpar, D., and Bock, C. (2017). Pooled CRISPR screening with single-cell transcriptome readout. *Nat. Methods*, **14**(3), 297–301.

De Smet, R. and Marchal, K. (2010). Advantages and limitations of current network inference methods. *Nat. Rev. Microbiol.*, **8**(10), 717–729.

Dinic, E. A. (1970). Algorithm for solution of a problem of maximum flow in networks with power estimation. *Sov. Math Dokl.*, **11**, 1277–1280.

Dorel, M., Klinger, B., Gross, T., Sieber, A., Prahallad, A., Bosdriesz, E., Wessels, L. F. A., and Blüthgen, N. (2018). Modelling signalling networks from perturbation data. *Bioinformatics*, **34**(23), 4079–4086.

Edmonds, J. and Karp, R. M. (1972). Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems. *J. ACM*, **19**(2), 248–264.

Even, S. and Even, G. (2012). *Graph Algorithms*. Cambridge University Press, Cambridge, NY, 2nd ed edition.

Fritsche-Guenther, R., Witzel, F., Sieber, A., Herr, R., Schmidt, N., Braun, S., Brummer, T., Sers, C., and Blüthgen, N. (2011). Strong negative feedback from Erk to Raf confers robustness to MAPK signalling. *Molecular Systems Biology*, **7**(1), 489.

Gardner, T. S., di Bernardo, D., Lorenz, D., and Collins, J. J. (2003). Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, **301**(5629), 102–105.

Godfrey, K. and DiStefano, J. (1985). Identifiability of Model Parameter. *IFAC Proceedings Volumes*, **18**(5), 89–114.

Greenfield, A., Hafemeister, C., and Bonneau, R. (2013). Robust data-driven incorporation of prior knowledge into the inference of dynamic regulatory networks. *Bioinformatics*, **29**(8), 1060–1067.

Gross, T., Wongchenko, M. J., Yan, Y., and Blüthgen, N. (2019). Robust network inference using response logic. *Bioinformatics*, **35**(14), i634–i642.

Hagberg, A., Swart, P., and S Chult, D. (2008). Exploring network structure, dynamics, and function using networkx. Technical Report LA-UR-08-05495; LA-UR-08-5495, Los Alamos National Lab. (LANL), Los Alamos, NM (United States).

Hengl, S., Kreutz, C., Timmer, J., and Maiwald, T. (2007). Data-based identifiability analysis of non-linear dynamical models. *Bioinformatics*, **23**(19), 2612–2618.

Jaitin, D. A., Weiner, A., Yofe, I., Lara-Astiaso, D., Keren-Shaul, H., David, E., Salame, T. M., Tanay, A., van Oudenaarden, A., and Amit, I. (2016). Dissecting Immune Circuits by Linking CRISPR-Pooled Screens with Single-Cell RNA-Seq. *Cell*, **167**(7), 1883–1896.e15.

Kanehisa, M., Sato, Y., Furumichi, M., Morishima, K., and Tanabe, M. (2019). New approach for understanding genome variations in KEGG. *Nucleic Acids Res.*, **47**(D1), D590–D595.

Kang, T., Moore, R., Li, Y., Sontag, E., and Bleris, L. (2015). Discriminating direct and indirect connectivities in biological networks. *Proc. Natl. Acad. Sci. U.S.A.*, **112**(41), 12893–12898.

Kholodenko, B. N. (2007). Untangling the signalling wires. *Nat. Cell Biol.*, **9**(3), 247–249.

Klinger, B. and Blüthgen, N. (2018). Reverse engineering gene regulatory networks by modular response analysis - a benchmark. *Essays Biochem.*, **62**(4), 535–547.

Klinger, B., Sieber, A., Fritsche-Guenther, R., Witzel, F., Berry, L., Schumacher, D., Yan, Y., Durek, P., Merchant, M., Schäfer, R., Sers, C., and Blüthgen, N. (2013). Network quantification of EGFR signaling unveils potential for targeted combination therapy. *Mol. Syst. Biol.*, **9**, 673.

Lorenz, D. R., Cantor, C. R., and Collins, J. J. (2009). A network biology approach to aging in yeast. *Proc. Natl. Acad. Sci. U.S.A.*, **106**(4), 1145–1150.

Marbach, D., Prill, R. J., Schaffter, T., Mattiussi, C., Floreano, D., and Stolovitzky, G. (2010). Revealing strengths and weaknesses of methods for gene network inference. *Proc. Natl. Acad. Sci. U.S.A.*, **107**(14), 6286–6291.

Meinshausen, N., Hauser, A., Mooij, J. M., Peters, J., Versteeg, P., and Bühlmann, P. (2016). Methods for causal inference from gene perturbation experiments and validation. *Proc. Natl. Acad. Sci. U.S.A.*, **113**(27), 7361–7368.

Menger, K. (1927). Zur allgemeinen Kurventheorie. *Fundam. Math.*, **10**(1), 96–115.

Minkenberg, B., Wheatley, M., and Yang, Y. (2017). Chapter Seven - CRISPR/Cas9-Enabled Multiplex Genome Editing and Its Application. In D. P. Weeks and B. Yang, editors, *Progress in Molecular Biology and Translational Science*, volume 149 of *Gene Editing in Plants*, pages 111–132. Academic Press.

Pearl, J. (2009). *Causality: Models, Reasoning and Inference*. Cambridge University Press, New York, NY, USA, 2nd edition.

Prabakaran, S., Gunawardena, J., and Sontag, E. (2014). Paradoxical results in perturbation-based signaling network reconstruction. *Biophys. J.*, **106**(12), 2720–2728.

Raue, A., Kreutz, C., Maiwald, T., Bachmann, J., Schilling, M., Klingmüller, U., and Timmer, J. (2009). Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. *Bioinformatics*, **25**(15), 1923–1929.

Raue, A., Kreutz, C., Maiwald, T., Klingmuller, U., and Timmer, J. (2011). Addressing parameter identifiability by model-based experimentation. *IET Syst Biol*, **5**(2), 120–130.

Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A., and Nolan, G. P. (2005). Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, **308**(5721), 523–529.

Santra, T., Kolch, W., and Kholodenko, B. N. (2013). Integrating Bayesian variable selection with Modular Response Analysis to infer biochemical network topology. *BMC Syst Biol*, **7**, 57.

Santra, T., Rukhlenko, O., Zhernovkov, V., and Kholodenko, B. N. (2018). Reconstructing static and dynamic models of signaling pathways using Modular Response Analysis. *Current Opinion in Systems Biology*, **9**, 11–21.

Tegner, J., Yeung, M. K. S., Hasty, J., and Collins, J. J. (2003). Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling. *Proc. Natl. Acad. Sci. U.S.A.*, **100**(10), 5944–5949.

Timme, M. (2007). Revealing network connectivity from response dynamics. *Phys. Rev. Lett.*, **98**(22), 224101.

Whitney, H. (1935). On the Abstract Properties of Linear Dependence. *American Journal of Mathematics*, **57**(3), 509.

SUPPLEMENT: Identifiability and experimental design in perturbation studies

Torsten Gross[1,2,3] and Nils Blüthgen[1,2,3]

[1]Charité - Universitätsmedizin Berlin, Institut für Pathologie, Berlin, Germany,
[2]IRI Life Sciences, Humboldt University, Berlin, Germany
[3]Berlin Institute of Health, Berlin, Germany

Sections S1 and S2 constitute an extended version of the sections on identifiability and identifiability relationships in the Methods part of the main text. They include additional information and examples, and explicit derivations that were abbreviated in the main text.

Section S4 provides additional details and data for the optimization of experimental design in KEGG pathways.

S1 Identifiability

We want to consider a network of n interacting nodes whose abundances or magnitudes, $\boldsymbol{x}$, evolve in time according to a set of (unknown) differential equations

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{p}). \tag{1}$$

We assume that we can experimentally manipulate the system with p different types of perturbations, each of which is represented by one of the p entries of parameter vector $\boldsymbol{p}$. We shall only consider binary perturbations that can either be fully switched on or off. To keep notation simple and without loss of generality, we thus define $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{p})$, such that the k-th type of perturbation changes parameter p_k from its unperturbed state $p_k = 0$ to a perturbed state $p_k = 1$.

One of the main assumptions about the observed system is that its temporal dynamics eventually relaxes into different constant states depending on the performed perturbation. These states are thought to represent stable fixed points, $\boldsymbol{\varphi}(\boldsymbol{p})$, of Equation 1, where stability arises because the real parts of the eigenvalue of the n × n Jacobian matrix, $J_{ij}(\boldsymbol{x}, \boldsymbol{p}) = \partial f_i(\boldsymbol{x}, \boldsymbol{p})/\partial x_j$, evaluated at these fixed points, $\boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{p})$, are all negative within the experimentally accessible perturbation space (no bifurcation points). This implies that $J(\boldsymbol{\varphi}(\boldsymbol{p}), \boldsymbol{p})$ is invertible, for which case the implicit function theorem states that $\boldsymbol{\varphi}(\boldsymbol{p})$ is unique and continuously differentiable, and

$$\frac{\partial \varphi_k}{\partial p_l} = -\left[J^{-1} S\right]_{kl}, \tag{2}$$

where n × p sensitivity matrix entry, $S_{ij} = \partial f_i(\boldsymbol{x}, \boldsymbol{p})/\partial p_j$, quantifies the effect of the j-th perturbation type on node i. Dropping functions' arguments is shorthand for the evaluation at the unperturbed state, $\boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{0})$ and $\boldsymbol{p} = \boldsymbol{0}$.

A linear response approximation

A perturbation experiment consists of q perturbations, each of which involves a single or a combination of perturbation types, represented by binary vector $\boldsymbol{p}$. These vectors shall form the p × q design matrix P. After each perturbation the system is allowed sufficient time until the newly established steady states, $\boldsymbol{\varphi}(\boldsymbol{p})$, can be measured. Let their differences to the unperturbed steady state form the columns of the n × q global response matrix R. The central approximation is to assume that perturbations are sufficiently mild, such that the steady state function becomes nearly linear within the relevant parameter domain,

$$\varphi_k(\boldsymbol{p}) - \varphi_k(\boldsymbol{0}) \approx \sum_{l=1}^{\mathrm{p}} \frac{\partial \varphi_k}{\partial p_l} p_l. \tag{3}$$

Replacing the partial derivative with the help of Equation 2 and writing the equation for all q perturbations yields

$$R \approx -J^{-1} S\, P. \tag{4}$$

Note that this equation holds exactly and independent of perturbation strength for a linear system

$$\dot{\boldsymbol{x}} = J\boldsymbol{x} + S\boldsymbol{p},$$

which can be seen by considering its steady state

$$\boldsymbol{x}^{\boldsymbol{0}} = J^{-1} S\, \boldsymbol{p}.$$

The crux of Equation 4 is that it relates the known experimental design matrix, P, and the measured global responses, R, to quantities that we wish to infer, namely the nodes' interaction strengths, J, and their sensitivity to perturbations, S. Thus, as a next step we shall rewrite the equation to disentangle the known and the unknown entries.

A dynamic system defined by rates $\tilde{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{p}) = W\, \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{p})$, with any full rank n × n matrix W, has the same steady states but different Jacobian and sensitivity matrices, namely $W\,J$ and $W\,S$, as the original

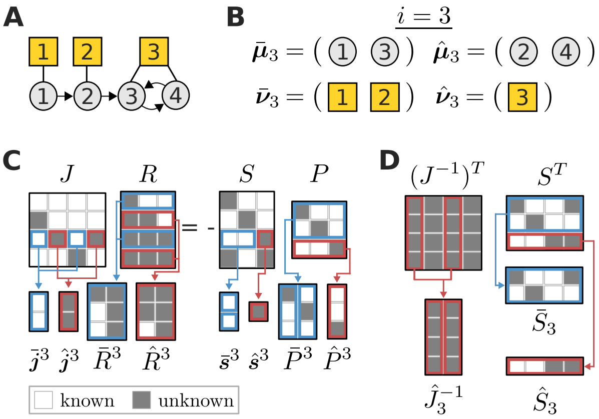
A
1
2
3
1
2
3
4
B
$i = 3$
$\bar{\boldsymbol{\mu}}_3 = (\,1\ \ 3\,)$
$\hat{\boldsymbol{\mu}}_3 = (\,2\ \ 4\,)$
$\bar{\boldsymbol{\nu}}_3 = (\,1\ \ 2\,)$
$\hat{\boldsymbol{\nu}}_3 = (\,3\,)$
C
J
R
S
P
= -
$\bar{\boldsymbol{j}}^3$ $\hat{\boldsymbol{j}}^3$ $\bar{R}^3$ $\hat{R}^3$
$\bar{\boldsymbol{s}}^3$ $\hat{\boldsymbol{s}}^3$ $\bar{P}^3$ $\hat{P}^3$
known
unknown
D
$(J^{-1})^T$
S^T
$\bar{S}_3$
$\hat{J}_3^{-1}$
$\hat{S}_3$

Figure S1: Three perturbations (yellow squares) are performed on a toy network (**A**). Network topology and perturbation targets determine the index lists from Equation 5 and Equation 6. Here they are depicted for $i = 3$ (**B**). A graphical representation of Equation 4 demonstrates the definition of various matrix partitions (**C** and **D**).

system, defined by Equation 1. It is thus impossible to uniquely infer J or S from observations of the global response alone. However, some entries in matrices J and S might be known a priori and thus further constrain the problem. This is the case, when e.g. certain reactions rates are known. Typically however, such values are hard to come by. Rather, we assume prior knowledge about the network topology. That is, we know the zero entries in J as they correspond to non-existent edges. Likewise, we assume to know the targets of the different types of perturbations which imply zero entries in S-rows corresponding to perturbations that that are known to not directly affect the network node associated with that row. In line with prior studies, we fix the diagonal of the Jacobian matrix

$$J_{ii} = -1.$$

Thus, for the i-th row of J we can define index lists $\bar{\boldsymbol{\mu}}_i$ and $\hat{\boldsymbol{\mu}}_i$, with

$$|\bar{\boldsymbol{\mu}}_i| + |\hat{\boldsymbol{\mu}}_i| = \mathrm{n}, \tag{5}$$

identifying its known and unknown entries. The first correspond to missing edges or the self loop and the second to edges going into node i. Analogously, for the i-th row of S we define index lists $\bar{\boldsymbol{\nu}}_i$ and $\hat{\boldsymbol{\nu}}_i$, with

$$|\bar{\boldsymbol{\nu}}_i| + |\hat{\boldsymbol{\nu}}_i| = \mathrm{p}, \tag{6}$$

to report its unknown and known entries. These describe the perturbations that do not target or respectively target node i, see Figure S1B.

To see whether prior knowledge about J and S entries could render other entries determinable, we first rewrite Equation 4 as n linear equation systems

$$R^T \boldsymbol{j}_i = -P^T \boldsymbol{s}_i, \quad i = 1, 2, \ldots \mathrm{n}, \tag{7}$$

one for each column in J^T and S^T, denoted as $\boldsymbol{j}_i$ and $\boldsymbol{s}_i$. Then, we collect the known and unknown $\boldsymbol{j}_i$-entries into vectors $\bar{\boldsymbol{j}}_i$ and $\hat{\boldsymbol{j}}_i$ following the indexing by $\bar{\boldsymbol{\mu}}_i$ and $\hat{\boldsymbol{\mu}}_i$. In the same manner, $\boldsymbol{s}^i$ is split into the known vector $\bar{\boldsymbol{s}}^i$ and unknown vector $\hat{\boldsymbol{s}}^i$ according to $\bar{\boldsymbol{\nu}}_i$ and $\hat{\boldsymbol{\nu}}_i$. To rewrite Equation 7 as a linear system of the unknown variables, we first partition its terms into known and unknown parts

$$R^T \boldsymbol{j}_i = \bar{R}_i\, \bar{\boldsymbol{j}}_i + \hat{R}_i\, \hat{\boldsymbol{j}}_i \quad \text{and} \quad P^T \boldsymbol{s}_i = \bar{P}_i\, \bar{\boldsymbol{s}}_i + \hat{P}_i\, \hat{\boldsymbol{s}}_i,$$

where $\mathrm{q} \times |\bar{\boldsymbol{\mu}}_i|$ matrix $\bar{R}_i$ and $\mathrm{q} \times |\hat{\boldsymbol{\mu}}_i|$ matrix $\hat{R}_i$ consist of those columns of R^T that are selected by $\bar{\boldsymbol{\mu}}_i$ and $\hat{\boldsymbol{\mu}}_i$, respectively. Analogously, $\mathrm{q} \times |\bar{\boldsymbol{\nu}}_i|$ matrix $\bar{P}_i$ and $\mathrm{q} \times |\hat{\boldsymbol{\nu}}_i|$ matrix $\hat{P}_i$ are formed from the columns of P^T selected by $\bar{\boldsymbol{\nu}}_i$ and $\hat{\boldsymbol{\nu}}_i$, respectively. These vector and matrix partitions are illustrated in Figure S1C. Introducing abbreviations

$$\boldsymbol{x}_i = \begin{bmatrix} \hat{\boldsymbol{j}}_i \\ \hat{\boldsymbol{s}}_i \end{bmatrix} \quad \text{and} \quad \boldsymbol{k}_i = \begin{bmatrix} \bar{R}_i & \bar{P}_i \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{j}}_i \\ \bar{\boldsymbol{s}}_i \end{bmatrix},$$

an equivalent reformulation of Equation 7 reads

$$\begin{bmatrix} \hat{R}_i & \hat{P}_i \end{bmatrix} \boldsymbol{x}_i = -\boldsymbol{k}_i, \quad i = 1, 2, \ldots \mathrm{n}. \tag{8}$$

The point of such algebraic acrobatics is that Equation 8 represents systems of linear equations, each in the

$$\mathrm{u}_i = |\hat{\boldsymbol{\mu}}_i| + |\hat{\boldsymbol{\nu}}_i|$$

unknown parameters $\boldsymbol{x}_i$, compared to Equation 7 in which the solution vector comprised unknown and known components. It thus allows to study the identifiability of $\boldsymbol{x}_i$.

Identifiability conditions

Clearly, Equation 8 is underdetermined if

$$\mathrm{d}_i = \mathrm{u}_i - \mathrm{rank}(\begin{bmatrix} \hat{R}_i & \hat{P}_i \end{bmatrix}) > 0.$$

To analyse this solution space dimensionality, let $\mathrm{n} \times |\hat{\boldsymbol{\mu}}_i|$ matrix $\hat{J}_i^{-1}$ consist of the columns of $\left(J^{-1}\right)^T$ that are selected by $\boldsymbol{\mu}_i$. Similarly, $|\bar{\boldsymbol{\nu}}_i| \times \mathrm{n}$ matrix $\bar{S}^i$ and $\hat{\boldsymbol{\nu}}_i \times \mathrm{n}$ matrix $\hat{S}^i$ shall be formed by taking rows of S^T according to indices in $\bar{\boldsymbol{\nu}}_i$ and $\hat{\boldsymbol{\nu}}_i$, as shown in Figure S1D. Also, we have I_i denote the i-dimensional identity matrix and $0_{i,j}$ the $i \times j$ zero-matrix. We use these definitions and Equation 4 to write

$$\hat{R}_i = -P^T S^T \hat{J}_i^{-1} \quad \text{and} \quad P^T S^T = \begin{bmatrix} \hat{P}_i & \bar{P}_i \end{bmatrix} \begin{bmatrix} \hat{S}_i \\ \bar{S}_i \end{bmatrix},$$

and arrive at

$$\begin{bmatrix} \hat{R}_i & \hat{P}_i \end{bmatrix} = -\begin{bmatrix} \hat{P}_i & \bar{P}_i \end{bmatrix} \Psi_i, \quad \text{with} \tag{9}$$

$$\Psi_i = \begin{bmatrix} \hat{S}_i \hat{J}_i^{-1} & I_{|\hat{\boldsymbol{\nu}}_i|} \\ \bar{S}_i \hat{J}_i^{-1} & 0_{|\bar{\boldsymbol{\nu}}_i|,|\hat{\boldsymbol{\nu}}_i|} \end{bmatrix}. \tag{10}$$

Note that $\begin{bmatrix} \hat{P}_i & \bar{P}_i \end{bmatrix}$ is nothing but a rearrangement of the columns of P^T and therefore

$$\mathrm{rank}(\begin{bmatrix} \hat{P}_i & \bar{P}_i \end{bmatrix}) = \mathrm{rank}(P) = \mathrm{p}.$$

Claiming P to have rank p assumes that throughout the experiment every type of perturbation was applied in a non-trivial combination. This is not a limiting constraint as it is for example satisfied for a perturbation scheme in which each type of perturbation is applied once individually, which is the case for the examples discussed here.

From $\begin{bmatrix}\hat{P}_i & \bar{P}_i\end{bmatrix}$ having full (column) rank follows that

$$\begin{aligned}&\operatorname{rank}([\hat{R}_i \quad \hat{P}_i]) = \operatorname{rank}(\Psi_i)\\ &= \operatorname{rank}([\hat{S}_i\hat{J}_i^{-1} \quad I_{|\hat{\boldsymbol{\nu}}_i|}]) + \operatorname{rank}([\bar{S}_i\hat{J}_i^{-1} \quad 0_{|\bar{\boldsymbol{\nu}}_i|,|\hat{\boldsymbol{\nu}}_i|}])\\ &= |\hat{\boldsymbol{\nu}}_i| + \operatorname{rank}\left(\bar{S}_i\hat{J}_i^{-1}\right),\end{aligned}$$

so that the solution subspace has dimensionality

$$\mathrm{d}_i = |\hat{\boldsymbol{\mu}}_i| - \operatorname{rank}\left(\bar{S}_i\hat{J}_i^{-1}\right). \qquad (11)$$

From the dimensionality of matrix product $\bar{S}_i\hat{J}_i^{-1}$ we can conclude that $d_i \geq \max(0,\, \mathrm{n} - |\bar{\boldsymbol{\mu}}_i| - |\bar{\boldsymbol{\nu}}_i|)$. Thus, to fully determine $\boldsymbol{x}_i$ we need to provide at least as many elements of prior knowledge as there are nodes in the network, which agrees with our earlier observation that we can transform the rate equations with an arbitrary $\mathrm{n} \times \mathrm{n}$ matrix without altering the steady states.

If indeed $\mathrm{d}_i > 0$, there is a $\mathrm{u}_i \times \mathrm{d}_i$ matrix V_i whose columns form a basis of the kernel of $\begin{bmatrix}\hat{R}_i & \hat{P}_i\end{bmatrix}$, so that, given $\tilde{\boldsymbol{x}}_i$, a specific solution to Equation 8, any

$$\boldsymbol{x}_i = V_i\boldsymbol{w} + \tilde{\boldsymbol{x}}_i,\ \forall \boldsymbol{w} \in \mathbb{R}^{\mathrm{d}_i} \qquad (12)$$

is also a solution of Equation 8. But even though the equation system is then underdetermined, not all network parameters are necessarily unidentifiable. Rather,

$$\begin{aligned}[\boldsymbol{x}_i]_j \text{ identifiable } &\iff \boldsymbol{e}_j^T\, V_i = 0\\ &\iff \exists\, \boldsymbol{w} \in \mathbb{R}^{\mathrm{q}} : \begin{bmatrix}\hat{R}_i & \hat{P}_i\end{bmatrix}^T \boldsymbol{w} = \boldsymbol{e}_j,\end{aligned} \qquad (13)$$

where $\boldsymbol{e}_j$ is the j-th standard basis vector of according length. We shall use Equation 9 to reformulate this identifiability condition. To this end, recall the earlier assertion about the full (column) rank of $\begin{bmatrix}\hat{P}_i & \bar{P}_i\end{bmatrix}$, from which follows that

$$\forall\, \tilde{\boldsymbol{w}} \in \mathbb{R}^{\mathrm{p}},\ \exists\, \boldsymbol{w} \in \mathbb{R}^{\mathrm{q}} :\ \tilde{\boldsymbol{w}}^T = \boldsymbol{w}^T \begin{bmatrix}\hat{P}_i & \bar{P}_i\end{bmatrix},$$

so that we can write

$$[\boldsymbol{x}_i]_j \text{ identifiable } \iff \exists\, \tilde{\boldsymbol{w}} \in \mathbb{R}^{\mathrm{p}} : \tilde{\boldsymbol{w}}^T\, \Psi_i = \boldsymbol{e}_j^T.$$

Next, let $\tilde{\boldsymbol{w}}_1$ and $\bar{\boldsymbol{w}}_2$ consist of the first $|\hat{\boldsymbol{\nu}}_i|$ and the last $|\bar{\boldsymbol{\nu}}_i|$ components of $\tilde{\boldsymbol{w}}$, such that $\tilde{\boldsymbol{w}}^T = [\tilde{\boldsymbol{w}}_1^T \quad \tilde{\boldsymbol{w}}_2^T]$. Accordingly, standard base vector $\boldsymbol{e}_j$ is split into its first $|\hat{\boldsymbol{\mu}}_i|$ and last $|\hat{\boldsymbol{\nu}}_i|$ components, $\boldsymbol{e}_j^T = [\boldsymbol{f}_j^T \quad \boldsymbol{g}_j^T]$. This allows to rewrite the previous equation as

$$\begin{aligned}\tilde{\boldsymbol{w}}_1 &= \boldsymbol{g}_j,\ \text{and}\\ \tilde{\boldsymbol{w}}_2^T\left(\bar{S}_i\hat{J}_i^{-1}\right) &= \boldsymbol{f}_j^T - \boldsymbol{g}_j^T\left(\hat{S}_i\hat{J}_i^{-1}\right).\end{aligned}$$

Recall that $[x_i]_j$ denotes unknown interaction strengths for $j \leq |\hat{\boldsymbol{\nu}}_i| \iff \boldsymbol{g}_j = \boldsymbol{0}$ and thus

$$\begin{aligned}[\hat{\boldsymbol{j}}_i]_j \text{ identifiable } &\iff \operatorname{rank}\left(\begin{bmatrix}\bar{S}_i\hat{J}_i^{-1}\\ \boldsymbol{f}_j^T\end{bmatrix}\right) = \operatorname{rank}(\bar{S}_i\hat{J}_i^{-1})\\ &\iff 1 + \operatorname{rank}(\bar{S}_i\hat{J}_{i\setminus j}^{-1}) = \operatorname{rank}(\bar{S}_i\hat{J}_i^{-1}),\end{aligned} \qquad (14)$$

where $\hat{J}_{i\setminus j}^{-1}$ is matrix $\hat{J}_i^{-1}$ with the j-th column removed. For the unknown sensitivity coefficients, where $j > |\hat{\boldsymbol{\nu}}_i| \iff \boldsymbol{f}_j = \boldsymbol{0}$, we find the identifiability conditions

$$\begin{aligned}&[\hat{\boldsymbol{s}}_i]_j \text{ identifiable}\\ &\iff \operatorname{rank}\left(\begin{bmatrix}\bar{S}_i\\ \hat{\boldsymbol{S}}_i^j\end{bmatrix}\hat{J}_i^{-1}\right) = \operatorname{rank}(\bar{S}_i\hat{J}_i^{-1}),\end{aligned} \qquad (15)$$

where $\hat{\boldsymbol{S}}_i^j$ denotes the j-th row of matrix $\hat{S}_i$.

Structural identifiability

The identifiability conditions in equations 14 and 15 relate the identifiability of the unknown parameters to a discussion of the rank of matrix product $\bar{S}_i\hat{J}_i^{-1}$. The product however depends on the unknown parameters themselves, so that its rank cannot be directly computed. Here we show that a reasonable assumption make this possible nevertheless and allows to express the identifiability conditions as a very intuitive maximum flow problem.

First, we rewrite the identity $J^{-1}J = I_{\mathrm{n}}$ as

$$[J^{-1}]_{kl} = \sum_{m \neq l} [J^{-1}]_{km}\, [J]_{ml} - \delta_{kl},$$

with δ_{kl} being the Kronecker delta (recall that $J_{ll} = -1$). We can view this equation as a recurrence relation and repeatedly replace the $[J^{-1}]_{km}$ terms in the sum. The sum contains non-vanishing terms for each edge that leaves node l. Therefore, each replacement leads to the next downstream node, so that eventually one arrives at

$$\begin{aligned}[J^{-1}]_{kl} &= l \rightarrowtail k\ \ [J^{-1}]_{kk},\ \text{with}\\ l \rightarrowtail k &= \sum_{\boldsymbol{\omega}\, \in\, \Omega_{l\to k}} \prod_{m=1}^{|\boldsymbol{\omega}|-1} [J]_{\omega_{m+1}\,\omega_m},\end{aligned}$$

where the set $\Omega_{l\to k}$ contains elements, $\boldsymbol{\omega}$, for every path from node l to node k, each of which lists the nodes along that path. Strictly speaking, these elements are walks rather than paths because some nodes will appear multiple times if loops exist between l and k. With loops, $\Omega_{l\to k}$ even contains an infinite number of walks of unbounded lengths. But as the real part of all eigenvalues of J are assumed negative, the associated products of interaction strengths will eventually converge to zero with increasing walk length. Here however, we can safely ignore these subtleties.

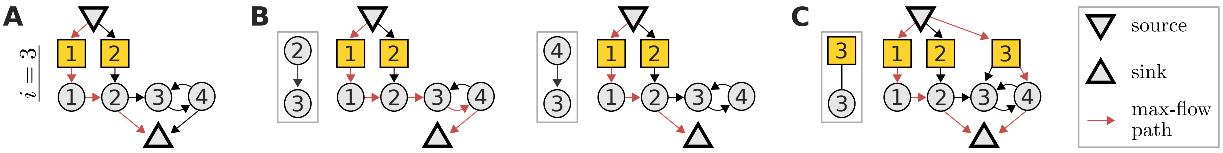

Figure S2: A maximum flow problem determines the identifiability of interaction strengths and perturbation sensitivities when reconstructing a network from perturbation data. Here, this is illustrated for the toy model from Figure S1A. To inquire about the identifiability of either edges going into node 3, or the sensitivity of node 3 to perturbations, we construct a flow network (**A**) with unit edge and node capacities, as described in the text. We highlight in red a path carrying the maximal flow of one. While this max-flow path is not unique, no other combination of paths could yield a larger flow. The interaction strength between a given node and node 3 is identifiable if and only if the maximum flow is reduced after removing that node's edge to the sink node (**B**). Yet here, we can always find alternative max-flow paths that re-establish a unit-flow after removal of the according edges. Thus the respective edges are non-identifiable. Similarly, the sensitivity of node 3 to perturbation 3 is identifiable if and only if a specific extension of the flow network (**C**) does not increase the maximum flow. But here the maximum flow is indeed increased by one, which again reveals non-identifiability. Such flow representations also provide an intuitive understanding on how alterations in the network or perturbation setting affect identifiability. For example, it is obvious that if the toy model would not contain an edge from node 3 to 4, the edge from 2 to 3 would become identifiable.

To simplify our notation, we want to expand the network by considering perturbations $\bar{\boldsymbol{\nu}}_i$ as additional nodes, each with edges that are directed towards that perturbation's targets. Furthermore, letting the interaction strength associated with these new edges be given by the appropriate entries in S we can rewrite the matrix product

$$\left[\bar{S}_i\hat{J}_i^{-1}\right]_{kl} = \bar{\nu}_{ik} \rightarrowtail \hat{\mu}_{il}\ [J^{-1}]_{\hat{\mu}_{il}\,\hat{\mu}_{il}}$$

where $\hat{\mu}_{il}$ and $\bar{\nu}_{il}$ denote the l-th entry in $\hat{\boldsymbol{\mu}}_i$ and $\bar{\boldsymbol{\nu}}_i$, respectively. As every finite-dimensional matrix has a rank decomposition, we can further write

$$\bar{S}_i\hat{J}_i^{-1} = \Upsilon_i\, Y_i, \tag{16}$$

where $|\bar{\boldsymbol{\nu}}_i| \times \mathrm{rank}(\bar{S}_i\hat{J}_i^{-1})$ matrix Υ_i and $\mathrm{rank}(\bar{S}_i\hat{J}_i^{-1}) \times |\hat{\boldsymbol{\mu}}_i|$ matrix Y_i have full rank. Finding such a decomposition therefore reveals the rank of $\bar{S}_i\hat{J}_i^{-1}$. To this end, we propose

$$[\Upsilon_i]_{kn} = \bar{\nu}_{ik} \rightarrowtail y_{in}, \text{ and } [Y_i]_{nl} = y_{in} \rightarrowtail \hat{\mu}_{il}\ [J^{-1}]_{\hat{\mu}_{il}\,\hat{\mu}_{il}},$$

where y_{in} denotes the n-th component of a certain node set $\boldsymbol{y}_i$. In order for Equation 16 to hold, it must be possible to split each path from any perturbation $\bar{\nu}_{il}$ to any node $\hat{\mu}_{il}$ into a section that leads from the perturbation to a node in $\boldsymbol{y}_i$ and a subsequent section that leads from this node to $\hat{\mu}_{il}$. For an extended graph that includes an additional source node, with outgoing edges to each perturbation in $\bar{\boldsymbol{\nu}}_i$, and an additional sink node, with incoming edges from all nodes in $\hat{\boldsymbol{\mu}}_i$ (see Figure S2A), $\boldsymbol{y}_i$ thus constitutes a vertex cut whose removal disconnects the graph and separates the source and the sink node into distinct connected components. Next, we want to show that if $\boldsymbol{y}_i$ is a minimum vertex cut, the rank of $\bar{S}_i\hat{J}_i^{-1}$ equals the size of $\boldsymbol{y}_i$. Because Equation 16 is a rank decomposition this is equivalent to showing that the according matrices Υ_i and Y_i have full rank. To do so we apply Menger's theorem [11], which states that the minimal size of $\boldsymbol{y}_i$ equals the maximum number of vertex-disjoint paths from the source to the sink node. This also implies that each of these vertex-disjoint paths goes through a different node of the vertex cut $\boldsymbol{y}_i$. Recall that entries in Υ_i constitute sums over paths from perturbation to vertex cut nodes, so that we could write

$$\Upsilon_i = \bar{\Upsilon}_i + \hat{\Upsilon}_i,$$

where $\bar{\Upsilon}_i$ only contains the vertex-disjoint paths and $\hat{\Upsilon}_i$ the sums over the remaining paths. As each of these vertex disjoint paths ends in a different vertex cut node, any column in $\bar{\Upsilon}_i$ can contain no more than a single non-zero entry. Furthermore, as a consequence of Menger's theorem there are exactly $|\boldsymbol{y}_i|$ non-zero columns. Because these paths are indeed vertex disjoint also no row in $\bar{\Upsilon}_i$ has more than a single non-zero entry. Thus, the non-zero columns are independent, showing that $\bar{\Upsilon}_i$ has full rank. We further assume that adding $\hat{\Upsilon}_i$ does not reduce rank, which also gives Υ_i full rank. In the context of biological networks there are two different scenarios that could lead to a violation of this non-cancellation assumption. The first is that network parameters are perfectly tuned to lie inside a specific algebraic variety (a manifold in parameter space) such that certain columns (or rows) of Υ_i become linearly dependent or zero. This would for example be the case if, for a given vertex disjoint path, there also is an alternative path whose associated product of interaction strengths has the same magnitude as that of the vertex disjoint path but opposite sign, making their sum vanish. However, we consider it implausible for biological networks to be fine-tuned to such a degree that they could achieve such perfect self-compensation of perturbations, and rule out this possibility. A more realistic scenario is that network parameters are zero and thereby lead to zero columns or rows in Υ_i or Y_i, which make these matrices rank deficient. In practice, such zero-parameters can occur, for example, if a perturbation is not effective on (one of) its target(s), or if robustness effects [7] obstruct the propagation of the perturbation signal at a certain link. But essentially, this means that our prior knowledge about the network included practically non-existing links or perturbation targets. If the network topology and perturbation targets are correctly stated and take these effects into consideration, there will be no zero-parameters and therefore the non-cancellation assumption holds. We explore the consequences of incomplete or flawed prior knowledge in section S5.

Having shown Υ_i to be of full rank, the same line of reasoning will demonstrate a full rank for matrix Y_i as well, which implies that indeed

$$\operatorname{rank}(\bar{S}_i \hat{J}_i^{-1}) = |\boldsymbol{y}_i|, \tag{17}$$

where $\boldsymbol{y}_i$ is a minimum vertex cut between source and sink node. This equation has the crucial benefit that $|\boldsymbol{y}_i|$ does not depend on any unknown parameters and can be computed as the maximum flow from source to sink node with all nodes having unit capacity [1], as detailed in Figure S2B. A flow is defined as a mapping from a network edge to a positive real number that is smaller than the edge's capacity. Additionally, the sum of flows entering a node must equal the sum of the flows exiting a node, except for the source and the sink nodes. The maximum flow problem is to attribute (permissible) flow values to all edges, such that the sum of flows leaving the source (which is equal to the sum of flows entering the sink) is maximal. In our case however, we did not define edge but node capacities, meaning that the sum of flows passing through any node must not exceed one. Yet, we can express such unit node capacities as unit edge capacities in an extended flow network. It is defined by replacing every node by an *in*- and an *out*-node, where all incoming edges target the *in*-node, all outgoing edges start from the *out*-node, and the *in*-node has an edge to the *out*-node.

This maximum flow problem allows to express the algebraic identifiability conditions 14 and 15 in terms of network properties, providing an intuitive relationship between network topology, perturbation targets and identifiability. Specifically, $J_{i\hat{\mu}_{ij}}$ is identifiable if and only if the removal of the edge from node $\hat{\mu}_{ij}$ to the sink node reduces the maximum flow of the network, see Figure S2C, and $S_{i\hat{\nu}_{ij}}$ is identifiable if the maximum flow does not increase when an additional edges connects the source node with perturbation node $\hat{\nu}_{ij}$, see Figure S2D. In section S5, we simulate a perturbation experiment to numerically verify these findings.

S2 Identifiability relationships

Network inference typically is an underdetermined problem for which the number of measurements falls short on the number of unknown interaction terms [4, 9], resulting in many non-identifiable parameters. To tackle this problem, we could construct identifiable models by fixing certain parameters to some constant values. Clearly, the remaining, inferred parameter values will then disagree with those that would have been obtained from a fully-determining experiment. Nevertheless, such effective models are useful as they allow for meaningful comparisons of the inferred parameters between perturbation experiments on similar systems, e.g. when studying the same signalling pathway in different cell lines [3]. To derive such a determined system requires to study the relationship between non-identifiable parameters in the sense that we ask which parameters need to be fixed in order to render which other parameters identifiable. Even though the dimensionality of the solution space, d_i, is known, this question is not trivial, because even groups with d_i or fewer parameters might already be linearly dependent and fixing them will therefore not effectively reduce the degrees of freedom of the equation system.

Take as example a case where the first two rows of kernel matrix V_i from Equation 12, are linearly dependent, that is $\alpha\,\boldsymbol{V}_i^1 = \boldsymbol{V}_i^2$. Then $[\boldsymbol{x}_i]_1$ and $[\boldsymbol{x}_i]_2$ are linearly dependent as well, $[\boldsymbol{x}_i]_2 = \boldsymbol{V}_i^2\,\boldsymbol{v} = \alpha\,\boldsymbol{V}_i^1\,\boldsymbol{v} = \alpha\,[\boldsymbol{x}_i]_1$, which implies that $[\boldsymbol{x}_i]_2$ becomes identifiable if $[\boldsymbol{x}_i]_1$ is known, and vice versa, even if $\mathrm{d}_i > 1$ ($\tilde{\boldsymbol{x}}_i$ was dropped to simplify notation). Moreover, prior knowledge on both $[\boldsymbol{x}_i]_1$ and $[\boldsymbol{x}_i]_2$ would overdetermine this linear subsystem and not further reduce the degrees of freedom for the remaining unknown parameters. Examining such parameter dependencies is a direct generalization of the original identifiability condition in Equation 13. There, identifiability of an unknown parameter relied on a V_i-row being zero, that is, on a one-row submatrix being rank deficient. Now, we inspect not only single but groups of V_i-rows for rank deficiency. But which groups of rows should we consider to achieve an effective description of dependency? To answer this question let us first generalize the previous example.

We were asking if the j-th $\boldsymbol{x}_i$ component becomes identifiable if a set of other $\boldsymbol{x}_i$ components is known. With $\mathcal{I}$ denoting the set of indices of these other components, let us recall Equation 12 and name their homogenous parts

$$\hat{x}_i^{\mathcal{I}} = \boldsymbol{V}_i^j \boldsymbol{v} \quad \text{and} \quad \bar{\boldsymbol{x}}_i^{\mathcal{I}} = V_i^{\mathcal{I}}\,\boldsymbol{v},$$

where $\boldsymbol{V}_i^j$ is the j-th row of V_i, and $V_i^{\mathcal{I}}$ the matrix that gathers all V_i rows with indices in $\mathcal{I}$. We can then put down a formal identifiability statement

$$\begin{aligned} \exists\, \mathcal{I} \subseteq \{1, \ldots, \mathrm{u}_i\} \setminus j,\ \exists\, \boldsymbol{w} \in \mathbb{R}^{|\mathcal{I}|} :\ \boldsymbol{V}_i^j &= \boldsymbol{w}^T\, V_i^{\mathcal{I}} \\ \Longleftrightarrow\ \hat{x}_i^{\mathcal{I}} = \boldsymbol{w}^T\, V_i^{\mathcal{I}}\, \boldsymbol{v} &= \boldsymbol{w}^T \bar{\boldsymbol{x}}_i^{\mathcal{I}}. \end{aligned} \tag{18}$$

In other words, if the j-th V_i-row lies within the rowspace of the set of V_i-rows with indices $\mathcal{I}$, the j-th unknown parameter can be expressed as a linear combination of the set of parameters with indices $\mathcal{I}$. This means that knowledge of the set of parameters with indices $\mathcal{I}$ then implies identifiability of the j-th parameter. However, this statement does not imply the uniqueness of $\mathcal{I}$. On the contrary, if the j-th V_i-row lies within the $\mathcal{I}$-associated rowspace, it will also do so if additional V_i rows are added to the set. Similarly, there could be a linearly dependent subset of V_i-rows that all lie within the $\mathcal{I}$ associated row-space. This would allow for multiple row-combinations to span the $\mathcal{I}$-associated rowspace and thus implicate the identifiability statement. Both cases show, that various combinations of additionally fixed parameters can imply the identifiability of a certain other parameter.

A comprehensive description of this combinatorial space arises from a mathematical structure that has been termed matroid [16]. Matroids are a generalized description of linear independence in vector spaces. Here we are concerned with representable matroids, which are those that specify linear (in-)dependence of any combination of columns of a matrix. Amongst their various equivalent definitions, the one that relates directly to our problem is the definition in terms of cyclic flats (also called circuit closures) and their ranks [13]. To specify these we need to define a few terms. First, let $\mathcal{E}$ be the ground set of matroid $\mathcal{M}$, that is, the set of indices enumerating the columns of the associated matrix. Furthermore, define a circuit as a dependent set (of columns) whose proper subsets are all independent. The set of circuits can be enumerated with an incremental polynomial-time algorithm [2]. Finally, we define a flat as a subset of $\mathcal{E}$, with the associated submatrix having rank r, such the addition of any other element to the set would increase the rank. With this we can define $\mathcal{C}_r$, a cyclic flat of rank r, as a flat that is the union of a set of circuits with rank r. We show in the next section how to obtain cyclic flats from circuits and vice versa.

Let us now consider $\mathcal{M}_i$, the matroid whose groundset ε_i covers the u_i columns of $(V_i)^T$. Each element in ε_i is thus associated with an unknown parameter. The key inside is that $\mathcal{M}_i$'s set of circuits fully characterizes the identifiability relationships between the non-identifiable parameters. This is because the circuit dependency implies that any parameter represented by a given circuit element is identifiable when the remaining circuit elements are known. Additionally, this set of remaining parameters is guaranteed to be minimal because they are linearly independent. The enumeration of the circuits with the aforementioned algorithm requires a dependence oracle that indicates whether a column subset is dependent or not. For this, we first consider another matroid $\mathcal{M}_i'$, which is associated with the u_i columns of Ψ_i, as defined in Equation 10. Because V_i spans the kernel of matrix Ψ_i, $\mathcal{M}_i'$ is dual to $\mathcal{M}_i$ [16]. This implicates that the rank of the $(V_i)^T$ column-subset $\mathcal{I}$ relates to that of the complementary columns $\tilde{\mathcal{I}} = \varepsilon_i \setminus \mathcal{I}$ of Ψ_i as follows

$$\mathrm{rank}_{\mathcal{M}_i}(\mathcal{I}) = \mathrm{rank}_{\mathcal{M}_i'}(\tilde{\mathcal{I}}) + |\mathcal{I}| - (\mathrm{u}_i - \mathrm{d}_i).$$

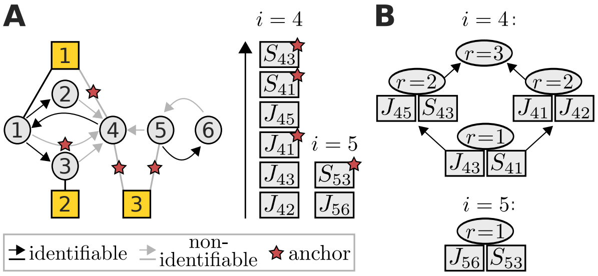

Figure S3: In this toy network (**A**), nodes 4 and 5 are associated with non-identifiable parameters. These can take values from certain linear sub-spaces whose hierarchy is represented by the lattices of cyclic flats of rank r (**B**). Each cyclic flat consists of the annotated elements in addition to elements from its preceding cyclic flats. To achieve identifiability requires to set certain parameters to a constant value. A preference to which parameters this should be is represented here as a ranked list (arrow indicates direction of increasing preference). The matroid formalism identifies the smallest and most preferred set of parameters that, when set to a constant value, render the network model fully identifiable. Here these are marked by red stars.

To investigate the dual rank, we note that we can establish the column subset of Ψ_i by a right multiplication with the $\mathrm{u}_i \times |\tilde{\mathcal{I}}|$ matrix $\mathcal{P}$, which is an identity matrix where columns that correspond to missing indices in $\tilde{\mathcal{I}}$ are removed. Furthermore, we subdivide elements in $\tilde{\mathcal{I}}$ into sets $\tilde{\mathcal{I}}_1$ and $\tilde{\mathcal{I}}_2$ based on whether they are less than or equal to $|\hat{\boldsymbol{\mu}}_i|$ or not, which allows to define matrices $\mathcal{P}_1$ and $\mathcal{P}_2$ by the partitioning

$$\mathcal{P} = \begin{bmatrix} \mathcal{P}_1 & 0_{|\hat{\boldsymbol{\mu}}_i|,|\tilde{\mathcal{I}}_2|} \\ 0_{|\hat{\boldsymbol{\nu}}_i|,|\tilde{\mathcal{I}}_1|} & \mathcal{P}_2 \end{bmatrix}. \tag{19}$$

Then,

$$\begin{aligned} \mathrm{rank}_{\mathcal{M}_i'}(\tilde{\mathcal{I}}) &= \mathrm{rank}(\Psi_i\,\mathcal{P}) \\ &= \mathrm{rank}\left(\begin{bmatrix} \hat{S}_i\hat{J}_i^{-1}\,\mathcal{P}_1 & \mathcal{P}_2 \\ \bar{S}_i\hat{J}_i^{-1}\,\mathcal{P}_1 & 0_{|\bar{\boldsymbol{\nu}}_i|,|\tilde{\mathcal{I}}_2|} \end{bmatrix}\right) \\ &= |\tilde{\mathcal{I}}_2| + \mathrm{rank}\left(\begin{bmatrix} \tilde{\mathcal{P}}_2^T\hat{S}_i \\ \bar{S}_i \end{bmatrix}\hat{J}_i^{-1}\,\mathcal{P}_1\right), \end{aligned} \tag{20}$$

where $\tilde{\mathcal{P}}_2$ is the identity matrix without the columns that appear in $\mathcal{P}_2$. Left-multiplication by $\tilde{\mathcal{P}}_2^T$ thus selects rows that correspond to missing indices in $\tilde{\mathcal{I}}_2$. The crucial point of this calculation is that we arrived at a matrix product that has the same form as the one discussed in the previous section. Therefore, the dual rank can be evaluated independently of the unknown entries in J and S because the last term in the previous equation equals to the maximum flow through the associated network, with connections from the source and to the sink nodes that are chosen according to $\tilde{\mathcal{I}}$, as shown. This allows to construct the oracle and identify the set of circuits. Therefore the identifiability relationships between unknown parameters can be inferred from information about network topology and perturbation targets alone.

Instead of listing the set of circuits, we propose cyclic flats as an equivalent but more concise representation of the identifiability relationships. They form a geometric lattice when ordered by inclusion (a cyclic flat precedes another if it is its proper subset) and can thus be graphically represented as a compact hierarchical structure. We demonstrate this for the example network shown in Figure S3. The depicted lattice makes the identifiability relationships evident. All elements of a cyclic flat with rank r become identifiable if at least r independent flat elements are fixed. A set of elements is independent if fixing any combination if its elements does not render any of its other element identifiable. Let us clarify this at an example where we are interested in determining the parameters that need to be fixed in order to make J_{43} identifiable. Following the previous rules, Figure S3 reveals that this could be achieved by fixing either S_{41} alone, or the parameter pairs $J_{45} \cup S_{43}$ or $J_{41} \cup S_{42}$. In the latter two cases S_{53} would become identifiable as well.

When the goal is to achieve a fully identifiable network model, as discussed before, there typically are preferences as to which non-identifiable parameters should be fixed. For example, if there is noisy external data on parameter values we would rather fix those parameters values in which we have high confidence. Or, if we are to construct the aforementioned effective signalling models for the comparison of different cell lines, we would want to fix those parameters, which we expect to be equal between different cell lines and infer those parameters for which cell line differences are expected [3]. Thus, in these scenarios fixing of each parameter is associated with a certain preference (weight) and our goal is to find a minimum number of parameters that need to be fixed such that their sum of weights is maximal. In fact, matroids owe their striking appearance in combinatorial optimization because this problem is solvable with the Greedy Algorithm [12, 8]: Amongst the set of non-identifiable parameters in ϵ_i, sequentially select the parameters with highest weight, that have not yet become identifiable from fixing the so-far selected set. Thus, instead of providing numerical weights for unknown parameters it is sufficient to rank them. We depict examples of such ordered lists in Figure S3 and show the resulting fully identifiable maximum-weight-model.

Circuits and circuit closures

As both, the set of circuits and the circuit closures combined with their ranks, are an equivalent definition of a matroid they imply each other. Recall that circuits that contain a given network parameter describe the minimal sets of network parameters that need to be fixed to render that parameter identifiable. The flat of closures conveniently display these circuits as follows. By definition, any circuit is a $r+1$-element subset, $\mathcal{S}$, of some cyclic flat $\mathcal{C}_r$ with rank r. Thus, to obtain all circuits containing a certain parameter, consider all such subsets of cyclic flats that include this parameter. Yet $\mathcal{S}$ is only a circuit if none of its subsets $\underline{\mathcal{S}} \subset \mathcal{S}$ is dependent, in which case there is another circuit $\underline{\mathcal{C}} \subseteq \underline{\mathcal{S}}$. Since the lattice of cyclic flats is ordered by inclusion, $\underline{\mathcal{C}}$ is a subset of a cyclic flat that precedes $\mathcal{C}_r$ in the lattice. Therefore, $\mathcal{S}$ is only a circuit if no cyclic flat preceding to $\mathcal{C}_r$ contains a circuit that is a proper subset of $\mathcal{S}$.

We mentioned that circuits can be enumerated in incremental polynomial-time [2]. In a next step, we generated circuit closures from the set of circuits. To this end, we first order circuits by size and iterate through that list. For each circuit of a given rank we identify circuits of up to its size whose intersection is equal or larger to its rank. Their union forms a circuit closure. Next, one continues the circuit iteration while skipping circuits that have already been assigned to a circuit closure. Eventually, this generates the entire ensemble of circuit closures. Find an implementation in the function `circuits2cyclic_flats` which is part of the `identifiability` module of the IdentiFlow package available at `github.com/GrossTor/IdentiFlow`.

S3 Experimental Design

Next, we describe the algorithmic implementation of the experimental design strategies.

Depth-first search in strategy graph

The power set of the set of perturbations ordered by inclusion forms a directed graph (more formally a graded poset), where ancestors are proper subsets with one less element (e.g. $\{P1, P2\}$, $\{P1, P3\}$, and $\{P2, P3\}$ are all ancestors of $\{P1, P2, P3\}$). Any perturbation sequences can thus be represented as a path on this graph, starting from the empty subset. Different experimental design strategies remove different subsets of edges, which yields what we want to call the strategy graph. Therefore strategies are associated with different subsets of (or even just single) sequences. To enumerate all strategy-associated perturbation sequences, we implemented a recursive depth-first search on the strategy graph:

```
1:  procedure DFS($\{S\}$)
2:      $S^+ \leftarrow$ NEXT_PERTS($\{S\}$)
3:      for $\{S\} \in S^+$ do
4:          fully_identifiable $\leftarrow$ MAX_FLOW($\{S\}$)
5:          if not fully_identifiable then
6:              DFS($\{S\}$)
7:          else
8:              save $S$
9:          end if
10:     end for
11: end procedure
```

Here, S denotes a perturbation sequence and $\{S\}$ the set of perturbations in the sequence. To enumerate perturbation sequences, the DFS procedure is called with the empty set. It then parses the strategy graph and adds perturbations to the sequence sequentially. The strategy graph is built up dynamically in line 2 by the NEXT_PERTS($\{S\}$) function, as it returns S^+, the set of descendants of $\{S\}$. We will define it for the different strategies further below. Every descendant (line 3) will instantiate a new instance of the DFS procedure (line 6), which will in turn continue parsing the graph, unless the descendent is either the maximal perturbation set, or a set of perturbations that fully determines the network. In this case, additional perturbations provide no additional network information, so that the search can be aborted and a strategy-associated sequence is found (and therefore saved in line 8). To query for network identifiability (line 4), we employ the maximum flow approach described in the previous sections. Note that, to enumerate all sequences the depth-first search does not terminate when it reaches a perturbation set that has previously been encountered. To avoid redundant computational effort, the calls to NEXT_PERT and MAX_FLOW are stored (Memoization). Strictly speaking, this violates the definition of a depth-first search. Nonetheless, we want to keep the terminology due to our procedure's apparent analogy.

Let $\{\bar{S}\}$ denote the set of all perturbations that are not in $\{S\}$, and let $\hat{S}$ denote all proper supersets of $\{S\}$ with size $|\{S\}| + 1$. That is, $\hat{S}$ contains all perturbation sets that we can obtain by adding one element from $\{\bar{S}\}$ to $\{S\}$. As mentioned before, NEXT_PERT($\{S\}$) returns a strategy-dependent subset of $\hat{S}$. For the *exhaustive* strategy a call to NEXT_PERT($\{S\}$) simply returns $\hat{S}$ itself. Therefore, in this case the strategy graph coincides with the original power set inclusion graph and DFS($\{\emptyset\}$) will store all perturbation sequences. The *random* strategy works similarly except that only a single perturbation set is chosen randomly from $\hat{S}$. Thereby, DFS($\{\emptyset\}$) will return a single random perturbation sequence. The *naive* strategy considers the perturbed nodes for each perturbation and computes the number of network nodes to which these are connected to by a path. It then selects the perturbations in $\{\bar{S}\}$ that maximize this number and NEXT_PERT($\{S\}$) returns the according subset from $\hat{S}$. In contrast, the *single-target* strategy selects the next perturbation candidates based on whether they efficiently reduce the degrees of freedom of the network. More specifically, the maximum-flow approach is applied (and memoized) for every perturbation set in $\hat{S}$. Amongst them, NEXT_PERT($\{S\}$) returns those that first maximize the number of identifiable interaction strengths and second minimize the sum of solution space dimensionalities, $\sum d_i$, as defined in Equation 11. Finally, the *multi-target* strategy is equivalent to the single-target strategy, except that it expands the set of possible perturbations by allowing for any combination of perturbations. For example, if originally there is a perturbation targeting each single node of the network, the multi-target approach would allow to pool perturbations such that there are single perturbations to target any set of nodes. Clearly, considering the entire power set of perturbation combinations makes such an approach feasible only for less than ten (original) perturbations (see a discussion on computational complexity further below). Therefore, we also also implemented a more efficient, hierarchical multi-target strategy, which was also the one applied in the analysis of the KEGG pathways (see next section and main text). Here, the considered set of perturbation combinations is built-up in a step-wise manner. First, we only consider single and pair perturbations (of elements in $\{S\}$). Out of these, we perform a selection as in the single-target strategy. If a pair perturbation was within the selection, we also consider all combinations of three perturbations for the selection procedure. This continues until no perturbation combination of largest size is in the selection or the entire power set of $\{S\}$ is considered.

Analogous to the random strategy that makes a random choice amongst the candidate perturbations of the exhaustive strategy, we implemented the option to randomly pick a single perturbation set amongst the possible candidates also for the naive, the single- and multi-target strategies. Thus, a run of DFS($\{\emptyset\}$) will then select a single perturbation sequence. Repeated calls to DFS($\{\emptyset\}$) will thus generate random samples amongst the set of perturbation sequences that are associated with the chosen strategy. This sampling procedure becomes essential if the number of perturbations and strategy associated sequences becomes too large to make a complete depth-first search computationally tractable. Let us thus briefly characterize the computation complexity of the experimental design strategies.

Computational complexity of experimental design strategies

The computational complexity of the depth-first search is dominated by the calls to the MAX_FLOW routine. We therefore want to count how many times it gets called by different strategies. As the parsing of the strategy graph stops whenever a fully determining perturbation set is reached, this number is not just a function of network size (n) and number of perturbations (p), but will crucially depend on the specific network topology and perturbation targets. To still provide a rough estimate for an upper complexity bound, we will disregard such early stopping. Due to memoization, MAX_FLOW will not be called repeatedly if a certain perturbation set is revisited during the depth-first search. Thus, its number of calls equals to the number of different perturbation sets that were parsed during the depth-first search. For the exhaustive strategy this will be all 2^{p} nodes. For all other strategies besides the random strategy, this number will again be highly sensitive to the specific perturbation network so that we cannot make any general statements. Thus, we want to consider the case where we randomly sample a single strategy associated sequence, as described before. Then the naive and random strategies will parse p perturbation sets (where for each perturbation set the naive strategy has the additional overhead of computing the most

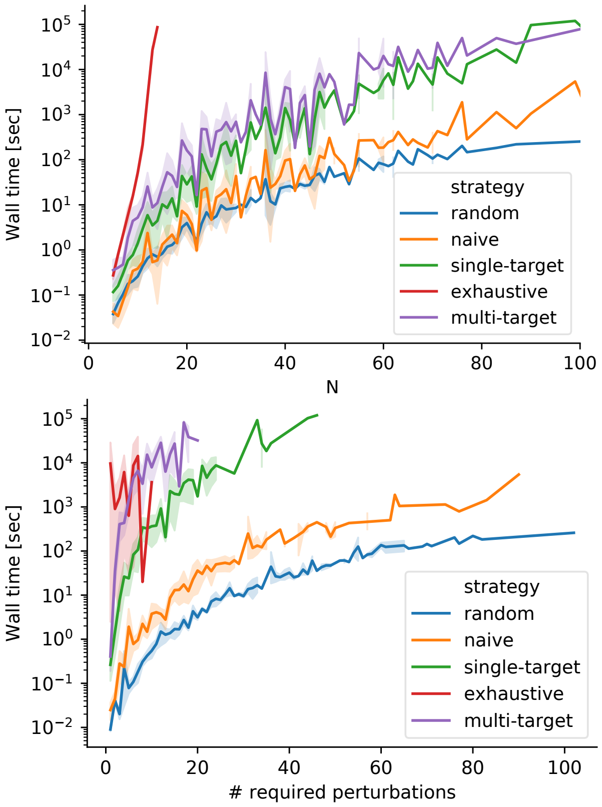

Figure S4: Computational running times to compute perturbation sequences on KEGG pathways with different experimental design strategies. Lines go through mean wall times over all KEGG pathways of the same size in the upper graph and all KEGG pathways that require the same number of perturbations for full identifiability in the lower graph.

upstream perturbations as described above). For every perturbation set $\{S\}$ that is parsed by the single-target strategy, NEXT_PERT($\{S\}$) will call MAX_FLOW for every descendant of $\{S\}$ in the strategy graph. Due to memoization, this yields $\sum_i^{\mathrm{p}}(\mathrm{p}-i) \propto \mathrm{p}^2$ calls. The number of MAX_FLOW calls used by the (hierarchical) multi-target strategy is again highly dependent on the specific perturbation network. But at least, this strategy will additionally consider all pair perturbations and thus yield more than $\sum_i^{\mathrm{p}}(\mathrm{p}+\mathrm{p}\,(\mathrm{p}-1)/2-i) \propto \mathrm{p}^3$ calls.

Finally, we also require an estimate for the complexity of MAX_FLOW($\{S\}$). Also here, the flow network as defined in previous sections varies with network topology and perturbation targets and so does the computational effort to compute maximum flow. We will thus make some estimations based on the assumption that biological networks are rather sparse, with a number of edges that is roughly proportional to the number of nodes and that perturbations tend to be specific to a few nodes. Thus for the unit edge capacity flow network (recall the definition in the Methods of the main text), we assume for the number of edges, $E \approx \mathrm{n}+\mathrm{n}+\mathrm{p}+\mathcal{O}(1)$ ($\approx$ edges in original network + edges between *in*- and *out*-nodes + source edges + sink edges). Due to the conversion in *in*- and *out*-nodes, a flow network has $N = 2\,\mathrm{n}$ nodes. As noted in the main text, algorithms are known to find maximal flows in unit capacity networks with $\mathcal{O}(\min(N^{2/3}E,\, E^{3/2})$ computations [1]. However, in the IdentiFlow package, we implemented the well established Edmonds-Karp algorithm which has a complexity of $\mathcal{O}(N\,E^2)$. To determine the identifiability of the entire networks requires to solve n maximum-flow problems. Thus, we can estimate the computational complexity of a call to MAX_FLOW($\{S\}$) with $\mathcal{O}(\mathrm{n}^4+\mathrm{n}^2\mathrm{p}^2)$. Overall, this gives the following upper bounds for the complexity of the experimental design computations

strategy	complexity
random / naive	$\mathcal{O}(\mathrm{n}^4\mathrm{p}+\mathrm{n}^2\mathrm{p}^3)$
single-target	$\mathcal{O}(\mathrm{n}^4\mathrm{p}^2+\mathrm{n}^2\mathrm{p}^4)$
multi-target	$\mathcal{O}(\mathrm{n}^4\mathrm{p}^3+\mathrm{n}^2\mathrm{p}^5)$
exhaustive	$\mathcal{O}(\mathrm{n}^4 2^{\mathrm{p}})$

In practice, the computational effort is often much lower than these theoretical bounds suggest. To this end, we measured computational running times that were needed to determine the experimental designs for a collection of KEGG pathways (for more details see next section). Figure S4 shows the according wall times, where each perturbation sequence was computed on a single core with 2.3 Ghz. Note that network size and the number of perturbations that is required for full identifiability are not independent of each other (see Figure S6).

S4 Perturbation experiments for KEGG pathways

KEGG data [10] was retrieved using the KEGG API. We retrieved KGML files for human pathways and from them build network representations based on their 'relation elements'. For each such representation we computed the size of its largest connected component. The pathway was filtered out if it was smaller than five.

The performance of the exhaustive strategy could be observed for small pathways Figure S5 A. In addition, we further confirmed our hypothesis that the isolation score is predictive with respect to the performance of the design strategies Figure S5 B. Furthermore, Figure S6 compares for each KEGG pathway the number of perturbations that are required to achieve a fully identifiable network using the different strategies. We also studied how our experimental design strategies compared against a strategy that chooses random sequence of combination perturbations. Thus, for each KEGG pathway, we generated perturbation sequences by sequentially drawing perturbation combinations from the power set of single perturbations (excluding the empty set) and measured their performance, shown in Figure S7 (annotated as multi-random). The multi-random

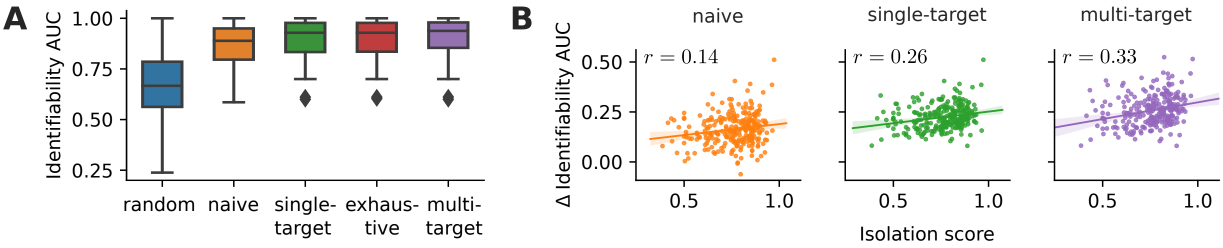

Figure S5: Performance of different experimental design strategies on 78 human KEGG pathways with up to 15 nodes (**A**). The single-target and exhaustive strategies show identical performance. The difference between the identifiability AUC between non-random to random strategies negatively correlates (Spearman correlation coefficient r) with the isolation score (**B**). Here, all human KEGG pathways are considered.

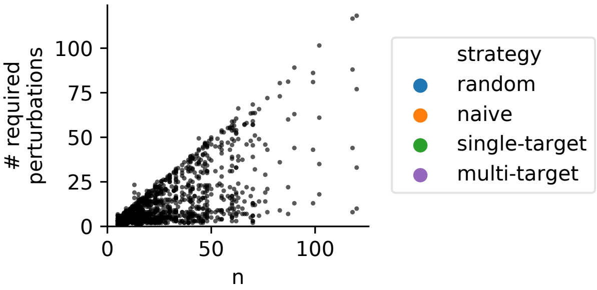

Figure S6: Number of perturbations required for full network identifiability for each considered KEGG pathway (of size n) and strategy. For the random strategy, we show the average number over 10 random perturbation sequences.

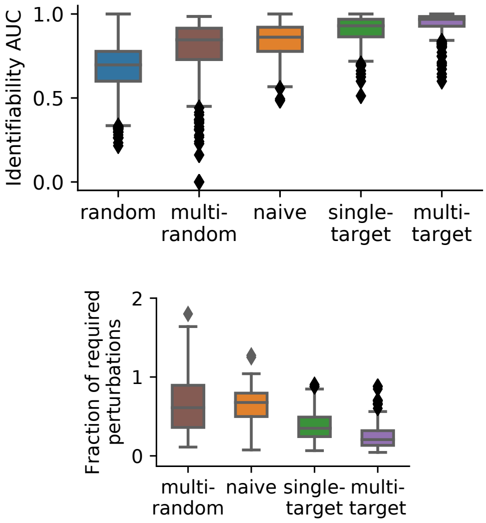

Figure S7: Identifiability AUC, defined as area under the number of identified nodes vs. number of perturbation curve, see Eqn. 13 in main text (top figure). Average number of perturbations required for full identifiability is shown relative to the average number required for a random strategy (bottom figure). Same data as in main text Figure 4 A and B with additional multi-random strategy.

strategy is approximately en par with the naive strategy but is outperformed by the single- and multi-target strategies.

S5 Verification by numerical simulation

To verify our analytical description of identifiability of network parameters, we numerically simulated the perturbation experiment depicted in Figure S3A. This was done by allocating random numbers to each network parameter, where all random numbers were drawn from a standard normal distribution. We then computed steady state responses to perturbations, R, according to Equation 4 (with $P = I_\text{p}$). From this synthetic data, we infer the original network parameters by solving the following least squares problem

$$\min_{J,S} \sum_{i}^{\text{n}} \sum_{j}^{\text{p}} \left(R_{ij} - \left[J^{-1}S\right]_{ij} \right)^2 ,$$

where only the unknown parameters in J and S are allowed to vary. To this end, we employ the least-squares solver from the SciPy library [15]. We repeated the procedure for 50 different sets of random network parameters. For each of these synthetic perturbation experiments, we perform the fitting with 50 different initial conditions generated by Latin Hypercube sampling within the interval -1 to 1. The absolute differences between the fitted and the original parameters are depicted in Figure S8A.

Each parameter that is declared identifiable by the maximum flow approach (see Figure S3A) shows indeed a near zero deviation. Whereas all non-identifiable parameters show considerable deviations. This confirms our analytical findings. It also shows that the numerical simulations are generally unreliable, as we observe many non-zero deviations for identifiable parameters and near-zero deviations for non-identifiable parameters. An identifiability analysis through numerical simulation thus relies on many repetitions and arbitrary thresholds, which also makes it computationally expensive and therefore inept for experimental design especially for larger systems.

Furthermore, we analysed the fit's sensitivity to noise. Random numbers drawn from normal distributions with zero mean and different standard deviations (Noise levels in Figure S8B) were added to each entry in the simulated R. The same fitting procedure was

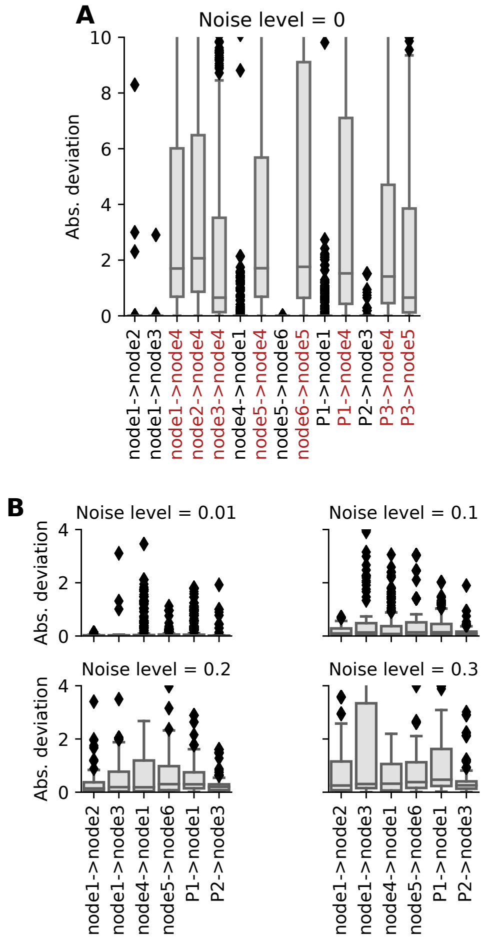

Figure S8: Absolute differences between inferred and original network parameters for the synthetic perturbation experiments depicted in Figure S3A. Shown are distributions over 50 random original parameter sets and 50 initial fitting conditions each. Network parameters that were declared non-identifiable by the maximum-flow approach are annotated in red. **A** Deviations for all network parameters in the absence of noise. Only identifiable parameters (according to maximum-flow approach, compare Figure S3A) have nearly zero deviations. **B** Deviations of identifiable network parameters with different levels of noise in the synthetic response data.

carried out for such noisy response data and the results are shown in Figure S8B (only identifiable parameters are shown). We observe that the median inference error for each parameter is approximately equal to the (additive) noise level. In individual fits however, some inferred parameters can drastically differ from their original counterpart. Yet, the fitting procedure is not the focus of this article and we refer to the reader to other references that aim to improve the robustness to noise [14, 5].

Numerical simulations also allow to investigate how the identifiability of network parameters is altered when assumptions in the maximum-flow approach are broken. For this purpose, we simulate saturation effects by setting some of the original network parameters to zero without considering them as known parameters. Therefore the identifiability analysis by the maximum-flow approach remains unchanged. However Figure S9 shows that indeed some previously identifiable parameters become non-identifiable and vice versa in such saturation setting. In detail, we performed noise-free numerical simulations of the perturbation experiment outlined in Figure S3A as before. However in Figure S9A and Figure S9B, we considered the possibilities that multi-target perturbation 3 is not effective with respect to either of its targets. Interestingly, Figure S9A shows that an ineffective perturbation of node 4 does not alter identifiability, including the fact that it remains impossible to infer from the response data that the sensitivity of node 4 to perturbation 3 is in fact zero. On the other hand, a zero sensitivity of node 5 to perturbation 4 is, contrary to the maximum-flow results, actually identifiable, as shown in Figure S9B (which is rather trivial as perturbation 3 no longer causes a response at node 5). However, this comes at the cost of losing identifiability of the interaction strength from node 5 to node 6. Similarly, we explored the loss of connectivity between nodes. Again, in our examples, we observed qualitatively different possibilities. While a vanishing interaction strength from node 2 to node 4 does not alter the identifiability of any network parameters (Figure S9C), we observe that the previously non-identifiable interaction strength from node 3 to node 4 becomes identifiable when it is set to zero (Figure S9D).

References

[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, Englewood Cliffs, N.J, 1993.

[2] Endre Boros, Khaled Elbassioni, Vladimir Gurvich, and Leonid Khachiyan. Algorithms for Enumerating Circuits in Matroids. In Toshihide Ibaraki, Naoki Katoh, and Hirotaka Ono, editors, *Algorithms and Computation*, Lecture Notes in Computer Science, pages 485–494, Berlin, Heidelberg, 2003. Springer.

[3] Evert Bosdriesz, Anirudh Prahallad, Bertram Klinger, Anja Sieber, Astrid Bosma, René Bernards, Nils Blüthgen, and Lodewyk F. A. Wessels. Comparative Network Reconstruction using mixed integer programming. *Bioinformatics*, 34(17):i997–i1004, January 2018.

[4] Riet De Smet and Kathleen Marchal. Advantages and limitations of current network inference methods. *Nat. Rev. Microbiol.*, 8(10):717–729, October 2010.

[5] Mathurin Dorel, Bertram Klinger, Torsten Gross, Anja Sieber, Anirudh Prahallad, Evert Bosdriesz,

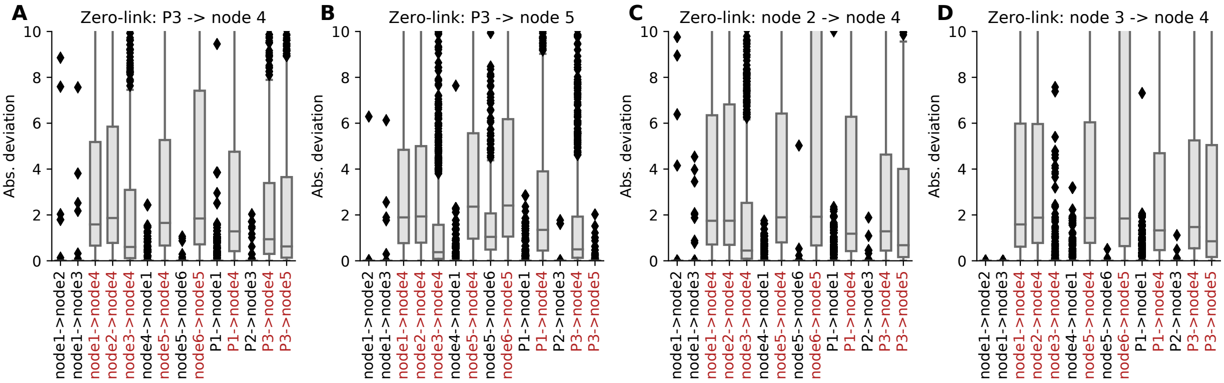

Figure S9: Absolute differences between inferred and original parameters for the same setting as in Figure S8, except that the parameters denoted in the titles of each subfigure are set to zero when simulating the response data. Again, network parameters that were declared non-identifiable by the maximum-flow approach are annotated in red.

Lodewyk F. A. Wessels, and Nils Blüthgen. Modelling signalling networks from perturbation data. *Bioinformatics*, 34(23):4079–4086, January 2018.

[6] Shimon Even and Guy Even. *Graph Algorithms*. Cambridge University Press, Cambridge, NY, 2nd ed edition, 2012.

[7] Raphaela Fritsche-Guenther, Franziska Witzel, Anja Sieber, Ricarda Herr, Nadine Schmidt, Sandra Braun, Tilman Brummer, Christine Sers, and Nils Blüthgen. Strong negative feedback from Erk to Raf confers robustness to MAPK signalling. *Molecular Systems Biology*, 7(1):489, January 2011.

[8] David Gale. Optimal assignments in an ordered set: An application of matroid theory. *Journal of Combinatorial Theory*, 4(2):176–180, March 1968.

[9] Torsten Gross, Matthew J. Wongchenko, Yibing Yan, and Nils Blüthgen. Robust network inference using response logic. *Bioinformatics*, 35(14):i634–i642, July 2019.

[10] Minoru Kanehisa, Yoko Sato, Miho Furumichi, Kanae Morishima, and Mao Tanabe. New approach for understanding genome variations in KEGG. *Nucleic Acids Res.*, 47(D1):D590–D595, January 2019.

[11] Karl Menger. Zur allgemeinen Kurventheorie. *Fundam. Math.*, 10(1):96–115, 1927.

[12] James Oxley. What is a Matroid? *CUBO Math. J.*, 5(3):176–215, October 2003.

[13] James G. Oxley. *Matroid Theory*. Number 3 in Oxford Graduate Texts in Mathematics. Oxford Univ. Press, Oxford, reprinted edition, 2006. OCLC: 255501879.

[14] Tapesh Santra, Oleksii Rukhlenko, Vadim Zhernovkov, and Boris N. Kholodenko. Reconstructing static and dynamic models of signaling pathways using Modular Response Analysis. *Current Opinion in Systems Biology*, 9:11–21, June 2018.

[15] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake Vand erPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1. 0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[16] Hassler Whitney. On the Abstract Properties of Linear Dependence. *American Journal of Mathematics*, 57(3):509, July 1935.

5 A TOTAL LEAST SQUARES APPROACH IMPROVES MRA OPTIMIZATION

5.1 ABSTRACT

Modular Response Analysis (MRA) is a network reconstruction method that is based on the assumption that a system's steady state responds linearly to a perturbation. It models this response by a set of interaction parameters that describe the influence of one system component on another. The computational task is to tune these parameters such that the predicted responses best match those that were experimentally observed. Generally, this is a complex non-linear optimization problem that is solved by gradient descent.

In the context of the MRA implementation that is proposed in the stasnet package (Dorel et al. 2018), it was observed that the optimization landscape contains many local optima. The pragmatic strategy to identify the global (or at least a satisfactory) optimum is to rerun the optimization multiple times (up to millions of repeats) with varying initial conditions. This gives rise to a computational bottleneck that currently limits the size of inferable networks. To this end, this chapter provides new results to improve the computational performance of MRA.

It can be shown that under a certain independence assumption, optimal MRA parameters form the solution of a set of total least squares (TLS) problems. These can be solved deterministically with a computational effort that is negligible in comparison to an iterative gradient descent optimization. However, the independence assumption breaks down under increasing levels of noise, such that the computed parameters no longer represent a network model that accurately describes the experimentally observed steady states responses. The retrieved parameter values nevertheless reside within the vicinity of the global optimum of the original optimization problem. This makes them ideal candidates for the initialization of a subsequent iterative optimization.

A benchmark on a range of toy networks generated from human pathway topologies shows that a TLS-based initialization drastically improves the optimization performance. In particular, it is shown to increase the fraction of optimization runs that reach the global optimum, while substantially decreasing the required number of iterations.

5.2 METHODS

The following provides a condensed summary of the derivation and solution of the TLS problem in MRA. The comprehensive description can be found in Appendices A to F.

A TLS solution to MRA has been proposed before (Andrec et al. 2005; Sontag 2008). However, previous formulations cannot incorporate prior network knowledge that fixes certain interaction parameters, and only describe the case where perturbations target each node individually. While these assumption considerably simplify the mathematical problem, they are rarely applicable in practice. Here, for the first time, a TLS solution is derived for the general case.

The MRA approach considers a system of n interacting components whose abundances, $\boldsymbol{x}$, evolve in time according to a set of (unknown) differential equations

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{p}). \tag{5.1}$$

The system can be experimentally manipulated in p independent ways, each of which is represented by one of the p components of parameter vector $\boldsymbol{p}$. The experimental set-up is assumed to exclusively allow for binary types of interventions, in which a particular type of perturbation can only be switched on or off. A perturbation experiment consists of q perturbations, each of which involves a single or a combination of perturbation types. It can thus be represented by a binary p × q matrix P. Each perturbation alters the system's steady state. It is assumed that we can experimentally observe the according steady state differences to the unperturbed state and collect them as columns of the n × q response matrix R. Assuming that perturbations are sufficiently mild as to cause a linear response, it can be shown that

$$R - E = -J^{-1} S\, P, \tag{5.2}$$

where $J_{ij} = \partial f_i / \partial x_j$ is the system's Jacobian, and $S_{ij} = \partial f_i / \partial p_j$ is the system's sensitivity matrix, both evaluated at the unperturbed steady state. Error matrix E accounts for the linear response approximation and measurement noise.

A Jacobian matrix entry J_{ij} quantifies the influence that the j-th system component exerts on the i-th component. Similarly, a sensitivity matrix entry S_{ij} quantifies the influence that the j-th type of perturbation exerts on the i-th component. These are the network characteristics that we hope to infer from the measurements of steady state difference. However, not all of the matrix entries are unknown. In line with prior studies (Kholodenko et al. 2002), the diagonal of the Jacobian matrix is fixed, $J_{ii} = -1$. Furthermore, we assume that we have prior knowledge about the network topology and the direct targets of the perturbations. This implies that there are zero entries in J whenever a component is not linked towards another component, and zero entries in S whenever a type of perturbation does not directly target a component.

The aim of MRA is to determine the remaining unknown parameters such that the (weighted) sum of squared residuals becomes minimal, that is

$$\underset{\text{unknown } J, S}{\text{minimize}} \sum_{i,j} \left[R + J^{-1} S\, P \right]_{ij}^{2} / \sigma_{ij}^{2}, \tag{5.3}$$

where the factor σ_{ij} quantifies the inverse weight that is allocated to the according response measurement, and could for example correspond to the standard deviation of the according measurement error. This optimization problem could then be solved by an iterative gradient descent based method that starts from some initial parameter configuration.

Alternatively, it is shown in Appendix A that Equation (5.2) can be rewritten as n linear systems in the unknown network parameters

$$\begin{bmatrix} \hat{R}^i - \hat{E}^i & \hat{P}^i \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{j}}^i \\ \hat{\boldsymbol{s}}^i \end{bmatrix} = -\left(\boldsymbol{k}^i - \boldsymbol{\epsilon}^i \right), \quad i \in \{1, 2, \ldots \mathrm{n}\}. \tag{5.4}$$

Vectors $\hat{\boldsymbol{j}}^i$ and $\hat{\boldsymbol{s}}^i$ denote the unknown parameters in the i-th row of J and S. Matrices $\hat{R}^i$, $\hat{P}^i$ and error matrix $\hat{E}^i$ are prior knowledge dependent submatrices of R, P, and E. Vector $\boldsymbol{k}^i$ and error vector $\boldsymbol{\epsilon}^i$ are prior knowledge dependent linear combinations of R and P, and E respectively. See Appendix A for detailed definitions.

The systems take the form of total least squares problems (Golub et al. 1980; Huffel et al. 1991; Schaffrin et al. 2008). However, depending on the configuration of prior knowledge, certain components of the error terms reappear in and thus effectively couple the n different linear systems. The error term is thus not independent, which means that a "closed-form solution ... may not exist" (Abatzoglou et al. 1991). To nevertheless allow for an efficient parameter inference, we shall first ignore the interdependence of the error terms and revisit this approximation in the next section. Specifically, we shall assume that all error terms in $\hat{E}^i$ and $\boldsymbol{\epsilon}^i$, $\forall i \in \{1, 2, \ldots \mathrm{n}\}$ are independent and identically normally distributed with zero mean. This decouples the equation systems such that the sets of unknown parameters $\hat{\boldsymbol{j}}^i$ and $\hat{\boldsymbol{s}}^i$ can be determined separately, as the solution to the following optimization problem

$$\begin{aligned} &\underset{\boldsymbol{\epsilon}^i\, \hat{E}^i}{\text{minimize}} \left\| \begin{bmatrix} \boldsymbol{\epsilon}^i & \hat{E}^i \end{bmatrix} \right\|_F, \\ &\text{subject to } \operatorname{rank}\left(\begin{bmatrix} \boldsymbol{k}^i - \boldsymbol{\epsilon}^i & \hat{R}^i - \hat{E}^i & \hat{P}^i \end{bmatrix} \right) = r_i, \end{aligned} \tag{5.5}$$

where the solution space dimension r_i is determined by the maximum flow through a prior knowledge dependent flow network, as illustrated in Figure 1 B of (Gross et al. 2020).

It is shown in Appendix A, that there is an analytical solution to this optimization problem, expressed by Equations (A.10) and (A.11). Its numerical evaluation requires one singular value decomposition and one QR decomposition, for which highly efficient deterministic algorithms are known (Anderson et al. 1999). In contrast to the iterative optimization of problem (5.3), these finish within a fixed number of steps.

However, the original assumption of identical noise levels for all entries in error matrix E is often not realistic. If replicate measurements are performed, noise levels can instead be estimated empirically, otherwise Appendix E provides an error model that accounts for a mixture of additive and multiplicative noise terms. Then, Appendices A and F show how to modify the TLS problem in order to account for non-identical noise levels.

5.3 RESULTS

The previous sections showed how a network reconstruction within the MRA framework can be solved as a TLS problem. This approach relied on the assumption that the set of linear systems (5.4) decouples, so that their sum of squared errors can be minimized independently of each other. However, as the same entries of the error matrix can appear in multiple of these systems, the solutions of the TLS problems are approximations that are expected to deteriorate with increasing noise levels. The question is how the TLS solutions compare to those that are found by solving optimization problem (5.3) with an iterative method. But before this can be addressed, initial parameter configurations have to be defined as the starting point of the iterative optimization. The stasnet package (Dorel et al. 2018) generates such initial conditions (ICs) by Latin hypercube sampling around parameter values of zero. Depending on the network, it was observed that many initializations do not converge or only find local optima of low quality. This requires a large number of repeated optimizations to increase the chances of convergence to the global optimum. For large systems this can be too computationally expensive.

Gradient-descent based optimization is known to perform better the closer the IC is to the optimum. Therefore, it could be beneficial to generate ICs within the neighbourhood of the TLS solution because, even if the TLS solution is erroneous due to the breakdown of the independence assumption, it might still be closer to the global optimum than ICs that are sampled from a Latin hypercube around the origin. In this section, we will therefore compare the performance of three different MRA optimization strategies, the TLS approach (non-iterative), an iterative optimization of Equation (5.3) with ICs drawn from a zero-centred Latin hypercube (iterative), and an iterative optimization with ICs drawn from a Latin hypercube that is centred around the TLS solution (iterative primed).

The different strategies were tested in simulated perturbation experiments on human KEGG pathway (Kanehisa et al. 2000) topologies with up to 30 nodes. Each of their (directed) edges is associated with an interaction strength drawn from a standard normal distribution. It is assumed that these networks can be perturbed at each node individually ($p = n$) with a perturbation sensitivity that is also drawn from a standard normal distribution. Perturbation responses are then simulated according to Equation (5.2), where the error matrix was generated according to the error model introduced in Appendix E with $\beta = 0.75$ and various magnitudes of error, ϵ. Given this (noisy) response matrix R, the goal is to reconstruct the network parameters using one of the three strategies. To this end, error rescaling was performed (see Appendix A) to account for the different noise levels as defined by the standard deviations generated by the noise model. Moreover, Latin hypercube sampling of ICs was carried out with a spread of two. Thus, initial parameter values were set to $x - 1$, x, or $x + 1$, where x is zero or the solution of the TLS problem, for the iterative or the primed iterative method, respectively. Iterative optimization was performed with a trust region reflective algorithm (Branch et al. 1999), as implemented in the SciPy library (Virtanen et al. 2020).

While the TLS approach is a deterministic algorithm that always delivers a solution, the iterative procedures are not guaranteed to converge. To compare the influence of the choice of ICs on convergence, network parameters were (iteratively) fitted 100 times and Figure 5.1a shows the fraction of optimization runs that converged after a maximum of $\sum_i^N v_i$ iterations, where $\sum_i^N v_i$ is the number of fitted parameters. This clearly indicates that while the iterative method does not converge most of the time, the large majority of primed iterative optimization runs find an optimum. In addition, Figure 5.1b shows that those runs that converge require fewer iterations if they were initialized around the TLS solution. This is a clear indication that the initialization around the TLS solution makes the iterative optimization more reliable and reduces computational effort.

However, the computational efficiency of the primed iterative method is only beneficial if it converges to the global optimum. Moreover, we need to investigate whether already the TLS solution alone reconstructs the network sufficiently well. Therefore, we consider a solu-

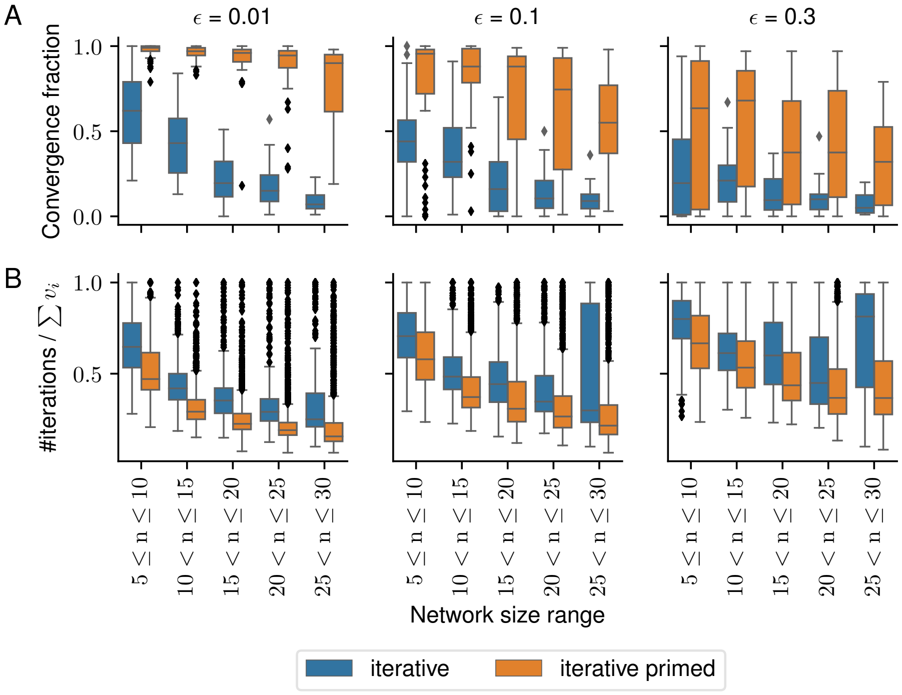

Figure 5.1: MRA network reconstruction for an ensemble of simulated perturbation experiments with different noise levels ϵ. Comparison of convergence behaviour of an iterative optimization where ICs were either generated by Latin hypercube sampling around the origin (iterative) or around the solution of the TLS problem (iterative primed). **A**: The fraction of 100 optimization runs that converged to an optimum. **B**: The number of iterations per fitted parameter needed to reach the optimum (for converged runs only).

tion's sum of squared residuals, recall Equation (5.3), normalized by number of data points

$$\langle e^2 \rangle = \frac{\sum_{i,j}^{n,p} \left[R + J^{-1} S\, P\right]_{ij}^2 / \sigma_{ij}^2}{\text{np}} \tag{5.6}$$

If we were to plug in the true parameter values into this formula, the term in square brackets would correspond to the according entry in the error matrix, which was drawn from a normal distribution. We would thus expect the entire numerator to be chi-square distributed with np degrees of freedom, so that $\langle e^2 \rangle$ would have a mean of one (and a slightly smaller median). This is therefore the smallest value that can be expected to be obtained without overfitting. And indeed, Figure 5.2a shows that this is what is found for the parameters that are obtained from both of the iterative optimization methods. We can thus conclude that both methods are capable of finding the global minimum, albeit with different computational performance as discussed before. In contrast, the TLS solution deteriorates as expected with increasing noise level and network size.

Furthermore, as the method performances is measured on synthetic data, we can compare the inferred (J and S) to the true parameters ($J^{\dagger}$

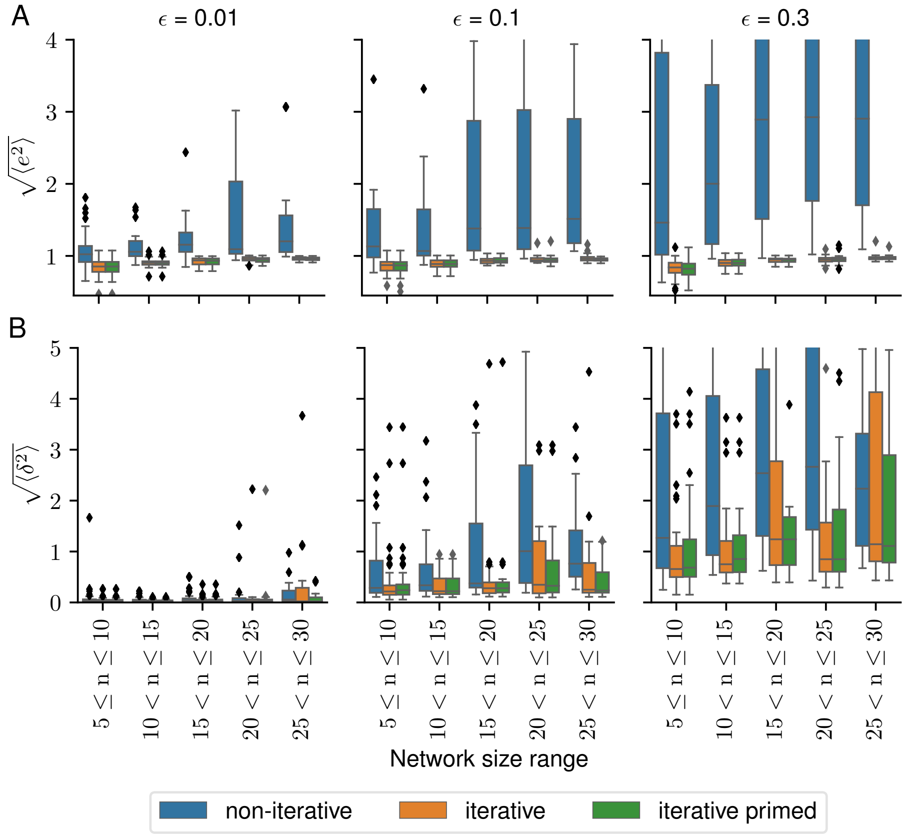

Figure 5.2: Comparison of the quality of inferred network parameters between the TLS solution, the iterative, and the TLS-primed iterative optimization. For each network, scores are computed for the (single) converged optimization run that minimizes $\langle e^2 \rangle$ (if any). **A**: The (square root) of the sum of squared residuals per data point, defined in Equation (5.6). **B**: The (square root) of the sum of squared parameter errors between inferred and true parameter values, normalized by number of parameters.

and $S^\dagger$) and define a sum of squared parameter errors, normalized by the number of inferred parameters

$$\langle \delta^2 \rangle = \left(\sum_{i,j \in \text{unknown}} (J_{ij} - J^\dagger_{ij})^2 + \sum_{i,j \in \text{unknown}} (S_{ij} - S^\dagger_{ij})^2 \right) / \sum_i^N v_i. \quad (5.7)$$

Even when the optimization finds the global optimum, it will not yield the true parameters exactly, due to the presence of noise. This is what can be observed in Figure 5.2b. If both iterative optimization methods converge to the global optimum, their solutions are expected to yield approximately the same error, which is in fact the case.

Notably, the parameter error has approximately the same magnitude as the noise. This seems plausible, yet, considering that the true parameters were drawn from a standard normal distribution, it points out some limits of MRA. For $\epsilon = 0.3$, which is a realistic noise level, for example in antibody-based phosphoprotein measurements, some of the inferred parameters can thus be considered inaccurate and even have the wrong sign. This the restricts the interpretability of MRA parameters. Yet, the impact of this issue and how to overcome it has already been studied in different contexts (Shayeghi et al. 2012; Klinger et al. 2018; Schulthess et al. 2011).

The error of the TLS solution is even larger and the solution is thus only acceptable at noise levels up to 0.1 or even below for larger networks. But this was expected due to the breakdown of the independence assumption at higher noise levels.

5.4 DISCUSSION

This chapter presented a numerical simulation of perturbation experiments which showed that iterative MRA optimization can be made more stable and less computationally expensive if its ICs are chosen within the vicinity of the solution of the TLS problem (5.5). The improved performance becomes especially beneficial when networks have to be refitted many times, as is the case in the network extension/reduction feature in the stasnet (Dorel et al. 2018) package. Here, network edges are added or removed from the network and the resulting change in residuals indicates whether a more appropriate topology was reached. As this requires many tests to sufficiently cover the space of topologies, the performance improvement by TLS initialization could become crucial. Likewise, a TLS initialization might become key to enable the inference of much bigger networks that are increasingly accessible as experimental perturbation techniques become more powerful (Adamson et al. 2016; Datlinger et al. 2017; Dixit et al. 2016; Jaitin et al. 2016; Schraivogel et al. 2020).

Yet, the presented results are preliminary and further steps are required to establish the benefits of the TLS initialization. An important concern is that so far only synthetic data has been analysed. It remains to be seen if the observed improvements withstand the non-linearities inherent to real data sets. Furthermore, the current TLS solution does not take into account two important features of the stasnet package. For one, this includes an extension of the MRA model to describe saturation effects, when inhibitions of non-active nodes do not trigger any measurable response. The other unaccounted feature is stasnets' capability to model nodes for which there are missing measurements or no measurements at all. Future research will need to assess if the TLS solution can also address these points while retaining its computational benefits.

6 CONCLUSION

Networks are indispensable descriptions of biological complexity. They are used to conceptualize transcriptional regulation (Sorrells et al. 2015), signal transduction (Janes et al. 2013), gene-disease relationships (Goh et al. 2007; Leiserson et al. 2015), intra-cellular interactions (Arnol et al. 2019), host-microbiota relationships (Guven-Maiorov et al. 2017), and many more biological systems (Alon 2003). A network perspective not only provides a coherent mental model of an intricate interplay of system components, it can also reveal the system's emergent properties (Bhalla et al. 1999) and allows to predict its response to perturbations or environmental changes (Hill et al. 2016). Unsurprisingly, it has become a common task in molecular biology to infer network models from experimental observations of the considered system.

For the past 20 years, a series of technological breakthroughs made it possible to detect or quantify an astounding array of cellular components in a high-throughput manner (Hasin et al. 2017). This unprecedented wealth of data has the potential to comprehensively explain the behaviour and mechanisms of complex cellular networks. But typically, 'Omics' technologies cannot directly elucidate interactions between the system's components. Network reconstruction methods therefore attempt to derive effective network models from observations of components' abundances or activities. Research on these methods has thrived since the onset of the 'Omics' revolution, and resulted in thousands of research articles (Jurman et al. 2019) about new approaches and applications. Nevertheless, the problem remains essentially unsolved (Saint-Antoine et al. 2020).

Depending on the available type of data and the goal of the analysis, different kinds of networks can be inferred. Observational data allows for the reconstruction of undirected networks to describe gene co-expression, functional associations of proteins, or amino-acid residue-residue contacts, amongst others (Stein et al. 2015). Mechanistic or causal relationships can be represented by directed networks. To be able to trace cause-effect relationships or information flow through the network, reconstruction methods typically rely on time-course or perturbation data to infer directed links. In many applications, obtaining sufficiently dense time samples is often excessively laborious or costly. Therefore, the focus of this thesis lies on the analysis of targeted perturbation data. Depending on context, such perturbations are performed, for example, by gene knockouts or kinase inhibitions, and the resulting changes of abundance or activity of the system components are measured. The data thus describes a global network response that is mediated by the aggregated effects of local interactions. Various methods attempt to derive these local interactions from global responses in a range of contexts (Wagner 2001; Bruggeman, Westerhoff, et al. 2002; Bonneau et al. 2006; Molinelli et al. 2013). Yet many of these approaches do not rigorously characterize network identifiability, cannot be contextualized, or return networks that are difficult to interpret. These issues motivated the development of the response logic approach (Gross et al. 2019), that was introduced in Chapter 3.

The response logic simply hypothesises that in a perturbation experiment, a node will only show a response if it is directly targeted by a perturbation or by another responding node. The objective of the method is then to identify the network topologies that imply the chains of perturbation responses that best fit to the (binarized) response data. As the number of possible topologies grows super-exponentially with network size, a powerful logic programming approach using Answer Set Programming (Gebser et al. 2014) is required to solve this hard combinatorial search problem. The response logic approach was tested in DREAM3 and DREAM4 (Marbach et al. 2010; Greenfield et al. 2010) community-wide reverse-engineering challenges, which provide synthetic perturbation data that can be compared against a known gold standard network and have become a de facto standard benchmark suite. In nearly all sub-challenges, it outperformed other participating methods. This indicates that the response logic approach is a suitable method to reconstruct biological networks.

One of the method's key features is to reveal all network topologies that conform to the data. It thereby characterizes network identifiability directly. Even if the solution set is too large to be enumerated, logic computations can nevertheless indicate whether a specific link is always present or absent (identifiable), or whether some conforming networks contain it whereas other do not (non-identifiable). This is different to most other methods that apply regularization techniques or sparsity constraints to single out one solution network (Wagner 2001; Bonneau et al. 2006), which is not necessarily the biologically relevant network. Instead, an explicit display of network identifiability avoids such bias and helps to select appropriate additional perturbations or external information to further constrain the solution space. Concerning the latter, the logic program formulation can flexibly incorporate prior network knowledge. This allows for adaptation to a biological context, for example by enforcing known links, or specifying restrictions on maximal or minimal numbers of connections in some part of the network.

Finally, the response logic approach stands out against methods for which inferred interactions are difficult to interpret. Often, methods are based on assumptions that are expressed in a mathematical form that is difficult to relate to a biological context. This impedes an intuitive understanding of how the data relates to the inferred network, which makes the method a "black box" whose suitability is hard to evaluate. In contrast, a link within the response logic setting solely indicates that a perturbation of a node is directly relayed to another node. While such a simple concept cannot capture details about the nature or scale of the interaction, it is applicable to and easily interpretable in a wide range of contexts. This also allowed the response logic approach to explain mutant-specific sensitivities to targeted drugs. Here, a perturbation experiment measured phosphorylation changes in kinases of the MAPK and PI3K signalling pathways in a colon cancer cell line, using a reverse-phase protein assay. The experiment was repeated on two different PI3K mutant cell lines, which enable the response logic approach to infer topological differences. In particular, both PI3K mutants seem to break the connection between the EGF receptor and the PI3K pathway, and one of them shows an EGF to EGFR feedback. These hypotheses helped to understand the mutant-specific growth differences in response to an array of targeted inhibitors.

Not always can biological systems be understood from their bare interaction topology. Sometimes, more gradual differences in a network can explain new phenotypes. A way to account for these more subtle aspects is to infer weighted networks. Weighted networks assign a continuous value to each edge to indicate a strength of interaction. For such a description to be meaningful, the weights need to be properly defined. A popular approach is thus to model a biological network as a system of ordinary differential equations. This allows to express interaction strengths as the entries of its Jacobian matrix. These values then describe the influence that one node has on the rate of change of another node. The goal of many reconstruction methods is to infer these entries from perturbation data (Gardner et al. 2003; Tegnér et al. 2003; Bonneau et al. 2006; Timme 2007). A well-established method amongst these is the Modular Response Analysis (MRA) (Bruggeman, Westerhoff, et al. 2002). It has been applied in a range of contexts (Santos et al. 2007; Stelniec-Klotz et al. 2012; Klinger et al. 2013; Halasz et al. 2016; Speth et al. 2017; Bosdriesz et al. 2018; Brandt et al. 2019; Jimenez-Dominguez et al. 2020) and its many extensions (Santra et al. 2018) turned it into an established tool of network inference. Yet until now, the identifiability of interaction strengths within the MRA formalism has not been sufficiently analysed. Some studies avoid the issue altogether by assuming extensive perturbations at each node that fully determine the system. Often, this is not realistic, because certain components cannot be experimentally targeted. Thus, other MRA approaches allow to relax these assumptions (Klinger et al. 2013;

Dorel et al. 2018) and study parameter identifiability with a profile likelihood approach (Raue et al. 2009). But this has the disadvantage that it requires to define thresholds that are not fail-proof and more importantly, it can only be applied after data has been collected. The analysis that is presented in Chapter 4 (Gross et al. 2020) addresses these challenges by showing analytically that identifiability can be understood as an intuitive maximum flow problem, which solely considers the structure of the network and the targets of the applied perturbations. In contrast to existing methods, the maximum flow perspective thus reveals identifiability without having to consider actual experimental data. This permits to systematically analyse any potential perturbation constellation and thereby to optimize experimental design.

Perturbation experiments are often highly laborious and costly, especially in phosphoproteomic studies for which MRA is commonly used. Therefore, improving experimental design and reducing the number of required perturbations is crucial for a comprehensive network analysis. To this end, we designed a depth-first search algorithm that builds on the maximum flow approach and determines from a set of possible perturbations those that maximally reduce the number of non-identifiable interaction strengths. We tested the approach on a set of KEGG topologies (Kanehisa et al. 2000). Provided that perturbations target a single node, the optimized experimental design reached full network identifiability with, on average, less than a third of the perturbations that would be needed in a random experimental design. If perturbations could also be combined to simultaneously target multiple nodes, this number could further be reduced to less than a quarter.

It is worth mentioning that MRA is based on a linearity assumption (see Equation (3) in Gross et al. 2020) that might not always correctly describe biological reality. Yet, this would not compromise the accuracy of the maximum flow approach, which merely reveals structural properties of such linear model. That being said, knowing that a parameter can be uniquely determined does not mean that its inferred value could be meaningfully interpreted, if model assumptions break. On a similar note, the maximum flow approach describes structural identifiability only (Bellman et al. 1970). Being agnostic to the actual data, it cannot assess whether a structurally identifiable parameter is practically non-determinable because of missing or noisy data. Nevertheless, the approach can help to design a structurally identifiable model, for which established methods can then handle practical non-identifiability (Raue et al. 2009; Dorel et al. 2018).

In summary, the maximum flow approach reliably reveals which network parameters are uniquely identifiable from a perturbation experiment, within an MRA context. As the approach builds on structural information only, it also allows to substantially optimize experimental design. This could considerably increase the scope of future perturbation experiments.

In the original formulations of MRA (Bruggeman, Westerhoff, et al. 2002; Kholodenko et al. 2002; de la Fuente, Brazhnik, et al. 2002), the Jacobian matrix could essentially be obtained from a simple inversion of the response matrix. However, to make the approach more applicable requires to account for prior network knowledge, unobserved nodes, measurement noise, and a limited number of perturbations, each of which could target multiple nodes. These requisites turn the inference of the Jacobian into a complex optimization problem. Current approaches (Klinger et al. 2013; Dorel et al. 2018) solve it with iterative methods that significantly increase the computational effort. In practice, interaction strengths can only be inferred in this way if the number of network nodes does not exceed the low tens. This calls for improvements that were presented in Chapter 5.

The derivation of the maximum flow problem showed that the MRA relationship between response data and network parameters can be described as a set of linear equation systems. These involve error terms to describe noise in the data. The goal is to determine network parameters that fulfil the equation systems while minimizing the weighted sum of squared errors, as described by Equation (5.5). It was possible to show in Chapter 5 that this leads to a set of total least squares (TLS) problems. This novel formulation of MRA can incorporate any prior network knowledge and any choice of perturbations. But generally, the different TLS problems are coupled, in which case closed-form solutions are not known. However, under an independence assumption these become available and can be obtained from matrix operations with negligible computational complexity compared to an iterative optimization. But with increasing noise levels the independence assumption breaks down and the inferred solutions become imprecise.Numerical experiments on synthetic response data generated for an ensemble of KEGG (Kanehisa et al. 2000) pathways indicate that the TLS solution becomes inaccurate for a signal to noise ratio of less than 10 (Figure 5.2). This makes the TLS solution unfit for typical experimental settings. Nevertheless, it is sufficiently close to the optimal solution to drastically improve the performance of an iterative optimization, when used as initial condition (IC). This was demonstrated in a comparison between the current standard procedure, which samples ICs from a Latin hypercube centred around the origin, and a sampling of ICs from a Latin hypercube centred around the TLS solution. Independent of the noise level, the fraction of optimization runs that converge to an optimum within a fixed number of iterations was drastically increased by using TLS-ICs (Figure 5.1a). Furthermore, the converging optimization runs needed a much smaller number of iterations to reach the optimum in the TLS-ICs case (Figure 5.1b). Therefore, even though the TLS solutions themselves are typically imprecise, they provide effective ICs for a subsequent iterative optimization. This makes the MRA optimization procedure more reliable and efficient, and thereby allows to infer larger networks. To substantiate these preliminary results, the next step will be to integrate the TLC-ICs into the stasnet (Dorel et al. 2018) package and to analyse their performance on real data sets.

More than two decades of research on network inference produced an impressive body of biological insights (Natale et al. 2017). Nevertheless, many questions are left unanswered (Saint-Antoine et al. 2020). This comes as no surprise.

> As biologists we often try to deduce the circuitry of modules by listing their component parts and determining how changing the input of the module affects its output. This reverse engineering is extremely difficult. Although an electrical engineer could design many different circuits that would amplify signals, he would find it difficult to deduce the circuit diagram of an unknown amplifier by correlating its outputs with its inputs. It is thus unlikely that we can deduce the circuity or a higher-level description of a module solely from genome-wide information about gene expression and physical interactions between proteins. Solving this problem is likely to require additional types of information and finding general principles that govern the structure and function of modules. (Hartwell et al. 1999)

But such "general principles" are sometimes unattainable from the reductionist perspective that is inherent to network reverse engineering (Anderson 1972; Gu et al. 2009). Sand patterns that emerge on a Chladni plate cannot be understood from a comprehensive characterization of the interactions between individual sand grains. Here, an effective description would rather deduce the rules that describe the form of the sand patterns as a function of the shape of the plate and the applied vibration frequency (Bizzarri et al. 2019). Conceptually, the idea here is to replace an enumeration of microscopic components and their individual interactions by a description of higher-order structures and overarching principles that govern them. Such top-down models were successfully applied in a range of biological systems (Pezzulo et al. 2016; Noble 2012). This includes the description of pattern formation in embryogenesis as a minimization of free-energy (Friston et al. 2015), understanding neural computations in terms of linear filtering and divisive normalization (Carandini 2012), or the decoding of cellular position by information theoretic constraints on morphogen gradients (Zagorski et al. 2017; Tkačik et al. 2015). However, such elegant, wholistic principles are often simply unavailable or imprecise, and the only available approach is indeed a bottom-up assembly of the cellular puzzle. In that regard, network reconstruction is an essential tool. Yet, its value crucially depends on whether inferred networks allow to derive meaningful biological interpretations, and whether the applied method clearly indicates which parts of the network can be uniquely determined from the data and which parts are non-identifiable. Achieving this was the main focus of this thesis. I believe that such emphasis on the methods' biological relevance is especially important as new generations of 'Omics' technologies (Adamson et al. 2016; Dixit et al. 2016; Jaitin et al. 2016; Datlinger et al. 2017; Schraivogel et al. 2020) expand the possibilities but also the complexity of biological network reconstruction.

APPENDICES

A A TOTAL LEAST SQUARES APPROACH TO MRA

Recap

To derive the total least squares (TLS) problem, I need to recapitulate a few of the key points from the study on "Identifiability and experimental design in perturbation studies" (Gross et al. 2020) that was introduced in Chapter 4.

The MRA approach considers a system of n interacting components whose abundances, $\boldsymbol{x}$, evolve in time according to a set of (unknown) differential equations

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{p}). \tag{A.1}$$

The system can be experimentally manipulated in p independent ways, each of which is represented by one of the p components of parameter vector $\boldsymbol{p}$. The experimental set-up is assumed to exclusively allow for binary types of interventions, in which a particular type of perturbation can only be switched on or off. A perturbation experiment consists of q perturbations, each of which involves a single or a combination of perturbation types. It can thus be represented by a binary p × q matrix P. Each perturbation alters the system's steady state. It is assumed that we can experimentally observe the according steady state differences to the unperturbed state and collect them as columns of the n × q response matrix R. Assuming that perturbations are sufficiently mild as to cause a linear response, it was shown in Equation 4 of Gross et al. 2020 that

$$R \approx -J^{-1} S\, P,$$

where

$$J_{ij} = \partial f_i / \partial x_j$$

is the system's Jacobian and

$$S_{ij} = \partial f_i / \partial p_j$$

the system's sensitivity matrix, both evaluated at the unperturbed steady state. This equation can be made exact, if we account for the error E that is due to the linear response approximation and measurement noise

$$R - E = -J^{-1} S\, P. \tag{A.2}$$

A Jacobian matrix entry J_{ij} quantifies the influence that the j-th system component exerts on the i-th component. Similarly, a sensitivity matrix entry S_{ij} quantifies the influence that the j-th type of perturbation exerts on the i-th component. These are the network characteristics that we hope to infer from the measurements of steady state difference. However, not all of the matrix entries are unknown. In line with prior studies (Kholodenko et al. 2002), the diagonal of the Jacobian matrix is fixed

$$J_{ii} = -1.$$

Furthermore, we assume that we have prior knowledge about the network topology and the direct targets of the perturbations. This implies that there are zero entries in J whenever a component is not linked towards another component, and zero entries in S whenever a type of perturbation does not directly target a component.

The aim of MRA is to determine the remaining unknown parameters such that the (weighted) sum of squared residuals becomes minimal, that is

$$\underset{\text{unknown } J, S}{\text{minimize}} \sum_{i,j} \left[R + J^{-1} S\, P \right]_{ij}^{2} / \sigma_{ij}^{2}, \tag{A.3}$$

where the factor σ_{ij} quantifies the inverse weight that is allocated to the according response measurement, and could for example correspond to the standard deviation of the according measurement noise. This optimization problem could then be solved by an iterative, gradient descent based method that starts from some initial parameter configuration.

Alternatively, Equation (A.2) can be rewritten as n linear systems

$$(R^T - E^T)\, \boldsymbol{j}^i = -P^T \boldsymbol{s}^i, \quad i \in \{1, 2, \dots \mathrm{n}\} \tag{A.4}$$

for each row in J and S, denoted as $\boldsymbol{j}^i$ and $\boldsymbol{s}^i$. We can then collect the $\bar{\mu}_i$ known entries of $\boldsymbol{j}^i$ in vector $\bar{\boldsymbol{j}}^i$ and the $\bar{\nu}_i$ known entries of $\boldsymbol{s}^i$ in vector $\bar{\boldsymbol{s}}^i$. Similarly, let us collect the remaining $\hat{\mu}_i = \mathrm{n} - \bar{\mu}_i$ unknown components of $\boldsymbol{j}^i$ in vector $\hat{\boldsymbol{j}}^i$, and the $\hat{\nu}_i = \mathrm{p} - \bar{\nu}_i$ unknown components of $\boldsymbol{s}^i$ in vector $\hat{\boldsymbol{s}}^i$. To rewrite Equation (A.4) as a linear system in those unknown variables, we first split its two sides into their known and unknown parts

$$\begin{aligned} \left(R^T - E^T\right) \boldsymbol{j}^i &= \left(\bar{R}^i - \bar{E}^i\right) \bar{\boldsymbol{j}}^i + \left(\hat{R}^i - \hat{E}^i\right) \hat{\boldsymbol{j}}^i \quad \text{and} \\ P^T \boldsymbol{s}^i &= \bar{P}^i \bar{\boldsymbol{s}}^i + \hat{P}^i \hat{\boldsymbol{s}}^i, \end{aligned}$$

where $\mathrm{q} \times \mathrm{z}_J^i$ matrices $\bar{R}^i$ and $\bar{E}^i$ consist of all columns of R^T or E^T with indices of known $\boldsymbol{j}^i$ elements, and $\mathrm{q} \times (\mathrm{n} - \mathrm{z}_J^i)$ matrices $\hat{R}^i$ and $\hat{E}^i$ collate the remaining columns of R^T or E^T. Analogously, $\mathrm{q} \times \mathrm{z}_S^i$

matrix $\bar{P}^i$ and $\mathrm{q} \times (\mathrm{p} - \mathrm{z}_S^i)$ matrix $\hat{P}^i$ are formed from the columns of P^T according to the known and unknown elements of $\boldsymbol{s}^i$. With abbreviations

$$\boldsymbol{k}^i = \begin{bmatrix} \bar{R}^i & \bar{P}^i \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{j}}^i \\ \bar{\boldsymbol{s}}^i \end{bmatrix} \quad \text{and} \quad \boldsymbol{\epsilon}^i = \bar{E}^i \, \bar{\boldsymbol{j}}^i,$$

we can then find a suitable equivalent formulation of Equation (A.4)

$$\begin{bmatrix} \hat{R}^i - \hat{E}^i & \hat{P}^i \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{j}}^i \\ \hat{\boldsymbol{s}}^i \end{bmatrix} = -\left(\boldsymbol{k}^i - \boldsymbol{\epsilon}^i\right), \quad i \in \{1, 2, \ldots \mathrm{n}\}, \tag{A.5}$$

which is a set of linear equations in the $\hat{\mu}_i + \hat{\nu}_i$ unknown parameters $\hat{\boldsymbol{j}}^i$ and $\hat{\boldsymbol{s}}^i$.

A central finding in Gross et al. 2020 is that

$$r_i \equiv \operatorname{rank}\left(\begin{bmatrix} \hat{R}^i - \hat{E}^i & \hat{P}^i \end{bmatrix}\right) = \hat{\nu}_i + y_i,$$

where y_i is the maximum flow between source and sink node in a specific flow network, as illustrated in Figure 1 B in Gross et al. 2020. It is there shown that this maximum flow does not depend on any unknown network parameters but can be uniquely determined from the knowledge about the network topology and the perturbation targets. Considering the sizes of the matrices involved in Equation (A.5), this implies the solution spaces of the linear systems to have dimensionality

$$\mathrm{d}_i = (\hat{\mu}_i + \hat{\nu}_i) - r_i = \hat{\mu}_i - y_i. \tag{A.6}$$

Total least squares problem

Equation (A.5) constitutes an ensemble of linear systems for the unknown parameters $\hat{\boldsymbol{j}}^i$ and $\hat{\boldsymbol{s}}^i$, $\forall i \in \{1, 2, \ldots \mathrm{n}\}$. However, the involved error terms are unknown. Therefore, we seek to solve the systems in a least squares sense, in which the identified solutions minimize the square sum of the error terms. However, depending on the configuration of prior knowledge certain components of the error terms reappear in and thus effectively couple the n different linear systems. The error term is thus not independent, which entails that a "closed-form solution . . . may not exist"(Abatzoglou et al. 1991). To nevertheless allow for an efficient parameter inference, we shall first ignore the interdependence of the error terms and revisit this approximation in Section 5.3. Specifically, we shall assume that each component in the error terms $\hat{E}^i$ and $\boldsymbol{\epsilon}^i$, $\forall i \in \{1, 2, \ldots \mathrm{n}\}$ is identically independently distributed with zero mean. This decouples the equation systems such that the sets of unknown parameters $\hat{\boldsymbol{j}}^i$ and $\hat{\boldsymbol{s}}^i$ can be determined separately.

Each of the according linear systems has errors in both the observation vector $\boldsymbol{k}^i$ and in the first columns of the data matrix $\begin{bmatrix}\hat{R}^i & \hat{P}^i\end{bmatrix}$. This is the form of a TLS problem (Golub et al. 1980; Huffel et al. 1991; Schaffrin et al. 2008) with a specific dimension of the solution space, given by Equation (A.6). To solve it, let us first reformulate Equation (A.5),

$$\left(\begin{bmatrix}\boldsymbol{k}^i & \hat{R}^i & \hat{P}^i\end{bmatrix} - \begin{bmatrix}\boldsymbol{\epsilon}^i & \hat{E}^i & 0\end{bmatrix}\right)\begin{bmatrix}1\\ \hat{\boldsymbol{j}}^i\\ \hat{\boldsymbol{s}}^i\end{bmatrix} = \mathbf{0}, \tag{A.7}$$

and state the appropriate objective

$$\begin{aligned}&\underset{\boldsymbol{\epsilon}^i\,\hat{E}^i}{\text{minimize}} \left\|\begin{bmatrix}\boldsymbol{\epsilon}^i & \hat{E}^i\end{bmatrix}\right\|_F,\\ &\text{subject to rank}\left(\begin{bmatrix}\boldsymbol{k}^i-\boldsymbol{\epsilon}^i & \hat{R}^i-\hat{E}^i & \hat{P}^i\end{bmatrix}\right) = r_i.\end{aligned} \tag{A.8}$$

In other words, we seek to find the nearest approximation to $\begin{bmatrix}\boldsymbol{k}^i\ \hat{R}^i\ \hat{P}^i\end{bmatrix}$ in terms of Frobenius norm, such that its rank becomes r_i, while keeping the $\hat{v}_i$ columns of $\hat{P}^i$ unchanged. The solution to this problem has been proposed in Golub et al. 1987 as follows.
For any $m \times l$ matrix X, with $m \geq l$, exists a singular value decomposition

$$\begin{aligned}&X = U\Sigma V^T, \text{ where}\\ &U^TU = I_m, \quad V^TV = I_l \quad \text{and}\\ &\Sigma^T = [\text{diag}(\sigma_1,\sigma_2,\ldots,\sigma_l) \quad 0_{l,m-l}], \text{ with } \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_l.\end{aligned}$$

Set

$$\hat{\Sigma}^T = [\text{diag}(\sigma_1,\sigma_2,\ldots,\sigma_k,0\ldots0) \quad 0_{l,m-l}].$$

According to the Eckart–Young–Mirsky theorem (Eckart et al. 1936),

$$\hat{X} = U\hat{\Sigma}V^T \equiv \mathcal{H}_k(X)$$

is a matrix with

$$\text{rank}(\hat{X}) \leq k, \quad \text{and} \quad \|\hat{X}-X\|_F = \inf_{\text{rank}(\bar{X})\leq \text{k}} \|\bar{X}-X\|_F.$$

Let $\mathcal{P}_i$ denote the orthogonal projector onto the column space of $\hat{P}^i$ and $\mathcal{P}_i^{\perp}$ the orthogonal projector onto its orthogonal complement. Then, the TLS-corrected data matrix

$$\begin{bmatrix}\boldsymbol{k}^i-\boldsymbol{\epsilon}^i & \hat{R}^i-\hat{E}^i\end{bmatrix} = \mathcal{P}_i\begin{bmatrix}\boldsymbol{k}^i & \hat{R}^i\end{bmatrix} + \mathcal{H}_{r_i-\hat{v}_i}\left(\mathcal{P}_i^{\perp}\begin{bmatrix}\boldsymbol{k}^i & \hat{R}^i\end{bmatrix}\right)$$

satisfies Equation (A.8). A small error in Golub et al.'s original formulation is discussed in Appendix B.

From the rank constraint in Equation (A.8) follows that there are

$$1 + \hat{\mu}_i + \hat{\nu}_i - r_i = 1 + d_i$$

distinct kernel basis vectors, which we collect in matrix K_D (dropping index i for convenience), as to write

$$\begin{bmatrix} \boldsymbol{k}^i - \boldsymbol{\epsilon}^i & \hat{R}^i - \hat{E}^i & \hat{P}^i \end{bmatrix} K_\mathrm{D} = 0_{\mathrm{q},d_i+1}. \tag{A.9}$$

We denote the first row of K_D as $\boldsymbol{K}_\mathbf{1}^T$ and that row's orthogonal complement as $K_1^\perp$, such that

$$K_\mathrm{D} = \begin{bmatrix} \boldsymbol{K}_\mathbf{1}^T \\ K_2 \end{bmatrix}, \quad \boldsymbol{K}_\mathbf{1}^T K_1^\perp = \mathbf{0}.$$

Comparing Equation (A.9) to Equation (A.7), we realize that we can express any element of the solution space as follows

$$\begin{aligned} \begin{bmatrix} 1 \\ \hat{\boldsymbol{j}}^i \\ \hat{\boldsymbol{s}}^i \end{bmatrix} &= K_\mathrm{D} \left(K_1^\perp \boldsymbol{w_i} + \boldsymbol{K_1} / |\boldsymbol{K_1}|^2 \right) \\ &= \begin{bmatrix} 0 \\ K_2 K_1^\perp \boldsymbol{w_i} \end{bmatrix} + \begin{bmatrix} 1 \\ K_2 \boldsymbol{K_1} / |\boldsymbol{K_1}|^2 \end{bmatrix}, \end{aligned}$$

with an undetermined vector $\boldsymbol{w_i}$ of length d_i. We abbreviate

$$\begin{bmatrix} \hat{\boldsymbol{j}}^i \\ \hat{\boldsymbol{s}}^i \end{bmatrix} \equiv \boldsymbol{\rho_i}, \quad K_2 K_1^\perp \equiv Q_i, \quad \text{and} \quad K_2 \boldsymbol{K_1} / |\boldsymbol{K_1}|^2 \equiv \boldsymbol{\rho_i^0},$$

and choose the kernel basis such that Q_i has orthonormal columns, so that

$$\boldsymbol{\rho_i} = Q_i \boldsymbol{w_i} + \boldsymbol{\rho_i^0}, \quad \text{and} \quad Q_i^T Q_i = I_{d_i}. \tag{A.10}$$

A procedure for the computation of Q_i and $\boldsymbol{\rho_i^0}$ can be found in Appendix C.

Summing up, under the assumption of independence of error terms, the ensemble of linear equations in (A.5) decouples, so that we can identify each solution space, with dimensionality stated in Equation (A.6), independently. This leads to the TLS problem (A.8), whose solution is found in Equation (A.10).

Minimum norm solution and parameter variance

In situations where a constellation of perturbations and prior knowledge does not fully determine the solution ($d_i > 0$), there might still be expectations on plausible parameter ranges that will restrict the solution space. Here, we propose a simple probabilistic assumption about such expectations and from it conclude that the most likely point in solution space coincides with the minimal norm solution. Furthermore, this will allow to quantify the degree to which a (non-trivial) solution space determines individual parameters.

It is a plausible assumption to avoid arbitrarily high parameter values but rather center them around zero. We could quantify this believe about the parameters $\boldsymbol{\rho_i}$ via an isotropic normal distribution of a certain spread σ_ρ,

$$\boldsymbol{\rho_i} \sim \mathcal{N}(0,\, \sigma_\rho^2\, I_{\hat{\mu}_i+\hat{\nu}_i}).$$

Yet, we know that $\boldsymbol{\rho_i}$ is bound to the solution space, specified in Equation (A.10). Still it is possible to identify the point in solution space with the highest probability according to the above distribution. To do so, we solve Equation (A.10) by $\boldsymbol{w_i}$,

$$\boldsymbol{w_i} = Q_i^T(\boldsymbol{\rho_i} - \boldsymbol{\rho_i^0}),$$

and interpret $\boldsymbol{\rho_i}$ as random variables under the above definition. Then, $\boldsymbol{w_i}$ will likewise be normally distributed with

$$\boldsymbol{w_i} \sim \mathcal{N}(-Q_i^T \boldsymbol{\rho_i^0},\, \sigma_\rho^2\, I_{d_i}).$$

This distribution maps the believe about the parameter distribution onto the solution space. Accordingly, we can find the most likely point in solution space by plugging the most likely $\boldsymbol{w_i}$, that is $\boldsymbol{w_i^*} = -Q_i^T \boldsymbol{\rho_i^0}$, into Equation (A.10), to obtain

$$\boldsymbol{\rho_i^*} = \left(I_{\hat{\mu}_i+\hat{\nu}_i} - Q_i\, Q_i^T\right) \boldsymbol{\rho_i^0}. \tag{A.11}$$

As can be seen in Appendix D, it is straightforward to show that $\boldsymbol{\rho_i^*}$ in fact constitutes the minimum norm solution

$$\min_{\boldsymbol{w_i}} \left|Q_i\, \boldsymbol{w_i} + \boldsymbol{\rho_i^0}\right| = \boldsymbol{\rho_i^*}.$$

While the derivation of the minimum norm solution does not require the presented probabilistic approach (as shown in Appendix D), it becomes useful to discuss the degree of determination of single parameters within the solution space. For example, if the solution space is (nearly) orthogonal to the j-th coordinate axis, different solutions that show a small difference in the j-th parameter will typically show large differences in the remaining parameters. Thus, the j-th parameter can be considered as nearly determined. To quantify this concept, let us return to the probabilistic approach and consider the variance of the j-th parameter,

$$\begin{aligned}\sigma_{\boldsymbol{w_i}}^2([\boldsymbol{\rho_i}]_j) &= \int \mathcal{P}(\boldsymbol{w_i}) \left([\boldsymbol{\rho_i^*}]_j - [\boldsymbol{\rho_i}]_j\right)^2 d\boldsymbol{w_i} \\ &= \int \mathcal{P}(\boldsymbol{w_i}) \left(\sum_k [Q_i]_{jk} \left([\boldsymbol{w_i}]_k + \sum_l \left[Q_i^T\right]_{kl} [\boldsymbol{\rho_i^0}]_l\right)\right)^2 d\boldsymbol{w_i},\end{aligned}$$

where $\mathcal{P}(\boldsymbol{w_i})$ represents the probability distribution of $\boldsymbol{w_i}$ as defined above. We can simplify the integral by a variable shift

$$\tilde{\boldsymbol{w}}_{\boldsymbol{i}} = \boldsymbol{w_i} + Q_i^T \boldsymbol{\rho}_{\boldsymbol{i}}^{\boldsymbol{0}} \;\Rightarrow\; \tilde{\boldsymbol{w}}_{\boldsymbol{i}} \sim \mathcal{N}(0,\, \sigma_\rho^2\, I_{d_i}),$$

and denote the shifted probability distribution as $\tilde{\mathcal{P}}(\tilde{\boldsymbol{w}}_{\boldsymbol{i}})$ to arrive at

$$\begin{aligned} \sigma^2_{\boldsymbol{w_i}}([\boldsymbol{\rho_i}]_j) &= \int \tilde{\mathcal{P}}(\tilde{\boldsymbol{w}}_{\boldsymbol{i}}) \underbrace{\left(\sum_k [Q_i]_{jk}\, [\tilde{\boldsymbol{w}}_{\boldsymbol{i}}]_k\right)^2}_{\sum_m \sum_n [Q_i]_{jm} [\tilde{\boldsymbol{w}}_{\boldsymbol{i}}]_m [Q_i]_{jn} [\tilde{\boldsymbol{w}}_{\boldsymbol{i}}]_n} d\tilde{\boldsymbol{w}}_{\boldsymbol{i}} \\ &= \sum_m \sum_n [Q_i]_{jm}\, [Q_i]_{jn} \int \tilde{\mathcal{P}}(\tilde{\boldsymbol{w}}_{\boldsymbol{i}})\; [\tilde{\boldsymbol{w}}_{\boldsymbol{i}}]_m\, [\tilde{\boldsymbol{w}}_{\boldsymbol{i}}]_n\, d\tilde{\boldsymbol{w}}_{\boldsymbol{i}} \\ &= \sigma_\rho^2 \left[Q_i Q_i^T\right]_{jj}. \end{aligned} \tag{A.12}$$

Thus, each parameter's relative variance, $\sigma^2_{\boldsymbol{w_i}}([\boldsymbol{\rho_i}]_j)/\sigma_\rho^2$, within the probabilistic solution space, as defined above, can be computed from the diagonal of the data-derived projector $Q_i Q_i^T$. The degree of determination of a particular parameter becomes higher with decreasing relative variance. In view of the mean of the relative variances,

$$\frac{1}{\hat{\mu}_i + \hat{\nu}_i} \sum_{j=1}^{\hat{\mu}_i+\hat{\nu}_i} \left[Q_i Q_i^T\right]_{jj} = \frac{d_i}{\hat{\mu}_i + \hat{\nu}_i} < 1,$$

we could consider a parameter $[\boldsymbol{\rho_i}]_j$ as approximately identifiable if

$$\left[Q_i Q_i^T\right]_{jj} \ll 1.$$

It is important to stress that this is not a strict criteria. Even if a parameter has a very small but non-zero relative variance it is still structurally non-identifiable, and would be characterized as such by the maximum flow approach presented in Chapter 4. The relative variance has to be interpreted in a statistical sense under the assumption that parameters are sampled from a centred normal distribution, as discussed above. Moreover, the relative variance is also not a measure of parameter uncertainty due to measurement noise. Even in the absence of noise, a parameter can have a large relative variance because the solution space is far from orthogonal to the according coordinate axis, and vice versa.

Altogether, assumptions about reasonable parameter ranges allow to identify the most appropriate parameter vector within the solution space, as described by Equation (A.11), and to quantify each parameter's degree of determination, as specified in Equation (A.12). In the following, we refer to this minimum norm parameter vector as the solution to the ensemble of TLS problems.

Error model and error scaling

A TLS regression as in Equation (A.8) finds a rank deficient approximation of a given input matrix whose difference is minimal in Frobenius norm. In this way, the deviations from different matrix entries all contribute equally to the sum of squares. Facing measurement error, this might be undesirable if the uncertainty about the values of different components of the input matrix varies considerably. Preferably, the rank deficient approximation will differ mostly in entries with a relatively high noise level and show little deviations for more precisely known entries. For that purpose, we will introduce error scaling matrices which rescale the response matrix in order to achieve homoscedasticity. Yet, the derivation of such scaling matrices requires a quantification of measurement error for each entry of the response matrix. These noise levels could be estimated from the variance of replicate measurements. However, often there are no or few replicate measurements so that it is not possible to estimate them from the data directly. To this end, we introduce a simple, two parameter error model in Appendix E. For each entry of the response matrix, R_{ij}, it specifies measurement noise as a normal distribution with zero mean and a specific standard deviation $[\sigma_R]_{ij}$. Considering that the response matrix describes steady state differences, the error model includes the contribution of noise from measurements of the unperturbed as well as the perturbed steady states, see Equation (E.2). It describes a mix between additive and multiplicative noise whose relative contributions can be tuned with a free parameter β, and has an overall error magnitude that is determined by an additional free parameter ϵ, as defined in Equation (E.1).

As discussed above, non-uniformity in σ_R will distort the solution of the TLS problem (A.8). A straightforward way to compensate for this, is to rescale the response matrix prior to the regression such that the associated standard deviations of all its entries are equal. Subsequently, we can apply an appropriate reverse scaling to the solution of the rescaled TLS problem to obtain the solution of the original problem. More precisely, we shall assume that there are diagonal row- and column scaling matrices $T = \mathrm{diag}(\mathbf{t})$ and $C = \mathrm{diag}(\mathbf{c})$, for which

$$\frac{[\sigma_R]_{ij}}{t_i\, c_j} \approx 1. \tag{A.13}$$

We can then define rescaled matrices

$$\begin{aligned}
&\tilde{R} = T^{-1}\,R\,C^{-1}\\
&\tilde{E} = T^{-1}\,E\,C^{-1}\\
&\tilde{J} = J\,T\\
&\tilde{S} = S\\
&\tilde{P} = P\,C^{-1},
\end{aligned}$$

for which Equation (A.2) holds in identical form

$$\tilde{R} - \tilde{E} = -\tilde{J}^{-1}\tilde{S}\,\tilde{P}.$$

The difference is that we now assume the entries of $\tilde{E}$ to be homoscedastic. We can then carry out the error-scaling corrected parameter inference in completely analogous form by replacing the matrices with their rescaled counterparts, and eventually scaling back (essentially, $J = \tilde{J}\,T^{-1}$). However, this requires scaling matrices T and C, which we shall derive in the following.

There are $\mathrm{n} \cdot \mathrm{q}$ components in E versus only $\mathrm{n} + \mathrm{q}$ components in $\mathbf{t}$ and $\mathbf{c}$. We can thus not expect a perfect error-scaling but will rather attempt to find $\mathbf{t}$ and $\mathbf{c}$ to maximally approach homoscedasticity of $\tilde{E}$. More formally, referring to Equation (A.13), we shall attempt to

$$\begin{aligned}
&\underset{\mathbf{t}\,\mathbf{c}}{\text{minimize}} \sum_{i=1}^{\mathrm{n}} \sum_{j=1}^{\mathrm{q}} \left(t_i\,c_j - [\sigma_R]_{ij}\right)^2,\\
&\text{subject to}\ \ t_i > 0,\ c_j > 0 \quad \forall\, i \in \{1 \ldots \mathrm{n}\},\ j \in \{1 \ldots \mathrm{q}\}.
\end{aligned} \tag{A.14}$$

We will first ignore the positivity constraint and identify $\boldsymbol{t}^*$ and $\boldsymbol{c}^*$, the extreme points of the objective function, by setting to zero its partial derivatives $\partial/\partial t_i$ and $\partial/\partial c_i$. Obviously the objective function remains constant if a scaling of vector $\boldsymbol{t}^*$ is compensated with the inverse scaling of $\boldsymbol{c}^*$. To make the extreme points unique we can thus set $|\boldsymbol{t}^*|^2 = 1$. Taken together, this produces two equations

$$\begin{aligned}
&\sigma_R\ \boldsymbol{c}^* = |\boldsymbol{c}^*|^2\,\boldsymbol{t}^*\\
&\sigma_R^T\ \boldsymbol{t}^* = \boldsymbol{c}^*,
\end{aligned}$$

that can be combined into an eigenvalue problem

$$\sigma_R\ \sigma_R^T\ \boldsymbol{t}^* = |\boldsymbol{c}^*|^2\ \boldsymbol{t}^*.$$

The $\mathrm{n} \times \mathrm{n}$ matrix $\sigma_R\ \sigma_R^T$ is real and symmetric and thus has n real eigenvalues with associated eigenvectors. To determine which of these solves the minimization problem, we can use the previous equations to determine the value of the objective function for the possible eigenvectors.

$$\begin{aligned}\sum_{i=1}^{n}\sum_{j=1}^{q}\left(t_i^* \, c_j^* - [\sigma_R]_{ij}\right)^2 &= \operatorname{tr}\left(\left(\boldsymbol{c}^* \, (\boldsymbol{t}^*)^T - \sigma_R^T\right)\left(\boldsymbol{t}^* \, (\boldsymbol{c}^*)^T - \sigma_R\right)\right) \\ &= \operatorname{tr}\left(\sigma_R^T \sigma_R - \boldsymbol{c}^* (\boldsymbol{c}^*)^T\right) \\ &= \|\sigma_R\|_F - |\boldsymbol{c}^*|^2\end{aligned}$$

This shows that the global minimum is found by choosing the eigenvector with the largest eigenvalue $|\boldsymbol{c}^*|^2$. Furthermore, standard deviations of the response matrix are strictly positive, such that also $\sigma_R \, \sigma_R^T$ is a positive matrix. In that case the Perron–Frobenius theorem (Perron 1907) holds, which states that a positive matrix has one real largest eigenvalue and its associated eigenvector is unique (up to scaling) and only has positive components. From this we can conclude that the optimal $\boldsymbol{t}^*$ (and the according $\boldsymbol{c}^*$) that satisfies Equation (A.14), including the positivity constraint, is given by the (unique) eigenvector of $\sigma_R \, \sigma_R^T$ that is associated with the largest eigenvalue. This transforms the constrained least squares optimization in (A.14) into a simple computation of eigenvectors, for which efficient algorithms exist (Anderson et al. 1999), especially because $\sigma_R \, \sigma_R^T$ is symmetric. Appendix F assesses the degree to which the derived error-scaling matrices can establish homoscedasticity for different classes of random matrices.

B ERRATUM TO: A GENERALIZATION OF THE ECKART-YOUNG-MIRSKY MATRIX APPROXIMATION THEOREM (GOLUB ET AL., 1987)

In their paper (Golub et al. 1987) about a generalization of the Eckart-Young-Mirsky matrix approximation theorem, Golub et al. show "how to obtain a best approximation of lower rank in which a specified set of columns of the matrix remains fixed". Their main result is expressed in the theorem on page 319:

THEOREM*. Let X be partitioned as in (1.8) where X_1 has k columns, and let $l = rank(X_1)$. Let P denote the orthogonal projection onto the column space of X and $P^\perp$ the orthogonal projection onto its orthogonal complement. If*

$$l \leq r,$$

then the matrix

$$\hat{X}_2 = PX_2 + H_{r-l}(P^\perp X_2)$$

satisfies (1.9).

A small error in the definition of P and $P^{\perp}$ render the theorem incorrect. A corrected definition should read:

> *Let P denote the orthogonal projection onto the column space of X_1 and $P^{\perp}$ the orthogonal projection onto its orthogonal complement.*

That is, X should be replaced by X_1. To justify the correction let us write down the corrected operator definitions. To do so, we make use of the QR decomposition of X as stated in the paper's Equation (2.3). Due to the upper triangular shape of R, the first k columns of X, that is X_1, are solely expressed as linear combinations of the first k columns of Q. Thus, as X_1 has full rank the first k columns of Q, that is Q_1, form an orthonormal basis of X_1 and the corrected projection operators write

$$\begin{aligned} P_{\text{corr}} &= Q_1 Q_1^T \\ P_{\text{corr}}^{\perp} &= \begin{bmatrix} Q_2 & Q_3 \end{bmatrix} \begin{bmatrix} Q_2 & Q_3 \end{bmatrix}^T . \end{aligned}$$

Let us apply these definitions to the final statement in the proof to the above theorem.

> *It now remains only to observe that*
>
> $$Q_1 R_{12} = P X_2 \quad \textit{and} \quad Q_2 R_{22} = P^{\perp} X_2. \qquad \blacksquare$$

Making use of the orthogonality of Q and extracting the expression of X_2 from the paper's Equation (2.3) one can easily see that the statement holds for the corrected definitions P_{corr} and $P_{\text{corr}}^{\perp}$. In contrast, the original P constituted a projection onto the column space of (all of) X. The columns of X_2 are obviously elements of this space. Therefore, by the definition of a projection this amounts to

$$P X_2 = X_2 = Q_1 R_{12} + Q_2 R_{22} \text{ and } P^{\perp} X_2 = 0,$$

which would only coincides with the above statement in the trivial case of

$$\begin{aligned} & Q_2 R_{22} = 0 \\ \Leftrightarrow\ & \operatorname{rank}(X) = \operatorname{rank}(X_1). \end{aligned}$$

C COMPUTING AN ORTHONORMAL SOLUTION SPACE BASIS

Here we present an algorithm to compute Q_i and $\boldsymbol{\rho}_i^0$, as used in Equation (A.10). For convenience, we will not always include index i in the notation.

1. Compute the singular value decomposition of the data matrix that solves the total least squares problem (A.8)

$$\begin{bmatrix} \boldsymbol{k}^i - \boldsymbol{\epsilon}^i & \hat{R}^i - \hat{E}_R^i & \hat{P}^i \end{bmatrix} = U \Sigma V^T.$$

The last $d_i + 1$ columns of V, denoted as K_D, form a basis of the data matrix's kernel, and thus $[1 \quad \boldsymbol{\rho}_i^T]^T \in \text{range}(K_D)$. Let us partition them as

$$K_D = \begin{bmatrix} \boldsymbol{K_1}^T \\ K_2 \end{bmatrix}.$$

2. Compute the QR-decomposition of vector $\boldsymbol{K_1}$ to obtain its orthogonal complement $K_1^\perp$,

$$\boldsymbol{K_1} = [\tilde{\boldsymbol{K}}_{\boldsymbol{1}} \quad K_1^\perp] \begin{bmatrix} |\boldsymbol{K_1}| \\ 0 \\ \vdots \end{bmatrix}.$$

where $[\tilde{\boldsymbol{K}}_{\boldsymbol{1}} \quad K_1^\perp]$ is orthogonal.

3. Compute

$$\begin{aligned} \boldsymbol{\rho}_i^0 &= K_2 \tilde{\boldsymbol{K}}_{\boldsymbol{1}} / |\boldsymbol{K_1}| \\ \tilde{Q}_i &= K_2 K_1^\perp. \end{aligned}$$

4. The columns of $\tilde{Q}_i$ are not orthonormal. But we show below that $\text{rank}(\tilde{Q}_i) = d_i$. Thus, compute a QR-decomposition of $\tilde{Q}_i$ and find Q_i as the first d_i columns of the resulting orthogonal matrix.

Since $\text{rank}(K_D) = d_i + 1$, there are two possibilities

$$\text{rank}(K_2) = \begin{cases} d_i + 1, & \text{if } \boldsymbol{K_1} \in \text{range}(K_2^T) \\ d_i, & \text{if } \boldsymbol{K_1} \notin \text{range}(K_2^T). \end{cases}$$

Recalling that $K_1^\perp$ has dimensions $(d_i + 1) \times d_i$, it is obvious that

$$\text{rank}(\tilde{Q}_i) = d_i,$$

in the first case. The second case implies that

$$\text{range}(K_2^T) = \text{range}(K_1^\perp),$$

so that we can write

$$K_2^T = K_1^\perp Z,$$

with any matrix Z, with $\text{rank}(Z) = d_i$

$$\tilde{Q}_i = Z^T \left(K_1^\perp\right)^T K_1^\perp = Z^T,$$

makes obvious that $\text{rank}(\tilde{Q}_i) = d_i$, also in the second case.

D DERIVING THE MINIMUM NORM SOLUTION

Section A identified a particular solution, $\boldsymbol{\rho}_{\boldsymbol{i}}^{*}$ from Equation (A.11) within the solution space described by Equation (A.10). Here we show that $\boldsymbol{\rho}_{\boldsymbol{i}}^{*}$ is in fact the minimal norm solution, by directly minimizing the norm of the solution vector. That is, we want to show that

$$\min_{\boldsymbol{w_i}} \left| Q_i \, \boldsymbol{w_i} + \boldsymbol{\rho}_{\boldsymbol{i}}^{\mathbf{0}} \right| = \boldsymbol{\rho}_{\boldsymbol{i}}^{*}.$$

Thus, let us analyse the extrema of the solution norm

$$\begin{aligned}
0 &= \left.\frac{\partial \boldsymbol{\rho}_{\boldsymbol{i}}^{T} \boldsymbol{\rho_i}}{\partial [\boldsymbol{w_i}]_j}\right|_{\boldsymbol{w}_{\boldsymbol{i}}^{*}} \\
&= \left.\frac{\partial \boldsymbol{w}_{\boldsymbol{i}}^{2}}{\partial [\boldsymbol{w_i}]_j}\right|_{\boldsymbol{w}_{\boldsymbol{i}}^{*}} + 2 \left.\frac{\partial \, \boldsymbol{w}_{\boldsymbol{i}}^{T} Q_i^T \boldsymbol{\rho}_{\boldsymbol{i}}^{\mathbf{0}}}{\partial [\boldsymbol{w_i}]_j}\right|_{\boldsymbol{w}_{\boldsymbol{i}}^{*}} \\
&= 2 \left([\boldsymbol{w}_{\boldsymbol{i}}^{*}]_j + \sum_{k=1}^{v} [Q_i^T]_{jk} [\boldsymbol{\rho}_{\boldsymbol{i}}^{\mathbf{0}}]_k \right) \\
\Leftrightarrow \boldsymbol{w}_{\boldsymbol{i}}^{*} &= -Q_i^T \boldsymbol{\rho}_{\boldsymbol{i}}^{\mathbf{0}} \\
\Leftrightarrow \boldsymbol{\rho}_{\boldsymbol{i}}^{*} &= \left(I_{\hat{\mu}_i + \hat{v}_i} - Q_i \, Q_i^T \right) \boldsymbol{\rho}_{\boldsymbol{i}}^{\mathbf{0}}.
\end{aligned}$$

This coincides with the statement in Equation (A.11).

E AN ERROR MODEL FOR THE RESPONSE MATRIX

A perturbation experiment yields read-outs for n unperturbed network nodes and n nodes under q different perturbations and each of which could have been measured in replicates. Thus, let U_i^r and V_{ij}^r denote the unperturbed and the perturbed read-outs of node $i \in \{1 \dots \mathrm{n}\}$, perturbation setting $j \in \{1 \dots \mathrm{q}\}$, and replicate $r \in \{1 \dots \mathrm{r}_i^U \text{ or } \mathrm{r}_{ij}^V\}$. We shall define an error model that assumes measurement error for some read-out x to be independently sampled from a Gaussian distribution with zero mean and standard deviation

$$\sigma(x) = \epsilon \left(\beta \, |\langle x \rangle| + (1 - \beta) \bar{x} \right), \tag{E.1}$$

where ϵ and β are free parameters, $\langle x \rangle$ is the mean taken over all replicates of x, that is

$$\langle U_i^r \rangle \equiv U_i = \frac{1}{\mathrm{r}_i^U} \sum_{r=1}^{\mathrm{r}_i^U} U_i^r \quad \text{and} \quad \langle V_{ij}^r \rangle \equiv V_{ij} = \frac{1}{\mathrm{r}_{ij}^V} \sum_{r=1}^{\mathrm{r}_{ij}^V} V_{ij}^r,$$

and where $\bar{x}$ quantifies a typical absolute read-out value within the perturbation experiment, e.g.

$$\bar{x} = \frac{1}{2} \left(\frac{1}{\mathrm{n}} \sum_{i=1}^{\mathrm{n}} |U_i| + \frac{1}{\mathrm{n}\,\mathrm{q}} \sum_{i=1}^{\mathrm{n}} \sum_{j=1}^{\mathrm{q}} |V_{ij}| \right).$$

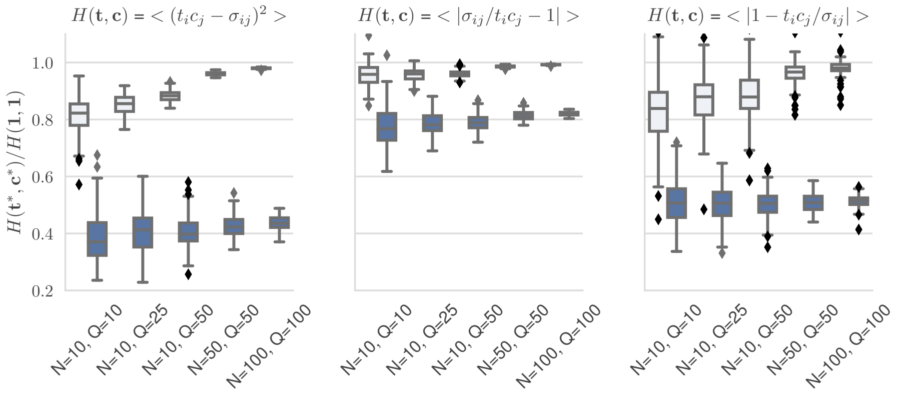

Figure F.1: 100 random matrices (see text) are rescaled (by row and column factors $\boldsymbol{t}^*$ and $\boldsymbol{c}^*$) according to Equation (A.14) to increase homoscedasticity. Each panel displays a different measure of homoscedasticity ($H(\boldsymbol{t}, \boldsymbol{c}) = 0 \Leftrightarrow$ perfect homoscedasticity) for which its ratio distribution of scaled to unscaled ($\boldsymbol{t} = \mathbf{1}$, $\boldsymbol{c} = \mathbf{1}$) matrices are shown. Matrix entries were either drawn from a folded normal distribution (grey boxes) or matrices are a mix of folded normal and structured entries (blue boxes), as detailed in text.

Thus, errors for replicates are identically distributed with a standard deviation that scales with error magnitude ϵ and that consists of a relative and an absolute term, with ratio β.

As expected, this implies that replicate measurements reduce the uncertain of the mean,

$$\sigma(U_i) = \frac{\sigma(U_i^r)}{\sqrt{\mathrm{r}_i^U}} \quad \text{and} \quad \sigma(V_{ij}) = \frac{\sigma(V_{ij}^r)}{\sqrt{\mathrm{r}_{ij}^V}}.$$

Thus, given β and ϵ the error model allows to specify the variances of the entries of the response matrix

$$\begin{aligned} R_{ij} &= \langle V_{ij} \rangle - \langle U_i \rangle \\ [\sigma_R]_{ij}^2 &= \sigma\left(V_{ij}\right)^2 + \sigma\left(U_i\right)^2. \end{aligned} \tag{E.2}$$

F EXAMINATION OF HOMOSCEDASTICITY AFTER ERROR-SCALING

The measure of homoscedasticity of the rescaled response matrix is not unique. While the one chosen in Equation (A.14) is convenient, we can evaluate the quality of the data rescaling in different ways. We do so by generating random matrices σ_R, rescaling them and computing homoscedasticity scores, see Figure F.1. As homoscedasticity measures, we choose in the left panel the one from Equation (A.14), the middle panel shows the mean deviation from unit standard deviation, whereas the relative mean deviation from the unit standard deviation,

$$1/(\mathrm{n}\,\mathrm{q}) \sum_{i=1}^{\mathrm{n}} \sum_{j=1}^{\mathrm{q}} |(\sigma_{ij}/t_i c_j - 1)/(\sigma_{ij}/t_i c_j)| = \langle |1 - t_i c_j/\sigma_{ij}| \rangle,$$

is illustrated in the right panel. All measures are positive and quantify the divergence from perfect homoscedasticity, for which they become zero. To make ratios of measures between scaled and unscaled matrices meaningful, the mean of the entries of the random matrices is 1. Consequently, if the rescaling increases homoscedasticity, the ratio will be between zero and one.

We evaluated two scenarios. First, random matrix entries were drawn from a folded normal distribution. Here, on average homoscedasticity could be increased independently of the considered homoscedasticity measure, but not in every case and the increase remains rather small. However, this is a particularly difficult scenario as matrix entries are independent of each other. When standard deviation are estimated from actual data sets, we might expect to find a more eminent structure. If the typical magnitude of certain read-outs differs considerably from that of other read-outs and there is relative error, the rows of σ_R will scale accordingly (the same holds for columns if certain perturbations are more potent than others). Then, we expect to be able to achieve homoscedasticity more easily. To model such a scenario we consider the sum of folded normal matrix with entry mean 0.5 and a structured matrix $\tilde{t}\,\tilde{c}^T$ with entry mean 0.5 (such that the sum has entry mean 1), where the entries of $\tilde{t}$ and $\tilde{c}$ are drawn from a folded normal distribution. As expected the rescaling of this more structured matrix increases homoscedasticity more drastically and in every case. For both scenarios we observe that increasing homoscedasticity becomes more difficult for larger matrices. This is to be expected as the ration between the number of scaling factors $n + q$ to the number of matrix entries $n \cdot q$ becomes smaller.

In summary, we established a two parameter error model that provides standard deviations for the entries of the response matrix. To account for them in the total least squares regression (A.8), we identified scaling factors, such that the standard deviations of a rescaled response matrix are maximally similar. Figure F.1 shows that this rescaling does in fact increase the level of homoscedasticity.

BIBLIOGRAPHY

Abatzoglou, T., Mendel, J., and Harada, G. (1991). The Constrained Total Least Squares Technique and Its Applications to Harmonic Superresolution. *IEEE Transactions on Signal Processing* 39.5, pp. 1070–1087.

Abou-Jaoudé, W., Traynard, P., Monteiro, P. T., Saez-Rodriguez, J., Helikar, T., Thieffry, D., and Chaouiya, C. (2016). Logical Modeling and Dynamical Analysis of Cellular Networks. *Front Genet* 7.

Adamson, B. et al. (2016). A Multiplexed Single-Cell CRISPR Screening Platform Enables Systematic Dissection of the Unfolded Protein Response. *Cell* 167.7, 1867–1882.e21.

Aderem, A. (2005). Systems Biology: Its Practice and Challenges. *Cell* 121.4, pp. 511–513.

Aebersold, R. and Mann, M. (2003). Mass Spectrometry-Based Proteomics. *Nature* 422.6928 (6928), pp. 198–207.

Aebersold, R. and Mann, M. (2016). Mass-Spectrometric Exploration of Proteome Structure and Function. *Nature* 537.7620 (7620), pp. 347–355.

Aho, A. V., Garey, M. R., and Ullman, J. D. (1972). The Transitive Reduction of a Directed Graph. *SIAM Journal on Computing* 1.2, pp. 131–137.

Aibar, S. et al. (2017). SCENIC: Single-Cell Regulatory Network Inference and Clustering. *Nature Methods* 14.11 (11), pp. 1083–1086.

Alon, U. (2003). Biological Networks: The Tinkerer as an Engineer. *Science* 301.5641, pp. 1866–1867.

Alon, U. (2007). Network Motifs: Theory and Experimental Approaches. *Nature Reviews Genetics* 8.6, pp. 450–461.

Altay, G. and Emmert-Streib, F. (2010). Revealing Differences in Gene Network Inference Algorithms on the Network Level by Ensemble Methods. *Bioinformatics* 26.14, pp. 1738–1744.

Altschuler, S. J. and Wu, L. F. (2010). Cellular Heterogeneity: Do Differences Make a Difference? *Cell* 141.4, pp. 559–563.

Anderson, E., Bai, Z., Bischof, C., Blackford, L. S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., and Sorensen, D. (1999). LAPACK Users' Guide. 3rd ed. Software, Environments and Tools. Philadelphia, PA: Society for Industrial and Applied Mathematics.

Anderson, P. W. (1972). More Is Different. *Science* 177.4047, pp. 393–396.

Andrec, M., Kholodenko, B. N., Levy, R. M., and Sontag, E. (2005). Inference of Signaling and Gene Regulatory Networks by Steady-

State Perturbation Experiments: Structure and Accuracy. *Journal of Theoretical Biology* 232.3, pp. 427–441.

Apweiler, R. et al. (2018). Whither Systems Medicine? *Experimental & Molecular Medicine* 50.3 (3), e453–e453.

Arnol, D., Schapiro, D., Bodenmiller, B., Saez-Rodriguez, J., and Stegle, O. (2019). Modeling Cell-Cell Interactions from Spatial Molecular Data with Spatial Variance Component Analysis. *Cell Reports* 29.1, 202–211.e6.

Artzy-Randrup, Y., Fleishman, S. J., Ben-Tal, N., and Stone, L. (2004). Comment on "Network Motifs: Simple Building Blocks of Complex Networks" and "Superfamilies of Evolved and Designed Networks". *Science* 305.5687, pp. 1107–1107.

Auffray, C., Chen, Z., and Hood, L. (2009). Systems Medicine: The Future of Medical Genomics and Healthcare. *Genome Medicine* 1.1, p. 2.

Bailey, J. E. (1998). Mathematical Modeling and Analysis in Biochemical Engineering: Past Accomplishments and Future Opportunities. *Biotechnology Progress* 14.1, pp. 8–20.

Bailey, J., Bailey, J. E., Ollis, D. F., Simpson, R. J., and Ollis, D. F. (1986). Biochemical Engineering Fundamentals. 2nd ed. Singapore New York, NY: McGraw-Hill.

Bandura, D. R., Baranov, V. I., Ornatsky, O. I., Antonov, A., Kinach, R., Lou, X., Pavlov, S., Vorobiev, S., Dick, J. E., and Tanner, S. D. (2009). Mass Cytometry: Technique for Real Time Single Cell Multitarget Immunoassay Based on Inductively Coupled Plasma Time-of-Flight Mass Spectrometry. *Analytical Chemistry* 81.16, pp. 6813–6822.

Barski, A., Cuddapah, S., Cui, K., Roh, T.-Y., Schones, D. E., Wang, Z., Wei, G., Chepelev, I., and Zhao, K. (2007). High-Resolution Profiling of Histone Methylations in the Human Genome. *Cell* 129.4, pp. 823–837.

Barzel, B. and Barabási, A.-L. (2013). Network Link Prediction by Global Silencing of Indirect Correlations. *Nature Biotechnology* 31.8 (8), pp. 720–725.

Basso, K., Margolin, A. A., Stolovitzky, G., Klein, U., Dalla-Favera, R., and Califano, A. (2005). Reverse Engineering of Regulatory Networks in Human B Cells. *Nature Genetics* 37.4 (4), pp. 382–390.

Bastiaens, P. et al. (2015). Silence on the Relevant Literature and Errors in Implementation. *Nature Biotechnology* 33.4 (4), pp. 336–339.

Bellman, R. and Åström, K. J. (1970). On Structural Identifiability. *Mathematical Biosciences* 7.3, pp. 329–339.

Berestovsky, N. and Nakhleh, L. (2013). An Evaluation of Methods for Inferring Boolean Networks from Time-Series Data. *PLOS ONE* 8.6, e66031.

Bhalla, U. S. and Iyengar, R. (1999). Emergent Properties of Networks of Biological Signaling Pathways. *Science* 283.5400, pp. 381–387.

Birget, J.-C., Lun, D. S., Wirth, A., and Hong, D. (2012). A Theoretical Approach to Gene Network Identification. *2012 IEEE Information Theory Workshop*. 2012 IEEE Information Theory Workshop, pp. 432–436.

Bizzarri, M., Brash, D. E., Briscoe, J., Grieneisen, V. A., Stern, C. D., and Levin, M. (2019). A Call for a Better Understanding of Causation in Cell Biology. *Nature reviews. Molecular cell biology* 20.5, pp. 261–262.

Bonneau, R., Reiss, D. J., Shannon, P., Facciotti, M., Hood, L., Baliga, N. S., and Thorsson, V. (2006). The Inferelator: An Algorithm for Learning Parsimonious Regulatory Networks from Systems-Biology Data Sets de Novo. *Genome Biology* 7.5, R36.

Bosdriesz, E., Prahallad, A., Klinger, B., Sieber, A., Bosma, A., Bernards, R., Blüthgen, N., and Wessels, L. F. A. (2018). Comparative Network Reconstruction Using Mixed Integer Programming. *Bioinformatics* 34.17, pp. i997–i1004.

Branch, M. A., Coleman, T. F., and Li, Y. (1999). A Subspace, Interior, and Conjugate Gradient Method for Large-Scale Bound-Constrained Minimization Problems. *SIAM Journal on Scientific Computing* 21.1, pp. 1–23.

Brandt, R. et al. (2019). Cell Type-Dependent Differential Activation of ERK by Oncogenic KRAS in Colon Cancer and Intestinal Epithelium. *Nature Communications* 10.1 (1), pp. 1–15.

Bruggeman, F. J. and Kholodenko, B. N. (2002). Modular Interaction Strengths in Regulatory Networks; An Example. *Molecular Biology Reports* 29.1, pp. 57–61.

Bruggeman, F. J., Westerhoff, H. V., Hoek, J. B., and Kholodenko, B. N. (2002). Modular Response Analysis of Cellular Regulatory Networks. *Journal of Theoretical Biology* 218.4, pp. 507–520.

Buchman, T. G. (2002). The Community of the Self. *Nature* 420.6912, pp. 246–251.

Buenrostro, J. D., Wu, B., Litzenburger, U. M., Ruff, D., Gonzales, M. L., Snyder, M. P., Chang, H. Y., and Greenleaf, W. J. (2015). Single-Cell Chromatin Accessibility Reveals Principles of Regulatory Variation. *Nature* 523.7561 (7561), pp. 486–490.

Busiello, D. M., Suweis, S., Hidalgo, J., and Maritan, A. (2017). Explorability and the Origin of Network Sparsity in Living Systems. *Scientific Reports* 7.1 (1), pp. 1–8.

Carandini, M. (2012). From Circuits to Behavior: A Bridge Too Far? *Nature Neuroscience* 15.4 (4), pp. 507–509.

Carrera, J. and Covert, M. W. (2015). Why Build Whole-Cell Models? *Trends in Cell Biology*. Special Issue: Quantitative Cell Biology 25.12, pp. 719–722.

Chiquet, J., Rigaill, G., and Sundqvist, M. (2019). A Multiattribute Gaussian Graphical Model for Inferring Multiscale Regulatory Networks: An Application in Breast Cancer. *Gene Regulatory Networks: Methods and Protocols*. Ed. by G. Sanguinetti and V. A. Huynh-Thu.

Methods in Molecular Biology. New York, NY: Springer, pp. 143–160.

Chis, O.-T., Banga, J. R., and Balsa-Canto, E. (2011). Structural Identifiability of Systems Biology Models: A Critical Comparison of Methods. *PLOS ONE* 6.11, e27755.

Chuang, H.-Y., Hofree, M., and Ideker, T. (2010). A Decade of Systems Biology. *Annual Review of Cell and Developmental Biology* 26.1, pp. 721–744.

Civelek, M. and Lusis, A. J. (2014). Systems Genetics Approaches to Understand Complex Traits. *Nature Reviews Genetics* 15.1 (1), pp. 34–48.

Costanzo, M. et al. (2010). The Genetic Landscape of a Cell. *Science (New York, N.Y.)* 327.5964, pp. 425–431.

Costanzo, M. et al. (2016). A Global Genetic Interaction Network Maps a Wiring Diagram of Cellular Function. *Science* 353.6306.

Cox, J. and Mann, M. (2011). Quantitative, High-Resolution Proteomics for Data-Driven Systems Biology. *Annual Review of Biochemistry* 80.1, pp. 273–299.

Crick, F. H. C. (1973). Project K: "The Complete Solution of E. Coli". *Perspectives in Biology and Medicine* 17.1, pp. 67–70.

Dal Palù, A., Dovier, A., Formisano, A., and Pontelli, E. (2018). Exploring Life: Answer Set Programming in Bioinformatics. *Declarative Logic Programming: Theory, Systems, and Applications*. Ed. by M. Kifer and A. L. Yanhong. Vol. 20. San Rafael, CA: Association for Computing Machinery and Morgan & Claypool, pp. 359–412.

Datlinger, P., Rendeiro, A. F., Schmidl, C., Krausgruber, T., Traxler, P., Klughammer, J., Schuster, L. C., Kuchler, A., Alpar, D., and Bock, C. (2017). Pooled CRISPR Screening with Single-Cell Transcriptome Readout. *Nature Methods* 14.3 (3), pp. 297–301.

Davis, M. M., Tato, C. M., and Furman, D. (2017). Systems Immunology: Just Getting Started. *Nature Immunology* 18.7 (7), pp. 725–732.

De la Fuente, A. (2014). Gene Network Inference: Verification of Methods for Systems Genetics Data. Berlin Heidelberg: Springer Science & Business Media. 135 pp.

De la Fuente, A., Brazhnik, P., and Mendes, P. (2002). Linking the Genes: Inferring Quantitative Gene Networks from Microarray Data. *Trends in Genetics* 18.8, pp. 395–398.

De la Fuente, A. and Mendes, P. (2002). Quantifying Gene Networks with Regulatory Strengths. *Molecular Biology Reports* 29.1, pp. 73–77.

De Martino, A. and De Martino, D. (2018). An Introduction to the Maximum Entropy Approach and Its Application to Inference Problems in Biology. *Heliyon* 4.4, e00596.

Ud-Dean, S. M. M. and Gunawan, R. (2014). Ensemble Inference and Inferability of Gene Regulatory Networks. *PLOS ONE* 9.8, e103812.

Ud-Dean, S. M. M. and Gunawan, R. (2016). Optimal Design of Gene Knockout Experiments for Gene Regulatory Network Inference. *Bioinformatics* 32.6, pp. 875–883.

Dean, T. and Kanazawa, K. (1989). A Model for Reasoning about Persistence and Causation. *Computational Intelligence* 5.2, pp. 142–150.

Dev, S. B. (2015). Unsolved Problems in Biology—The State of Current Thinking. *Progress in Biophysics and Molecular Biology* 117.2, pp. 232–239.

Dhurjati, P., Ramkrishna, D., Flickinger, M. C., and Tsao, G. T. (1985). A Cybernetic View of Microbial Growth: Modeling of Cells as Optimal Strategists. *Biotechnology and Bioengineering* 27.1, pp. 1–9.

Ding, L. et al. (2018). Perspective on Oncogenic Processes at the End of the Beginning of Cancer Genomics. *Cell* 173.2, 305–320.e10.

Dixit, A. et al. (2016). Perturb-Seq: Dissecting Molecular Circuits with Scalable Single-Cell RNA Profiling of Pooled Genetic Screens. *Cell* 167.7, 1853–1866.e17.

Doolittle, W. F. (2013). Is Junk DNA Bunk? A Critique of ENCODE. *Proceedings of the National Academy of Sciences* 110.14, pp. 5294–5300.

Dorel, M., Klinger, B., Gross, T., Sieber, A., Prahallad, A., Bosdriesz, E., Wessels, L. F. A., and Blüthgen, N. (2018). Modelling Signalling Networks from Perturbation Data. *Bioinformatics* 34.23, pp. 4079–4086.

Drees, B. L., Thorsson, V., Carter, G. W., Rives, A. W., Raymond, M. Z., Avila-Campillo, I., Shannon, P., and Galitski, T. (2005). Derivation of Genetic Interaction Networks from Quantitative Phenotype Data. *Genome Biology* 6.4, R38.

Eckart, C. and Young, G. (1936). The Approximation of One Matrix by Another of Lower Rank. *Psychometrika* 1.3, pp. 211–218.

Eddy, S. R. (2013). The ENCODE Project: Missteps Overshadowing a Success. *Current Biology* 23.7, R259–R261.

Elowitz, M. B., Levine, A. J., Siggia, E. D., and Swain, P. S. (2002). Stochastic Gene Expression in a Single Cell. *Science* 297.5584, pp. 1183–1186.

Elsasser, W. M. (1984). Outline of a Theory of Cellular Heterogeneity. *Proceedings of the National Academy of Sciences* 81.16, pp. 5126–5129.

ENCODE Project Consortium (2012). An Integrated Encyclopedia of DNA Elements in the Human Genome. *Nature* 489.7414, pp. 57–74.

Faith, J. J., Hayete, B., Thaden, J. T., Mogno, I., Wierzbowski, J., Cottarel, G., Kasif, S., Collins, J. J., and Gardner, T. S. (2007). Large-Scale Mapping and Validation of Escherichia Coli Transcriptional Regulation from a Compendium of Expression Profiles. *PLOS Biology* 5.1, e8.

Fell, D. A. and Small, J. R. (1986). Fat Synthesis in Adipose Tissue. An Examination of Stoichiometric Constraints. *Biochemical Journal* 238.3, pp. 781–786.

Ferrell, J. E. and Machleder, E. M. (1998). The Biochemical Basis of an All-or-None Cell Fate Switch in Xenopus Oocytes. *Science (New York, N.Y.)* 280.5365, pp. 895–898.

Friedman, N. (2004). Inferring Cellular Networks Using Probabilistic Graphical Models. *Science* 303.5659, pp. 799–805.

Friedman, N. and Koller, D. (2003). Being Bayesian about Network Structure: A Bayesian Approach to Structure Discovery in Bayesian Networks. *Machine Learning* 50.1-2, pp. 95–125.

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000). Using Bayesian Networks to Analyze Expression Data. *Journal of Computational Biology* 7.3-4, pp. 601–620.

Friston, K., Levin, M., Sengupta, B., and Pezzulo, G. (2015). Knowing One's Place: A Free-Energy Approach to Pattern Regulation. *Journal of The Royal Society Interface* 12.105, p. 20141383.

Gardner, T. S., Bernardo, D. di, Lorenz, D., and Collins, J. J. (2003). Inferring Genetic Networks and Identifying Compound Mode of Action via Expression Profiling. *Science* 301.5629, pp. 102–105.

Gardner, T. S. and Faith, J. J. (2005). Reverse-Engineering Transcription Control Networks. *Physics of Life Reviews* 2.1, pp. 65–88.

Gates, A. J. and Rocha, L. M. (2016). Control of Complex Networks Requires Both Structure and Dynamics. *Scientific Reports* 6.1 (1), pp. 1–11.

Gebser, M., Kaminski, R., Kaufmann, B., and Schaub, T. (2014). Clingo = ASP + Control: Preliminary Report.

Gebser, M., Konig, A., Schaub, T., Thiele, S., and Veber, P. (2010). The BioASP Library: ASP Solutions for Systems Biology. *2010 22nd IEEE International Conference on Tools with Artificial Intelligence*. Arras, France: IEEE, pp. 383–389.

Gerber, D. E. (2008). Targeted Therapies: A New Generation of Cancer Treatments. *American Family Physician* 77.3, pp. 311–319.

Ghanbari, M., Lasserre, J., and Vingron, M. (2015). Reconstruction of Gene Networks Using Prior Knowledge. *BMC Systems Biology* 9.1, p. 84.

Ghanbari, M., Lasserre, J., and Vingron, M. (2019). The Distance Precision Matrix: Computing Networks from Non-Linear Relationships. *Bioinformatics* 35.6, pp. 1009–1017.

Glass, L. and Kauffman, S. A. (1973). The Logical Analysis of Continuous, Non-Linear Biochemical Control Networks. *Journal of Theoretical Biology* 39.1, pp. 103–129.

Godoy, L. M. F. de, Olsen, J. V., Cox, J., Nielsen, M. L., Hubner, N. C., Fröhlich, F., Walther, T. C., and Mann, M. (2008). Comprehensive Mass-Spectrometry-Based Proteome Quantification of Haploid versus Diploid Yeast. *Nature* 455.7217 (7217), pp. 1251–1254.

Goh, K.-I., Cusick, M. E., Valle, D., Childs, B., Vidal, M., and Barabási, A.-L. (2007). The Human Disease Network. *Proceedings of the National Academy of Sciences* 104.21, pp. 8685–8690.

Golub, G. H., Hoffman, A., and Stewart, G. W. (1987). A Generalization of the Eckart-Young-Mirsky Matrix Approximation Theorem. *Linear Algebra and its Applications* 88-89, pp. 317–327.

Golub, G. and van Loan, C. (1980). An Analysis of the Total Least Squares Problem. *SIAM Journal on Numerical Analysis* 17.6, pp. 883–893.

Green, S., ed. (2016). Philosophy of Systems Biology: Perspectives from Scientists and Philosophers. New York, NY: Springer.

Greenfield, A., Madar, A., Ostrer, H., and Bonneau, R. (2010). DREAM4: Combining Genetic and Dynamic Information to Identify Biological Networks and Dynamical Models. *PLOS ONE* 5.10, e13397.

Gross, F. G. (2017). The Sum of the Parts: Large-Scale Modeling in Systems Biology. *Philosophy & Theory in Biology* 9.

Gross, T. and Blüthgen, N. (2020). Identifiability and Experimental Design in Perturbation Studies. *Bioinformatics* [accepted].

Gross, T., Wongchenko, M. J., Yan, Y., and Blüthgen, N. (2019). Robust Network Inference Using Response Logic. *Bioinformatics* 35.14, pp. i634–i642.

Gu, M., Weedbrook, C., Perales, Á., and Nielsen, M. A. (2009). More Really Is Different. *Physica D: Nonlinear Phenomena* 238.9, pp. 835–839.

Guet, C. C., Elowitz, M. B., Hsing, W., and Leibler, S. (2002). Combinatorial Synthesis of Genetic Networks. *Science* 296.5572, pp. 1466–1470.

Guven-Maiorov, E., Tsai, C.-J., and Nussinov, R. (2017). Structural Host-Microbiota Interaction Networks. *PLOS Computational Biology* 13.10, e1005579.

Guziolowski, C., Videla, S., Eduati, F., Thiele, S., Cokelaer, T., Siegel, A., and Saez-Rodriguez, J. (2013). Exhaustively Characterizing Feasible Logic Models of a Signaling Network Using Answer Set Programming. *Bioinformatics* 29.18, pp. 2320–2326.

Haibe-Kains, B. and Emmert-Streib, F., eds. (2015). Quantitative Assessment and Validation of Network Inference Methods in Bioinformatics. Lausanne: Frontiers Media SA. 192 pp.

Halasz, M., Kholodenko, B. N., Kolch, W., and Santra, T. (2016). Integrating Network Reconstruction with Mechanistic Modeling to Predict Cancer Therapies. *Science Signaling* 9.455, ra114–ra114.

Hartwell, L. H., Hopfield, J. J., Leibler, S., and Murray, A. W. (1999). From Molecular to Modular Cell Biology. *Nature* 402.6761, pp. C47–C52.

Hasin, Y., Seldin, M., and Lusis, A. (2017). Multi-Omics Approaches to Disease. *Genome Biology* 18.1, p. 83.

Hecker, M., Lambeck, S., Toepfer, S., van Someren, E., and Guthke, R. (2009). Gene Regulatory Network Inference: Data Integration in Dynamic Models—A Review. *Biosystems* 96.1, pp. 86–103.

Heinrich, R. and Rapoport, T. A. (1974). A Linear Steady-State Treatment of Enzymatic Chains. *European Journal of Biochemistry* 42.1, pp. 89–95.

Hickman, G. J. and Hodgman, T. C. (2009). Inference of Gene Regulatory Networks Using Boolean-Network Inference Methods. *Journal of Bioinformatics and Computational Biology* 07.06, pp. 1013–1029.

Hill, S. M., Lu, Y., Molina, J., Heiser, L. M., Spellman, P. T., Speed, T. P., Gray, J. W., Mills, G. B., and Mukherjee, S. (2012). Bayesian Inference of Signaling Network Topology in a Cancer Cell Line. *Bioinformatics* 28.21, pp. 2804–2810.

Hill, S. M. et al. (2016). Inferring Causal Molecular Networks: Empirical Assessment through a Community-Based Effort. *Nature Methods* 13.4, pp. 310–318.

Hoadley, K. A. et al. (2018). Cell-of-Origin Patterns Dominate the Molecular Classification of 10,000 Tumors from 33 Types of Cancer. *Cell* 173.2, 291–304.e6.

Huffel, S. V. and Vandewalle, J. (1991). The Total Least Squares Problem: Computational Aspects and Analysis. Philadelphia, PA: SIAM.

Ideker, T. E., Thorsson, V., and Karp, R. M. (2000). Discovery of Regulatory Interactions through Perturbation: Inference and Experimental Design. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pp. 305–316.

Ideker, T., Galitski, T., and Hood, L. (2001). A New Approach to Decoding Life: Systems Biology. *Annual Review of Genomics and Human Genetics* 2.1, pp. 343–372.

Ideker, T., Ozier, O., Schwikowski, B., and Siegel, A. F. (2002). Discovering Regulatory and Signalling Circuits in Molecular Interaction Networks. *Bioinformatics* 18 (suppl_1), S233–S240.

Ingram, P. J., Stumpf, M. P., and Stark, J. (2006). Network Motifs: Structure Does Not Determine Function. *BMC Genomics* 7.1, p. 108.

Invergo, B. M. and Beltrao, P. (2018). Reconstructing Phosphorylation Signalling Networks from Quantitative Phosphoproteomic Data. *Essays in Biochemistry* 62.4, pp. 525–534.

Jaitin, D. A., Weiner, A., Yofe, I., Lara-Astiaso, D., Keren-Shaul, H., David, E., Salame, T. M., Tanay, A., van Oudenaarden, A., and Amit, I. (2016). Dissecting Immune Circuits by Linking CRISPR-Pooled Screens with Single-Cell RNA-Seq. *Cell* 167.7, 1883–1896.e15.

Janes, K. A. and Lauffenburger, D. A. (2013). Models of Signalling Networks – What Cell Biologists Can Gain from Them and Give to Them. *Journal of Cell Science* 126.9, pp. 1913–1921.

Jansen, R. C. (2003). Studying Complex Biological Systems Using Multifactorial Perturbation. *Nature Reviews Genetics* 4.2 (2), pp. 145–151.

Jimenez-Dominguez, G., Ravel, P., Jalaguier, S., Cavaillès, V., and Colinge, J. (2020). aiMeRA: A Generic Modular Response Analysis R

Package and Its Application to Estrogen and Retinoic Acid Receptors Crosstalk. *bioRxiv*, p. 2020.01.30.925800.

Johnson, D. S., Mortazavi, A., Myers, R. M., and Wold, B. (2007). Genome-Wide Mapping of in Vivo Protein-DNA Interactions. *Science* 316.5830, pp. 1497–1502.

Jonas, E. and Kording, K. P. (2017). Could a Neuroscientist Understand a Microprocessor? *PLOS Computational Biology* 13.1, e1005268.

Jurman, G., Filosi, M., Visintainer, R., Riccadonna, S., and Furlanello, C. (2019). Stability in GRN Inference. *Gene Regulatory Networks: Methods and Protocols*. Ed. by G. Sanguinetti and V. A. Huynh-Thu. Methods in Molecular Biology. New York, NY: Springer, pp. 323–346.

Kacser, H. and Burns, J. A. (1973). The Control of Flux. *Symposia of the Society for Experimental Biology* 27, pp. 65–104.

Kanehisa, M. and Goto, S. (2000). KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research* 28.1, pp. 27–30.

Karr, J. R., Sanghvi, J. C., Macklin, D. N., Gutschow, M. V., Jacobs, J. M., Bolival, B., Assad-Garcia, N., Glass, J. I., and Covert, M. W. (2012). A Whole-Cell Computational Model Predicts Phenotype from Genotype. *Cell* 150.2, pp. 389–401.

Kashtan, N. and Alon, U. (2005). Spontaneous Evolution of Modularity and Network Motifs. *Proceedings of the National Academy of Sciences* 102.39, pp. 13773–13778.

Kauffman, S. A. (1969). Metabolic Stability and Epigenesis in Randomly Constructed Genetic Nets. *Journal of Theoretical Biology* 22.3, pp. 437–467.

Kauffman, S. A. (1993). The Origins of Order: Self-Organization and Selection in Evolution. Oxford: Oxford University Press.

Kholodenko, B. N., Kiyatkin, A., Bruggeman, F. J., Sontag, E., Westerhoff, H. V., and Hoek, J. B. (2002). Untangling the Wires: A Strategy to Trace Functional Interactions in Signaling and Gene Networks. *Proceedings of the National Academy of Sciences* 99.20, pp. 12841–12846.

Kim, M.-S. et al. (2014). A Draft Map of the Human Proteome. *Nature* 509.7502 (7502), pp. 575–581.

King, R. D. et al. (2009). The Automation of Science. *Science* 324.5923, pp. 85–89.

Kitano, H. (2002). Computational Systems Biology. *Nature* 420.6912, pp. 206–210.

Klinger, B. and Blüthgen, N. (2018). Reverse Engineering Gene Regulatory Networks by Modular Response Analysis – a Benchmark. *Essays in Biochemistry* 62.4. Ed. by W. Kolch, D. Fey, and C. J. Ryan, pp. 535–547.

Klinger, B., Sieber, A., Fritsche-Guenther, R., Witzel, F., Berry, L., Schumacher, D., Yan, Y., Durek, P., Merchant, M., Schäfer, R., Sers, C., and Blüthgen, N. (2013). Network Quantification of EGFR Signaling Unveils Potential for Targeted Combination Therapy. *Molecular Systems Biology* 9.1, p. 673.

Koller, D. and Friedman, N. (2009). Probabilistic Graphical Models: Principles and Techniques. Cambridge, MA: MIT Press. 1268 pp.

Kollmann, M., Løvdok, L., Bartholomé, K., Timmer, J., and Sourjik, V. (2005). Design Principles of a Bacterial Signalling Network. *Nature* 438.7067, pp. 504–507.

Krogan, N. J. et al. (2006). Global Landscape of Protein Complexes in the Yeast Saccharomyces Cerevisiae. *Nature* 440.7084 (7084), pp. 637–643.

Krohs, U. and Callebaut, W. (2007). 9 - Data without Models Merging with Models without Data. *Systems Biology*. Ed. by F. C. Boogerd, F. J. Bruggeman, J.-H. S. Hofmeyr, and H. V. Westerhoff. Amsterdam: Elsevier, pp. 181–213.

Kuepfer, L., Peter, M., Sauer, U., and Stelling, J. (2007). Ensemble Modeling for Analysis of Cell Signaling Dynamics. *Nature Biotechnology* 25.9 (9), pp. 1001–1006.

Kurdyukov, S. and Bullock, M. (2016). DNA Methylation Analysis: Choosing the Right Method. *Biology* 5.1.

Kussell, E. and Leibler, S. (2005). Phenotypic Diversity, Population Growth, and Information in Fluctuating Environments. *Science* 309.5743, pp. 2075–2078.

Lander, A. D. (2010). The Edges of Understanding. *BMC Biology* 8.1, p. 40.

Lander, E. S. et al. (2001). Initial Sequencing and Analysis of the Human Genome. *Nature* 409.6822, pp. 860–921.

Lang, M., Summers, S., and Stelling, J. (2014). Cutting the Wires: Modularization of Cellular Networks for Experimental Design. *Biophysical Journal* 106.1, pp. 321–331.

Langfelder, P. and Horvath, S. (2008). WGCNA: An R Package for Weighted Correlation Network Analysis. *BMC Bioinformatics* 9.1, p. 559.

Lauffenburger, D. A. and Linderman, J. J. (1996). Receptors: Models for Binding, Trafficking, and Signaling. New York, NY: OUP USA.

Lazebnik, Y. (2002). Can a Biologist Fix a Radio?—Or, What I Learned While Studying Apoptosis. *Cancer Cell* 2.3, pp. 179–182.

Leclerc, R. D. (2008). Survival of the Sparsest: Robust Gene Networks Are Parsimonious. *Molecular Systems Biology* 4.1, p. 213.

Leiserson, M. D. M. et al. (2015). Pan-Cancer Network Analysis Identifies Combinations of Rare Somatic Mutations across Pathways and Protein Complexes. *Nature Genetics* 47.2 (2), pp. 106–114.

Liang, S., Fuhrman, S., and Somogyi, R. (1998). Reveal, A General Reverse Engineering Algorithm for Inference of Genetic Network Architectures.

Lingeman, J. M. and Shasha, D. (2012). Network Inference in Molecular Biology: A Hands-on Framework. SpringerBriefs in Electrical and Computer Engineering. New York, NY: Springer-Verlag.

Linnarsson, S. and Teichmann, S. A. (2016). Single-Cell Genomics: Coming of Age. *Genome Biology* 17.1, p. 97.

Liu, A., Trairatphisan, P., Gjerga, E., Didangelos, A., Barratt, J., and Saez-Rodriguez, J. (2019). From Expression Footprints to Causal Pathways: Contextualizing Large Signaling Networks with CARNIVAL. *npj Systems Biology and Applications* 5.1 (1), pp. 1–10.

Luo, C. et al. (2017). Single-Cell Methylomes Identify Neuronal Subtypes and Regulatory Elements in Mammalian Cortex. *Science* 357.6351, pp. 600–604.

Marbach, D., Prill, R. J., Schaffter, T., Mattiussi, C., Floreano, D., and Stolovitzky, G. (2010). Revealing Strengths and Weaknesses of Methods for Gene Network Inference. *Proceedings of the National Academy of Sciences* 107.14, pp. 6286–6291.

Markowetz, F., Kostka, D., Troyanskaya, O. G., and Spang, R. (2007). Nested Effects Models for High-Dimensional Phenotyping Screens. *Bioinformatics* 23.13, pp. i305–i312.

Markowetz, F. and Spang, R. (2007). Inferring Cellular Networks – a Review. *BMC Bioinformatics* 8.6, S5.

Marks, D. S., Colwell, L. J., Sheridan, R., Hopf, T. A., Pagnani, A., Zecchina, R., and Sander, C. (2011). Protein 3D Structure Computed from Evolutionary Sequence Variation. *PLOS ONE* 6.12, e28766.

Matsumoto, H., Kiryu, H., Furusawa, C., Ko, M. S. H., Ko, S. B. H., Gouda, N., Hayashi, T., and Nikaido, I. (2017). SCODE: An Efficient Regulatory Network Inference Algorithm from Single-Cell RNA-Seq during Differentiation. *Bioinformatics* 33.15, pp. 2314–2321.

Meisig, J. and Blüthgen, N. (2018). The Gene Regulatory Network of mESC Differentiation: A Benchmark for Reverse Engineering Methods. *Philosophical Transactions of the Royal Society B: Biological Sciences* 373.1750, p. 20170222.

Menon, R., Ramanan, V., and Korolev, K. S. (2018). Interactions between Species Introduce Spurious Associations in Microbiome Studies. *PLOS Computational Biology* 14.1, e1005939.

Milo, R., Itzkovitz, S., Kashtan, N., Levitt, R., Shen-Orr, S., Ayzenshtat, I., Sheffer, M., and Alon, U. (2004). Superfamilies of Evolved and Designed Networks. *Science* 303.5663, pp. 1538–1542.

Milo, R., Shen-Orr, S. S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2002). Network Motifs: Simple Building Blocks of Complex Networks. *Science* 298.5594, pp. 824–827.

Molinelli, E. J. et al. (2013). Perturbation Biology: Inferring Signaling Networks in Cellular Systems. *PLoS Computational Biology* 9.12.

Mora, T. and Bialek, W. (2011). Are Biological Systems Poised at Criticality? *Journal of Statistical Physics* 144.2, pp. 268–302.

Morris, M. K., Saez-Rodriguez, J., Sorger, P. K., and Lauffenburger, D. A. (2010). Logic-Based Models for the Analysis of Cell Signaling Networks. *Biochemistry* 49.15, pp. 3216–3224.

Natale, J. L., Hofmann, D., Hernández, D. G., and Nemenman, I. (2017). Reverse-Engineering Biological Networks from Large Data Sets.

Navin, N. et al. (2011). Tumour Evolution Inferred by Single-Cell Sequencing. *Nature* 472.7341 (7341), pp. 90–94.

Needham, C. J., Bradford, J. R., Bulpitt, A. J., and Westhead, D. R. (2006). Inference in Bayesian Networks. *Nature Biotechnology* 24.1 (1), pp. 51–53.

Newman, J. R. S., Ghaemmaghami, S., Ihmels, J., Breslow, D. K., Noble, M., DeRisi, J. L., and Weissman, J. S. (2006). Single-Cell Proteomic Analysis of S. Cerevisiae Reveals the Architecture of Biological Noise. *Nature* 441.7095 (7095), pp. 840–846.

Noble, D. (2008). Genes and Causation. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 366.1878, pp. 3001–3015.

Noble, D. (2012). A Theory of Biological Relativity: No Privileged Level of Causation. *Interface Focus* 2.1, pp. 55–64.

Orth, J. D., Conrad, T. M., Na, J., Lerman, J. A., Nam, H., Feist, A. M., and Palsson, B. Ø. (2011). A Comprehensive Genome-Scale Reconstruction of Escherichia Coli Metabolism—2011. *Molecular Systems Biology* 7.1, p. 535.

Ostrowski, M., Schaub, T., Durzinsky, M., Marwan, W., and Wagler, A. (2011). Automatic Network Reconstruction Using ASP.

Ozbudak, E. M., Thattai, M., Kurtser, I., Grossman, A. D., and Oudenaarden, A. van (2002). Regulation of Noise in the Expression of a Single Gene. *Nature Genetics* 31.1 (1), pp. 69–73.

Parikh, A. P., Wu, W., Curtis, R. E., and Xing, E. P. (2011). TREEGL: Reverse Engineering Tree-Evolving Gene Networks Underlying Developing Biological Lineages. *Bioinformatics* 27.13, pp. i196–i204.

Pearl, J. (2009). Causality. Cambridge: Cambridge University Press. 486 pp.

Pe'er, D., Regev, A., Elidan, G., and Friedman, N. (2001). Inferring Subnetworks from Perturbed Expression Profiles. *Bioinformatics* 17 (suppl_1), S215–S224.

Perron, O. (1907). Zur Theorie Der Matrices. *Mathematische Annalen* 64.2, pp. 248–263.

Pezzulo, G. and Levin, M. (2016). Top-down Models in Biology: Explanation and Control of Complex Living Systems above the Molecular Level. *Journal of The Royal Society Interface* 13.124, p. 20160555.

Polychronidou, M. and Lemberger, T. (2017). Methods to Drive Systems Biology Forward. *Molecular Systems Biology* 13.12, p. 996.

Raue, A., Kreutz, C., Maiwald, T., Bachmann, J., Schilling, M., Klingmüller, U., and Timmer, J. (2009). Structural and Practical Identifiability Analysis of Partially Observed Dynamical Models by Exploiting the Profile Likelihood. *Bioinformatics* 25.15, pp. 1923–1929.

Regev, A. et al. (2017). The Human Cell Atlas. *eLife* 6. Ed. by T. R. Gingeras, e27041.

Rubin, H. (1990). The Significance of Biological Heterogeneity. *Cancer Metastasis Reviews* 9.1, pp. 1–20.

Rung, J., Schlitt, T., Brazma, A., Freivalds, K., and Vilo, J. (2002). Building and Analysing Genome-Wide Gene Disruption Networks. *Bioinformatics* 18 (suppl_2), S202–S210.

Runge, J., Heitzig, J., Petoukhov, V., and Kurths, J. (2012). Escaping the Curse of Dimensionality in Estimating Multivariate Transfer Entropy. *Physical Review Letters* 108.25, p. 258701.

Saez-Rodriguez, J., Alexopoulos, L. G., Epperlein, J., Samaga, R., Lauffenburger, D. A., Klamt, S., and Sorger, P. K. (2009). Discrete Logic Modelling as a Means to Link Protein Signalling Networks with Functional Analysis of Mammalian Signal Transduction. *Molecular Systems Biology* 5.1, p. 331.

Saint-Antoine, M. M. and Singh, A. (2020). Network Inference in Systems Biology: Recent Developments, Challenges, and Applications. *Current Opinion in Biotechnology* 63, pp. 89–98.

Sanchez-Vega, F. et al. (2018). Oncogenic Signaling Pathways in The Cancer Genome Atlas. *Cell* 173.2, 321–337.e10.

Sanguinetti, G. and Huynh-Thu, V. A., eds. (2019). Gene Regulatory Networks: Methods and Protocols. Vol. 1883. Methods in Molecular Biology. New York, NY: Springer New York.

Santos, S. D. M., Verveer, P. J., and Bastiaens, P. I. H. (2007). Growth Factor-Induced MAPK Network Topology Shapes Erk Response Determining PC-12 Cell Fate. *Nature Cell Biology* 9.3 (3), pp. 324–330.

Santra, T., Kolch, W., and Kholodenko, B. N. (2013). Integrating Bayesian Variable Selection with Modular Response Analysis to Infer Biochemical Network Topology. *BMC Systems Biology* 7, p. 57.

Santra, T., Rukhlenko, O., Zhernovkov, V., and Kholodenko, B. N. (2018). Reconstructing Static and Dynamic Models of Signaling Pathways Using Modular Response Analysis. *Current Opinion in Systems Biology*. Mathematic Modelling 9, pp. 11–21.

Schäfer, J. and Strimmer, K. (2005). An Empirical Bayes Approach to Inferring Large-Scale Gene Association Networks. *Bioinformatics (Oxford, England)* 21.6, pp. 754–764.

Schaffrin, B. and Felus, Y. A. (2008). On the Multivariate Total Least-Squares Approach to Empirical Coordinate Transformations. Three Algorithms. *Journal of Geodesy* 82.6, pp. 373–383.

Schena, M., Shalon, D., Davis, R. W., and Brown, P. O. (1995). Quantitative Monitoring of Gene Expression Patterns with a Complementary DNA Microarray. *Science* 270.5235, pp. 467–470.

Schraivogel, D., Gschwind, A. R., Milbank, J. H., Leonce, D. R., Jakob, P., Mathur, L., Korbel, J. O., Merten, C. A., Velten, L., and Steinmetz, L. M. (2020). Targeted Perturb-Seq Enables Genome-Scale Genetic Screens in Single Cells. *Nature Methods* 17.6 (6), pp. 629–635.

Schulthess, P. and Blüthgen, N. (2011). Chapter Twenty - From Reaction Networks to Information Flow—Using Modular Response

Analysis to Track Information in Signaling Networks. *Methods in Enzymology*. Ed. by D. Jameson, M. Verma, and H. V. Westerhoff. Vol. 500. Methods in Systems Biology. Academic Press, pp. 397–409.

Schwab, D. J., Nemenman, I., and Mehta, P. (2014). Zipf's Law and Criticality in Multivariate Data without Fine-Tuning. *Physical Review Letters* 113.6, p. 068102.

Sever, R. and Brugge, J. S. (2015). Signal Transduction in Cancer. *Cold Spring Harbor Perspectives in Medicine* 5.4.

Sharan, R., Ulitsky, I., and Shamir, R. (2007). Network-Based Prediction of Protein Function. *Molecular Systems Biology* 3, p. 88.

Sharma, K., D'Souza, R. C. J., Tyanova, S., Schaab, C., Wiśniewski, J. R., Cox, J., and Mann, M. (2014). Ultradeep Human Phosphoproteome Reveals a Distinct Regulatory Nature of Tyr and Ser/Thr-Based Signaling. *Cell Reports* 8.5, pp. 1583–1594.

Shayeghi, N., Ng, T., and Coolen, A. C. C. (2012). Direct Response Analysis in Cellular Signalling Networks. *Journal of Theoretical Biology* 304, pp. 219–225.

Shen-Orr, S. S., Milo, R., Mangan, S., and Alon, U. (2002). Network Motifs in the Transcriptional Regulation Network of Escherichia Coli. *Nature Genetics* 31.1 (1), pp. 64–68.

Shew, W. L. and Plenz, D. (2013). The Functional Benefits of Criticality in the Cortex. *The Neuroscientist* 19.1, pp. 88–100.

Shreeve, J. (2007). The Genome War: How Craig Venter Tried to Capture the Code of Life and Save the World. New York, NY: Random House Publishing Group. 418 pp.

Siahpirani, A. F., Chasman, D., and Roy, S. (2019). Integrative Approaches for Inference of Genome-Scale Gene Regulatory Networks. *Gene Regulatory Networks: Methods and Protocols*. Ed. by G. Sanguinetti and V. A. Huynh-Thu. Methods in Molecular Biology. New York, NY: Springer, pp. 161–194.

Slatko, B. E., Gardner, A. F., and Ausubel, F. M. (2018). Overview of Next-Generation Sequencing Technologies. *Current Protocols in Molecular Biology* 122.1, e59.

Sontag, E. (2008). Network Reconstruction Based on Steady-State Data. *Essays Biochem* 45, pp. 161–176.

Sorrells, T. R. and Johnson, A. D. (2015). Making Sense of Transcription Networks. *Cell* 161.4, pp. 714–723.

Speth, Z., Islam, T., Banerjee, K., and Resat, H. (2017). EGFR Signaling Pathways Are Wired Differently in Normal 184A1L5 Human Mammary Epithelial and MDA-MB-231 Breast Cancer Cells. *Journal of Cell Communication and Signaling* 11.4, pp. 341–356.

Spieth, C., Streichert, F., Speer, N., and Zell, A. (2004). Iteratively Inferring Gene Regulatory Networks with Virtual Knockout Experiments. *Applications of Evolutionary Computing*. Ed. by G. R. Raidl, S. Cagnoni, J. Branke, D. W. Corne, R. Drechsler, Y. Jin, C. G. Johnson, P. Machado, E. Marchiori, F. Rothlauf, G. D. Smith, and G. Squillero.

Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 104–112.

Spirtes, P., Glymour, C. N., Scheines, R., and Heckerman, D. (2000). Causation, Prediction, and Search. Cambridge, MA: MIT Press. 576 pp.

Stein, R. R., Marks, D. S., and Sander, C. (2015). Inferring Pairwise Interactions from Biological Data Using Maximum-Entropy Probability Models. *PLOS Computational Biology* 11.7, e1004182.

Steinke, F., Seeger, M., and Tsuda, K. (2007). Experimental Design for Efficient Identification of Gene Regulatory Networks Using Sparse Bayesian Models. *BMC Systems Biology* 1.1, p. 51.

Stelniec-Klotz, I., Legewie, S., Tchernitsa, O., Witzel, F., Klinger, B., Sers, C., Herzel, H., Blüthgen, N., and Schäfer, R. (2012). Reverse Engineering a Hierarchical Regulatory Network Downstream of Oncogenic KRAS. *Molecular Systems Biology* 8.1, p. 601.

Stern, C. D. (2019). The 'Omics Revolution: How an Obsession with Compiling Lists Is Threatening the Ancient Art of Experimental Design. *BioEssays* 41.12, p. 1900168.

Steuer, R., Kurths, J., Daub, C. O., Weise, J., and Selbig, J. (2002). The Mutual Information: Detecting and Evaluating Dependencies between Variables. *Bioinformatics* 18 (suppl_2), S231–S240.

Stolovitzky, G., Monroe, D., and Califano, A. (2007). Dialogue on Reverse-Engineering Assessment and Methods. *Annals of the New York Academy of Sciences* 1115.1, pp. 1–22.

Stolovitzky, G., Prill, R. J., and Califano, A. (2009). Lessons from the DREAM2 Challenges. *Annals of the New York Academy of Sciences* 1158.1, pp. 159–195.

Stricker, S. H., Köferle, A., and Beck, S. (2017). From Profiles to Function in Epigenomics. *Nature Reviews Genetics* 18.1 (1), pp. 51–66.

Stuart, T. and Satija, R. (2019). Integrative Single-Cell Analysis. *Nature Reviews Genetics* 20.5, pp. 257–272.

Sugihara, G., May, R., Ye, H., Hsieh, C.-h., Deyle, E., Fogarty, M., and Munch, S. (2012). Detecting Causality in Complex Ecosystems. *Science* 338.6106, pp. 496–500.

Szederkényi, G., Banga, J. R., and Alonso, A. A. (2011). Inference of Complex Biological Networks: Distinguishability Issues and Optimization-Based Solutions. *BMC Systems Biology* 5.1, p. 177.

Székely, G. J., Rizzo, M. L., and Bakirov, N. K. (2007). Measuring and Testing Dependence by Correlation of Distances. *Annals of Statistics* 35.6, pp. 2769–2794.

Tang, F., Barbacioru, C., Wang, Y., Nordman, E., Lee, C., Xu, N., Wang, X., Bodeau, J., Tuch, B. B., Siddiqui, A., Lao, K., and Surani, M. A. (2009). mRNA-Seq Whole-Transcriptome Analysis of a Single Cell. *Nature Methods* 6.5 (5), pp. 377–382.

Tatsuya, A. (2018). Algorithms For Analysis, Inference, And Control Of Boolean Networks. Singapore: World Scientific. 227 pp.

Tegnér, J., Yeung, M. K. S., Hasty, J., and Collins, J. J. (2003). Reverse Engineering Gene Networks: Integrating Genetic Perturbations with Dynamical Modeling. *Proceedings of the National Academy of Sciences* 100.10, pp. 5944–5949.

Thomas, R. (1973). Boolean Formalization of Genetic Control Circuits. *Journal of Theoretical Biology* 42.3, pp. 563–585.

Thomas, R. and D'Ari, R. (1990). Biological Feedback. Raton, Fl: CRC Press.

Thomas, R., Gathoye, A.-M., and Lambert, L. (1976). A Complex Control Circuit. *European Journal of Biochemistry* 71.1, pp. 211–227.

Timme, M. (2007). Revealing Network Connectivity from Response Dynamics. *Physical Review Letters* 98.22, p. 224101.

Titeca, K., Lemmens, I., Tavernier, J., and Eyckerman, S. (2019). Discovering Cellular Protein-Protein Interactions: Technological Strategies and Opportunities. *Mass Spectrometry Reviews* 38.1, pp. 79–111.

Tkačik, G., Dubuis, J. O., Petkova, M. D., and Gregor, T. (2015). Positional Information, Positional Error, and Readout Precision in Morphogenesis: A Mathematical Framework. *Genetics* 199.1, pp. 39–59.

Todorov, H., Cannoodt, R., Saelens, W., and Saeys, Y. (2019). Network Inference from Single-Cell Transcriptomic Data. *Gene Regulatory Networks: Methods and Protocols*. Ed. by G. Sanguinetti and V. A. Huynh-Thu. Methods in Molecular Biology. New York, NY: Springer, pp. 235–249.

Tomita, M. (2001). Whole-Cell Simulation: A Grand Challenge of the 21st Century. *Trends in Biotechnology* 19.6, pp. 205–210.

Tong, A. H. Y. et al. (2004). Global Mapping of the Yeast Genetic Interaction Network. *Science* 303.5659, pp. 808–813.

Tringe, S. G., Wagner, A., and Ruby, S. W. (2004). Enriching for Direct Regulatory Targets in Perturbed Gene-Expression Profiles. *Genome Biology* 5.4, R29.

Uhlén, M. et al. (2015). Tissue-Based Map of the Human Proteome. *Science* 347.6220.

Vailati-Riboni, M., Palombo, V., and Loor, J. J. (2017). What Are Omics Sciences? *Periparturient Diseases of Dairy Cows: A Systems Biology Approach*. Ed. by B. N. Ametaj. Cham: Springer International Publishing, pp. 1–7.

Venter, J. C. et al. (2001). The Sequence of the Human Genome. *Science* 291.5507, pp. 1304–1351.

Verma, T. and Pearl, J. (1990). Equivalence and Synthesis of Causal Models. *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*. UAI '90. USA: Elsevier Science Inc., pp. 255–270.

Videla, S., Guziolowski, C., Eduati, F., Thiele, S., Gebser, M., Nicolas, J., Saez-Rodriguez, J., Schaub, T., and Siegel, A. (2015). Learning Boolean Logic Models of Signaling Networks with ASP. *Theoretical*

Computer Science. Advances in Computational Methods in Systems Biology 599, pp. 79–101.

Virtanen, P. et al. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 17.3 (3), pp. 261–272.

Vlastaridis, P., Kyriakidou, P., Chaliotis, A., Van de Peer, Y., Oliver, S. G., and Amoutzias, G. D. (2017). Estimating the Total Number of Phosphoproteins and Phosphorylation Sites in Eukaryotic Proteomes. *GigaScience* 6.2, pp. 1–11.

Volkov, I., Banavar, J. R., Hubbell, S. P., and Maritan, A. (2009). Inferring Species Interactions in Tropical Forests. *Proceedings of the National Academy of Sciences* 106.33, pp. 13854–13859.

Wagner, A. (2001). How to Reconstruct a Large Genetic Network from n Gene Perturbations in Fewer than N2 Easy Steps. *Bioinformatics* 17.12, pp. 1183–1197.

Wang, S., Karikomi, M., MacLean, A. L., and Nie, Q. (2019). Cell Lineage and Communication Network Inference via Optimization for Single-Cell Transcriptomics. *Nucleic Acids Research* 47.11, e66–e66.

Weber, A. P. M. (2015). Discovering New Biology through Sequencing of RNA. *Plant Physiology* 169.3, pp. 1524–1531.

Weigt, M., White, R. A., Szurmant, H., Hoch, J. A., and Hwa, T. (2009). Identification of Direct Residue Contacts in Protein–Protein Interaction by Message Passing. *Proceedings of the National Academy of Sciences* 106.1, pp. 67–72.

Wetterstrand, K. A. (2020). DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP). URL: www.genome.gov/sequencingcostsdata (visited on 05/17/2020).

Wilhelm, M. et al. (2014). Mass-Spectrometry-Based Draft of the Human Proteome. *Nature* 509.7502 (7502), pp. 582–587.

Wolkenhauer, O. (2001). Systems Biology: The Reincarnation of Systems Theory Applied in Biology? *Briefings in Bioinformatics* 2.3, pp. 258–270.

Wood, V., Lock, A., Harris, M. A., Rutherford, K., Bähler, J., and Oliver, S. G. (2019). Hidden in Plain Sight: What Remains to Be Discovered in the Eukaryotic Proteome? *Open Biology* 9.2, p. 180241.

Yamane, J., Aburatani, S., Imanishi, S., Akanuma, H., Nagano, R., Kato, T., Sone, H., Ohsako, S., and Fujibuchi, W. (2016). Prediction of Developmental Chemical Toxicity Based on Gene Networks of Human Embryonic Stem Cells. *Nucleic Acids Research* 44.12, pp. 5515–5528.

Yeung, M. K. S., Tegnér, J., and Collins, J. J. (2002). Reverse Engineering Gene Networks Using Singular Value Decomposition and Robust Regression. *Proceedings of the National Academy of Sciences* 99.9, pp. 6163–6168.

Zagorski, M., Tabata, Y., Brandenberg, N., Lutolf, M. P., Tkačik, G., Bollenbach, T., Briscoe, J., and Kicheva, A. (2017). Decoding of Position in the Developing Neural Tube from Antiparallel Morphogen Gradients. *Science* 356.6345, pp. 1379–1383.

Zhu, C., Preissl, S., and Ren, B. (2020). Single-Cell Multimodal Omics: The Power of Many. *Nature Methods* 17.1 (1), pp. 11–14.

Zou, C. and Feng, J. (2009). Granger Causality vs. Dynamic Bayesian Network Inference: A Comparative Study. *BMC Bioinformatics* 10.1, p. 122.

SELBSTSTÄNDIGKEITSERKLÄRUNG

Hiermit erkläre ich, dass ich diese Dissertation selbstständig und ausschließlich unter Verwendung der angegebenen Hilfsmittel und Quellen angefertig habe. Es fand dabei keine Zusammenarbeit mit gewerblichen Promotionsberatern statt. Ich habe die Promotionsordnung der Lebenswissenschaftlichen Fakultät vom 05. März 2015 zur Kenntnis genommen. Die Dissertation oder Teile davon wurden nicht bereits bei einer anderen wissenschaftlichen Einrichtung eingereicht, angenommen oder abgelehnt. Ich habe mich nicht anderwärts um einen Doktorgrad beworben und besitze keinen entsprechenden Doktorgrad. Die Grundsätze der Humboldt-Universität zu Berlin zur Sicherung guter wissenschaftlicher Praxis wurden eingehalten.

Berlin, June 2020

Torsten Groß