Modeling signal transduction pathways and their transcriptional response

Ewa Szczurek

Oktober 2010

Dissertation zur Erlangung des Grades
eines Doktors der Naturwissenschaften (Dr. rer. nat.)
am Fachbereich Mathematik und Informatik
der Freien Universität Berlin

1. Referent: Prof. Dr. Martin Vingron
2. Referent: Prof. Dr. Jerzy Tiuryn

Tag der Promotion: 29 April 2011

Preface

Acknowledgements I am grateful to my supervisor Martin Vingron for his scientific support, and his wise advice and help whenever I was in trouble. Martin and Jerzy Tiuryn were my true mentors and I am deeply grateful for the chance of working under their guidance. I thank Alexander Bockmayr for fruitful discussions and the opportunity to assist him in his teaching class. I am indebted to Alexander, Jerzy and Martin for their counsel as members of my PhD committee.

My very special thanks go to Irit Gat-Viks, who in many ways influenced my scientific thinking and shared her advice both as a collaborator and as a friend. I learned a lot during my work with Przemysław Biecek and with Florian Markowetz. My biological interests were largely shaped by the collaboration with Christine Sers and Andrea Solf, who deepened my knowledge with patient explanations and discussions. I owe my gratitude to Julia Lassere for sharing her expertise on mixture models. During my doctorial work I benefitted from discussions with Roman Brinzanik, Szymon Kiełbasa, and Christine Steinhoff. I want to acknowledge my dear office mates, Marta Łuksza, Matthias Heining and Marcel Schulz in Berlin as well as Bogusław Kluge, Mikołaj Rybiński and Maciej Sykulski in Warsaw, for all enjoyable moments of work and fun together.

This thesis was proof-read and greatly improved by Marcin Grynberg (chapter 1) Przemysław Biecek (chapter 2), Marta Łuksza (chapter 3), Ruben Schilling (chapter 3), Matthias Heinig (chapter 4), and Roman Brinzanik (chapters 1 and 5). The German abstract of the thesis (Zusammenfassung) was prepared with the help of Roman Brinzanik, Kirsten Kelleher, Roland Krause and Florian Markowetz.

I thank Staś Szczurek for comments on the introduction to the thesis, and for his encouragement and constant support of my scientific ambitions throughout the entire course of doctorial work.

Publications Chapters of this thesis grew out of papers that I worked on during my doctorial studies. Chapter 2 was published in Journal of Computational Biology [118]. Chapter 3 appeared in Molecular Systems Biology [117], and chapter 4 is now submitted.

Ewa Szczurek Berlin, Oktober 2010

Contents

Chapter 1

Introduction

This thesis is concerned with revealing regulation of gene expression. The basic motivation behind our work is that gene regulation can be better resolved when analyzed in a cellular context of the upstream signaling pathway and known regulatory targets. Our source of data are perturbation experiments, which are performed on pathway components and induce changes in gene expression. In such a way, they connect the signaling pathway to its downstream target genes. This chapter starts with an introduction to the cellular context considered in the thesis (section 1.1) and the principles of perturbation experiments (section 1.2). We end with a concise summary of three approaches that comprise this thesis. The approaches tackle various problems in the process of revealing context-specific regulatory networks (section 1.3). We deal with differential expression analysis of the perturbation data, enhanced with known transcription factor targets serving as examples of differential genes (chapter 2), pathway model-based planning of informative perturbation experiments (chapter 3), and finally, with deregulation analysis, i.e., comparing changes in gene regulation between two different cell populations (chapter 4).

1.1 Signaling pathway and downstream gene regulation

Basic biological notions Following a comprehensive book by Alberts *et al.* [3], we shortly introduce the basic components and processes present in the cell that are important for this thesis. We analyze eukaryotic cells of yeast and human. A simplified scheme of an eukaryotic cell in Fig.1.1 A presents the surrounding plasma membrane, the interior cytoplasm and nucleus in the center (we ignore other organelles). The nucleus stores a *deoxyribonucleic acid (DNA)* molecule, which is the cellular carrier of genetic information, inherited by the daughter of a dividing cell in a process of replication. Chemically, DNA is a helical structure built from two long polymers, called *DNA strands*. Each strand is composed of a sequence of four basic molecules, called *nucleotides*: adenine, guanine, cytosine and thymine. Each nucleotide on one strand forms a bond with only one other nucleotide on the other strand. This is called sequence *complementarity*. Within the nucleus, DNA is organized into long

structures called *chromosomes*. This *chromosomal DNA* is differentiated from separate DNA molecules in the cell, like *plasmids*. A plasmid is a double-stranded DNA molecule, which is not part of the chromosomal DNA, but it can survive and be replicated independently in the cell.

The central dogma of molecular biology postulates that portions of the chromosomal DNA, called *genes*, serve as templates for production of *messenger RNA* (mRNA). An enzyme called RNA polymerase *transcribes* the sequence of the gene into the mRNA sequence. The amount of mRNA defines the level of gene *expression*. Experiments performing thousands of simultaneous measurements on a population of cells are called *high-throughput*. For example, we analyze data from high-throughput gene expression experiments, also called *genome-wide* experiments. The process of *translation* utilizes the mRNA sequences to produce *proteins*. Proteins play a role in almost all processes in the cell.

Signaling pathways The cell membrane acts as a filter to the outside environment, transmitting selected stimulatory cues. Examples of such *stimulation* are hormones, growth factors, cytokines or chemokines. The stimulation may also come from the inside of the cell. Stimulation is recognized by *receptors*, which include G-protein coupled receptors (recognizing e.g., chemokines) or receptor tyrosine kinases, (e.g., growth factor receptors), and many other types. Activated receptors in turn induce activation of a *signaling pathway*, which conveys the signal further through a cascade of interactions between cellular molecules. In this thesis, we focus on a broad class of signaling pathways with protein components, which regulate each other's activity, for example by phosphorylation. We say that the regulating and the regulated proteins are in a *signaling relation*. The signal is commuted down to a special kind of proteins: *transcription factors*, which then regulate expression of genes. Therefore, transcription factors (abbreviated TFs throughout the text) are the biological connection between the signaling pathway and the genes. Fig.1.1 A presents a simple signaling pathway with three components, A, B and C in the cytoplasm, A and C being TFs. Four exemplary genes g_1–g_4 are shown in the nucleus, with the TF A regulating g_1 and g_3. We say that the gene regulation occurring due to activation of a certain signaling pathway happens *downstream* of the pathway and determines the response of the cell to the signal. The signaling pathway is said to be *upstream* of this gene regulation. Below we describe the details of this process.

Transcription factors Transcription factors control expression of genes by recognizing and binding to specific sequences of DNA, called *binding motifs*, in the *promoter* or *enhancer* regions of the genes. Those regions are portions of the DNA, placed adjacent or distant to the gene, respectively, and are together called *regulatory regions*. Binding of TFs to regulatory regions influences recruitment of RNA polymerase to the gene. The control of RNA polymerase recruitment is not due to the TF alone, but requires involvement of a complex of many other proteins. Transcription factors are distinguished from the other members of this complex by domains,

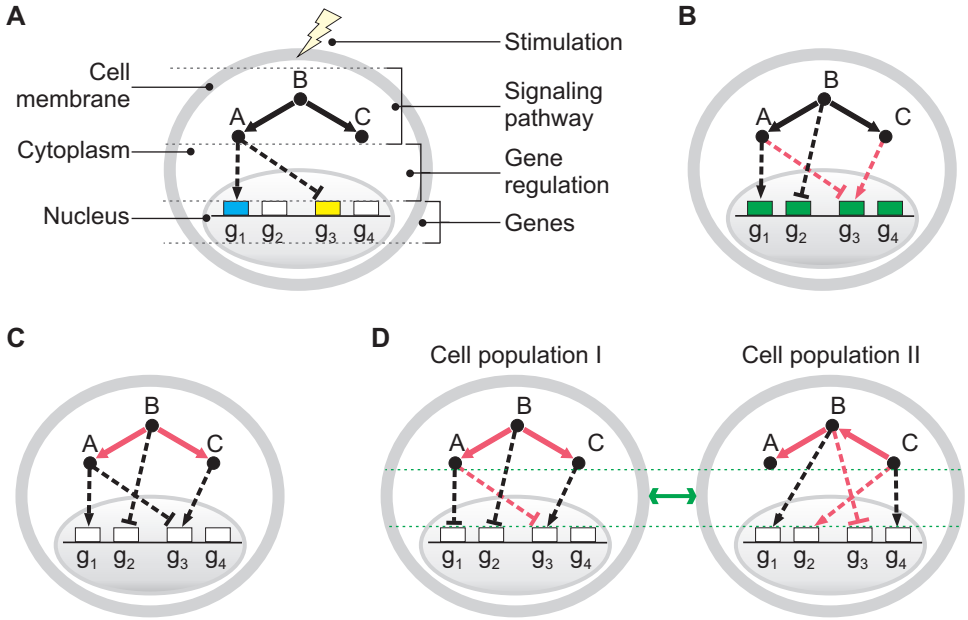


Figure 1.1: Problems in resolving context-specific gene regulation. (**A**) The cellular context of gene regulation. The oval is a schematic representation of an eukaryotic cell with a signaling pathway (solid edges) and its downstream regulatory relations (dashed edges, pointed arrows indicating activation and blunted arrows indicating inhibition). In our studies, we deal with populations of such cells. ΔA – an experiment, where A is perturbed. Genes are colored according to the effect of the perturbation. In this example, the perturbation is a knockout or a knockdown, where the regulator A is repressed. In such a case gene g_1, which is activated by A, is down-regulated by the perturbation (colored in blue), where as gene g_3, which is inhibited by A, is up-regulated (colored in yellow). Genes not dependent on A do not change their expression (marked in white). (**B**–**D**) Problems solved in chapters 2–4. In each problem, the question is illustrated with green items. Items marked in red represent parts of the cellular context which are known and given as input. Black items are part of the context but are neither known nor asked. (**B**) Partially supervised differential expression analysis of the perturbation data (chapter 2). For some of the TFs their regulatory relations are known (e.g., here we know that A inhibits g_3 and C activates g_3). We are given data from each performed perturbation experiment. The task is to analyze the data and correctly identify up- and down-regulated genes. Note that here the pathway structure is ignored. (**C**) Planning of pathway-targeted perturbation experiments (chapter 3). The regulatory relations in the pathway are given. The question is which perturbation experiments to perform in order to unambiguously recover downstream regulatory relations. (**D**) Deregulation analysis (chapter 4). We consider two different cell populations, knowing both their pathway topologies, and the regulatory relations for selected TFs. We are given data from perturbations of each pathway component in both cell populations. Our aim is to characterize changes in gene regulation that occur between the populations.

which enable binding to the DNA in the regulatory regions. *Combinatorial* regulation happens when several TFs bind together and collaborate to control expression of a specific gene.

A gene regulated by a given TF is referred to as its *target gene*. We say that the gene and the TF are in a *regulator-target relation* or, shortly, in a *regulatory relation* (as we shall see in section 1.2, also proteins in the signaling pathway may act as regulators). The actual effect the activity of a TF has on its target gene's expression is described by a *regulatory mechanism*. A bound TF may act as gene expression *activator* or *inhibitor*, by either promoting or repressing the RNA polymerase. There are different ways in which activation or inhibition is carried out, either by a single, or by several TFs. Thus, the regulatory relations state "who regulates who" and the regulatory mechanisms state "how".

Gene regulatory networks In this thesis, a *gene regulatory network* refers to regulatory relations for a set of TFs and their target genes. Note that in general, a gene activated by a given TF may code for a TF itself and regulate transcription of many other genes. Regulatory networks show such hierarchy of regulatory relations.

1.2 Pathway-targeted perturbation experiments

Perturbation experiments considered in this work are molecular interventions in the form of single gene *knockout*, *knockdown* or *overexpression*, combined with high-throughput gene expression measurement (referred to as *perturbation data*). Gene perturbation changes expression of the gene: knockdown and (more drastically) knockout decrease its expression level, while overexpression increases it. The expression of genes which are either directly or indirectly regulated by the perturbed gene also changes. To determine this effect, the accompanying genome-wide expression measurement is always a comparison between the populations of perturbed and normal cells. In chapters 2 and 3 we analyze perturbation data from yeast, where compendia of knockouts [92, 57] or overexpressions [21] of multiple genes are available. In chapter 4 we work with knockdown data in human cells [32].

Basic experimental techniques [3] Most perturbation experiments are carried out with the use of *plasmid vectors*. Plasmid vectors are DNA sequences artificially engineered from natural plasmids that occur in bacterial cells or in viruses. They contain DNA sequence fragments, inserted into the plasmid at wish of the researcher. Plasmid vectors, which serve as means to introduce a specific gene into target cells are called *expression vectors*. Expression vectors carry inserted DNA fragments coding for the gene itself and for its promoter region, from which expression of the gene may be very efficiently controlled.

The desired DNA fragments can be introduced into the plasmid by a cutting and sealing machinery of the cell, carried out by the *restriction endonucleases* and *DNA ligases*, respectively. Restriction nucleases are bacterial enzymes, which cut the DNA at specific sites, called *recognition sites*, defined by a local sequence of nucleotides. Some of the nucleases make "uneven" cuts, leaving single-stranded tails hanging at the end of each fragment. Those tails are called *cohesive ends*, as they can bind to any other complementary tail produced by the restriction nuclease. To insert a specific DNA fragment into the plasmid, both the inserted fragment and the plasmid have to have recognition sites for the same restriction nuclease. First, the fragment has to be cut out from a larger portion of DNA using the restriction nuclease. Second, the restriction nuclease has to cut the plasmid. Finally, the complementary cohesive ends of the plasmid are bound by complementarity with the ends of the DNA fragment. The resulting recombinant molecule is then covalently sealed together by the DNA ligase.

To increase the efficiency of inserting the desired DNA fragment into the plasmid, the fragment can be multiplied via a polymerase chain reaction (PCR). PCR, carried out *in vitro*, generates an exponential number of copies of the given DNA fragment. First, two short DNA sequences, called *primers*, which flank the fragment are identified. Two sets of these primers are next synthesized by chemical methods. The DNA is then heated in order to separate its strands. Primers bind to ends of the cloned DNA fragment on both strands, thereby initiating synthesis of complementary strands by a DNA polymerase. These steps of DNA synthesis are repeated in rounds, each round doubling the DNA fragments serving as templates for the next (hence the term "chain reaction").

Insertion of a plasmid vector into the target cell is called *transfection*. Accordingly, the target cells are called *transfected*. Once the vector is placed inside the cell, cellular machinery takes up its replication, and, in case of expression vectors, expression of the carried gene and production of the protein. To select for transfected cells in a larger mixture, the mixture is treated with a *selective agent*. The selective agent is a substance normally able to kill or suppress cellular growth. An example of selective agent used for bacterial cells is an antibiotic, to which the cells are sensitive. The selection is made possible by a *selectable marker gene* carried by the plasmid vector. The selectable marker (e.g., a gene for the antibiotic resistance) protects the cells which contain the vector from the selective agent. From the mixture of cells, only those that either took up the plasmid vector, or inherited it, survive the treatment with the selective agent.

Before transfecting the final target cells, (e.g. yeast), the vectors may be replicated to obtain their multiple copies in bacterial cells. Bacterial cells, which are competent to accept exogenous DNA are transfected with the plasmid vector. Next, in a natural process of bacterial growth and division, the inserted plasmids are replicated. As a result, the number of plasmid vector copies may be doubled every 30 minutes. Finally, the resulting mixture of cells is treated with a selective agent to select for those bacterial cells, which contain a copy of the plasmid.

Knockout A gene knockout is a genetic mutation method in which the gene is forced to loose its function. A knockout experiment can be carried out in two ways:

1. *Gene deletion* Gene deletion is a construction of a mutant cell that is missing the gene. To establish a gene deletion mutant in yeast [131] the following procedure is employed: First, short regions of DNA surrounding the perturbed gene are identified. In the next step, an expression vector is constructed, which contains the identified surrounding regions and a selectable marker gene in between. Finally, this expression vector replaces the perturbed gene in the yeast genome in a natural cellular process of *homologous recombination*. In short, the replacement is enabled by the fact that the regions at the ends of the selectable marker gene on the vector match the regions surrounding the original gene to be deleted. As a result, more than 95% of the resulting yeast cells have the expression vector in place of the perturbed gene. As a selective agent is added to the pool of all cells, only the ones which carry the deletion remain.
2. *Promoter replacement* In contrast to gene deletion, the promoter replacement experiment maintains the gene itself in the DNA. An expression vector carrying a promoter sequence, which enables easy control of the gene's expression is placed instead of the original gene promoter in the genome. In yeast, a tetracycline-regulatable promoter is applied [39], and gene repression can be controlled by addition of doxycycline (a member of the tetracycline antibiotics group) to the growth medium.

Overexpression To overexpress a gene in yeast, expression vectors containing this gene and an easily controllable promoter region are utilized. Exemplary expression vectors [143, 142] contain the promoter of a gene GAL1 incorporated alongside the sequence of the gene. The promoter of GAL1 is induced and starts transcription of the nearby gene at high rate in the presence of galactose, and is shut down, repressing transcription, in a glucose medium. Galactose induction results in an intensive production of the protein coded by the gene carried by the expression vectors.

Knockdown Gene knockdown is a perturbation technique which reduces the expression of the perturbed gene by a mechanism other than genetic modification. Here we discuss gene knockdown experiments, which degrade the gene's mRNA transcript, exploiting the process of *RNA interference (RNAi)* [48]. RNAi utilizes a double-stranded RNA (dsRNA) with a sequence similar to the gene to be knocked down. Once the dsRNA enters the cell, a RNAi pathway proceeds in four steps. First, an enzyme named Dicer recognizes and cleaves the long dsRNA molecules into short fragments of around twenty nucleotides, called *short interfering RNAs (siRNAs)*. In each resulting fragment, one of the two strands can be distinguished as the *guide strand*. In the second step, the guide strand is incorporated into so-called *RNA-induced silencing complex (RISC)*. In the third step this strand "guides" RISC to the mRNA that is transcribed from the gene to be knocked down. To this end, the guide strand binds by complementarity to the mRNA molecule. Knockdown of the gene

is obtained in the fourth step by cleavage of its mRNA molecule, carried out by the catalytic component of the RISC complex called Argonaute.

In mammalian cells introduction of dsRNA may result in an anti-viral interferon response, disturbing protein synthesis in the cell [9]. To avoid this problem, synthetic siRNAs, already around twenty nucleotides in length, can be used instead. Finally, transfecting the cells with vectors expressing so called *small hairpin RNAs (shRNAs)* has been shown to induce gene knockdown effects on longer time scales. The name comes from the fact that the structure of shRNAs makes a sharp hairpin turn. Such expression vectors are equipped with easily controllable promoters (e.g., the tetracycline-regulated U6 promoter), which ensure that the shRNAs are abundantly expressed [91].

Basis for regulatory network reconstruction In our work, perturbation experiments are at focus because of their applicability in the task of gene regulatory network reconstruction [86, 85, 121]. The basic principle is that genes, being in a regulatory relation with a TF, *respond* by showing an effect in their expression when this TF is perturbed.

Importantly, transcriptional effects of perturbation are observed also on target genes that are indirectly connected to the perturbed gene through a series of direct signaling or regulatory relations. When the perturbed gene is not a TF, but codes for any component of a signaling pathway, its perturbation has an effect first on TFs and next on the target genes downstream of the pathway. In general, in this thesis we define a set of *regulators*, which is the subset of all pathway components and represents proteins having a direct or indirect transcriptional control over response of target genes. In this generalized view we assume that the regulators may be in a regulatory relation with a target gene and control its expression via the same palette of regulatory mechanisms as TFs (see section 1.1).

Table 1.1 summarizes the expected effect of a target gene depending on the type of perturbation of the regulator and on the type of regulatory mechanism (here, for two simple mechanisms of activation and inhibition) controlling the gene's expression. The possible effects are assessed by comparison of expression level of the target gene upon the regulator perturbation with the expression level in wild-type cells. The effect can either be *down-regulation* of the gene (when its expression goes down compared to wild-type), or *up-regulation* (expression goes up), or, finally there can be no effect (the expression is unchanged in comparison to wild-type). For example, as illustrated in Fig. 1.1 A, if the TF A is knocked-out, its activated target gene g_1 is down-regulated. More advanced regulatory mechanisms are considered in chapter 3.

Perturbation of a regulator	Regulatory mechanism	Perturbation effect on the target gene
knockout or knockdown	activation	down-regulation
	inhibition	up-regulation
overexpression	activation	up-regulation
	inhibition	down-regulation

Table 1.1: Summary of perturbation effects depending on the type of regulator perturbation and regulatory mechanism. Such effects are expected for the target genes of the regulator which is perturbed.

1.3 Interconnected problems faced in this thesis

The primary goal of this thesis is discovering regulatory relations, taking into account available knowledge about their *cellular context*: the upstream signaling pathway and TF targets (Fig. 1.1 A). In chapters 2–4 we tackle three different problems, each dealing with different aspects of this general goal.

Differential expression analysis with examples High-throughput gene expression experiments allow for a comparison between two different experimental conditions. The measurements need to be analyzed statistically in order to determine sets of genes that are up-, or down-regulated, or unchanged in a chosen condition. Researcher's expertise, often based on literature knowledge and experimental intuition, can suggest examples of genes which may belong to one of these sets. Established differential expression analysis tools [27, 113, 114] do not take such imprecise examples into account. In chapter 2 we put forward a novel methodology that systematically incorporates imprecise knowledge into differential expression analysis. We use *partially supervised mixture modeling* that separates one-dimensional expression data into clusters of differentially expressed and unchanged genes, and utilizes imprecise examples to find these clusters.

The proposed methodology is of special importance for the analysis of perturbation data. Here, the sets of genes that are up-regulated, down-regulated and unchanged upon the perturbation are interpreted as genes that are inhibited, activated, or not dependent on the perturbed gene, respectively. Researchers can often provide examples from the sets of up-/down-regulated, or unchanged genes in the analyzed experiment. This knowledge is rarely certain and can rather be quantified in distributions over those sets. Fig. 1.1 B presents the setup of the problem presented in the cellular context assumed in this thesis. In a particular cell population under study, some of the TFs may be believed to bind promoters and regulate some of the genes. Expression of such genes is expected, but not sure, to change after their believed transcription activator is knocked out. The methodology introduced in chapter 2 is an important step towards utilizing this knowledge for the reconstruction of the remaining regulator-target relations.

Planning perturbation experiments In chapter 3, we introduce an algorithm called MEED (*model expansion experimental design*). MEED is meant to guide experimentalists who focus their research on a chosen signaling pathway and are interested in the regulation of its downstream targets. We assume that the researcher has initial qualitative knowledge about the signaling relations in the studied pathway and wishes to systematically perturb the pathway components to characterize the response of the downstream target genes. In contrast to large compendia of perturbation data, such experimental studies [104, 138, 97] are focused on perturbing a specific signaling system to infer its downstream regulation mechanisms. Gat-Viks and Shamir [41] improve this inference using a formal model of the perturbed pathway in their approach called *model expansion*.

All these approaches heavily depend on which and how particular pathway components were perturbed. In chapter 3 we bring up and tackle the problem of *ambiguity* in the identification of regulatory relations. For example, it is possible that a TF is not affected and remains inactive in all experiments and therefore its targets cannot be revealed. Alternatively, consider two TFs located in different parts of the signaling pathway, with a different role and different target genes. In a given set of experiments, if their target genes have similar expression profiles, they will be falsely considered as co-regulated. Moreover, taking any of the two TFs as the common regulator of these targets will be equally supported by the experimental data, leading to ambiguous hypothesis about their transcriptional regulation. To avoid such problems, the experiments must generate enough information to draw unambiguous conclusions about regulatory relations.

Fig. 1.1 C presents the biological setup of the problem solved in chapter 3. Here, we know the relations between components of the upstream signaling pathway and we want to know which perturbation experiments to perform. Given a model of the pathway, the MEED algorithm aims to select the smallest number of experiments, which together allow for unambiguous identification of regulatory relations downstream of the pathway. In the end, the experiments designed using MEED are used in a model expansion procedure. Building on ideas of Gat-Viks and Shamir [41], the procedure reconciles experimental data with model predictions to elucidate the regulatory relations downstream of the given pathway model.

Deregulation analysis In chapter 4 we put forward an approach for *joint deregulation analysis*, abbreviated JODA. Our aim is to delineate *deregulation*, defined as changes in gene regulation between two different populations. Extant deregulation analysis approaches [84, 120, 34, 56, 134] do not take the cellular context of these changes into account.

JODA combines cell-specific perturbation data and knowledge presented in Fig. 1.1 D. The data comes from perturbation experiments that need to be performed on the same genes in both cell populations. We assume that knowledge about the cellular context of gene regulation is given by: signaling relations in the upstream pathway and established relations between the TFs in the pathway and their target genes. This

cellular context is provided for both cell populations. The approach combines ideas introduced in the previous chapters. The known TF targets are utilized as examples of up- or down-regulated genes in the partially supervised differential expression analysis of the perturbation data (chapter 2). Information about the topology of the signaling pathways active in the two cell populations is formalized in two simple models. Next, the models are used for reconstruction of regulatory relations as described in chapter 3.

1.3.1 Software

Our partially supervised mixture modeling approach is implemented in an `R` package `bgmm`, freely available from `http://bgmm.molgen.mpg.de`, together with the data used for the analysis presented in the thesis. The package provides practical support in the application of our methodology to differential data analysis.

The MEED framework software is freely available from `http://meed.molgen.mpg.de/`. The software supports:

- building a logical model of the signaling pathway under study, and using it to provide predictions for a set of candidate experiments,
- selecting perturbation experiments on the pathway components from the set of candidates,
- elucidating gene regulation downstream of the pathway.

The steps of the JODA algorithm are implemented and available in an `R` package `joda`, available from `http://joda.molgen.mpg.de`.

Chapter 2

Introducing knowledge into differential expression analysis

This chapter discusses our novel knowledge-based methodology for differential expression analysis. The approach is implemented by two partially supervised mixture modeling methods: a newly introduced belief-based *modeling, and* soft-label *modeling, a method proved efficient in other applications. Our methodology benefits from knowledge about genes that should be up- or down-regulated in the analyzed expression data. To introduce the theory, we bring together variants of utilizing labeled data by extant mixture modeling methods, including the soft-label method (section 2.1). Next, we describe our belief-based modeling (section 2.2). To introduce the application, we first cover existing mixture model-based methods for differential expression analysis (section 2.4). Next, we show how the soft-label and belief-based methods can be applied for this task (section 2.5). In section 2.6, the performance of the two partially supervised methods is validated on synthetic data. Finally, we show three applications of the methods to gene expression data: first, identification of targets of Ste12 from knockout data in yeast, given knowledge from a Ste12 DNA-binding experiment (section 2.7); second, distinguishing miR-1 from miR-124 human target genes based on expression data from transfection experiments of either microRNAs, with the use of their predicted targets (section 2.8); third, clustering of cell cycle genes based on their time-course expression profiles (section 2.9).*

2.1 Mixture modeling variants: the aspect of incorporating knowledge

In the problem of clustering, a dataset of observations $X = \{x_1, ..., x_N\}$ is given, and one looks for an assignment of the observations to clusters in $\mathcal{Y} = \{1, ..., K\}$. In this thesis we assume that the number of clusters K is known, and that the data points $x_i \in X$ are one-dimensional. In our application the clusters correspond to differentially expressed (shortly, *differential*) or unchanged genes, and data consist of expression ratios comparing measurements from two conditions. To find the clusters, mixture modeling is applied. Mixture modeling associates each cluster with a model component, which is defined by an underlying distribution estimated from the data.

Mixture modeling variants differ in the way they utilize additional knowledge. We assume the knowledge is available for a subset of first M observations $\{x_1, ..., x_M\}$, called *examples*. The knowledge about an example can either be precise and give exactly one cluster the example belongs to, or can be imprecise and described by a probability distribution over the clusters in $\mathcal{Y}$. The precisely assigned cluster or the most probable cluster for an example is also called a *label*, and the examples are also referred to as *labeled data*.

Mixture modeling assumes that the cluster labels are realizations of random variables $Y_1, ..., Y_N$ that take values in $\mathcal{Y}$ and follow a multinomial distribution $M(1, \pi_1, ..., \pi_K)$, so $\pi_k = P(Y_i = k)$, for $i \in \{1, ..., N\}$ and $k \in \mathcal{Y}$. The π_ks are called *mixing proportions*, or *priors*, and satisfy $\sum_{k=1}^{K} \pi_k = 1$. The observations in X are assumed to be generated by continuous random variables $X_1, ..., X_N$ with values in $\mathcal{R}$ and a conditional density function $f(x_i|Y_i = k) = f(x_i; \theta_k)$, where $i \in \{1, ..., N\}$, $k \in \mathcal{Y}$, while θ_k denotes the parameters of the density function. We are concerned with Gaussian mixtures, where $\theta_k = (\mu_k, \sigma_k^2)$. The model parameters, denoted $\Psi = \{\pi_1, ..., \pi_K, \theta_1, ..., \theta_K\}$, are usually estimated from the data.

Unsupervised mixture modeling In unsupervised modeling, no cluster labels are known for the input data $X = \{x_1, ..., x_N\}$. Fig.2.1 A shows a graphical representation of this model. Model parameters are estimated by maximizing the log likelihood of the data given the model, referred to as the *incomplete data* likelihood:

$$l(\Psi, X) = \sum_{i=1}^{N} \log \Big(\sum_{k=1}^{K} \pi_k f(x_i; \theta_k) \Big). \tag{2.1}$$

To estimate the model parameters, the Expectation Maximization (EM) algorithm [29, 144] is applied. The algorithm starts by initializing the parameters in step 0. Next, the E and M steps are iterated until stop criteria are met. The standard stop criteria are given by user-defined parameters: a (small) interval ϵ and a (large) number Q. The iterations stop either when the consecutive incomplete likelihood values differ by less than ϵ or when the number of iterations exceeds Q. In the E step of the $(q+1)-$th iteration we compute the *posterior probabilities* for each data point x_i to belong to cluster k:

$$t_{ik}^{(q+1)} = \frac{\pi_k^{(q)} f(x_i; \theta_k^{(q)})}{\sum_{k'=1}^{K} \pi_{k'}^{(q)} f(x_i; \theta_{k'}^{(q)})}. \tag{2.2}$$

In the M step we update the parameters, assuring that with the new values the incomplete likelihood will be higher than in the previous step. For the mixing proportions and the Gaussian parameters the update formulas are:

$$\pi_k^{(q+1)} = \sum_{i=1}^{N} t_{ik}^{(q+1)} / N, \tag{2.3}$$

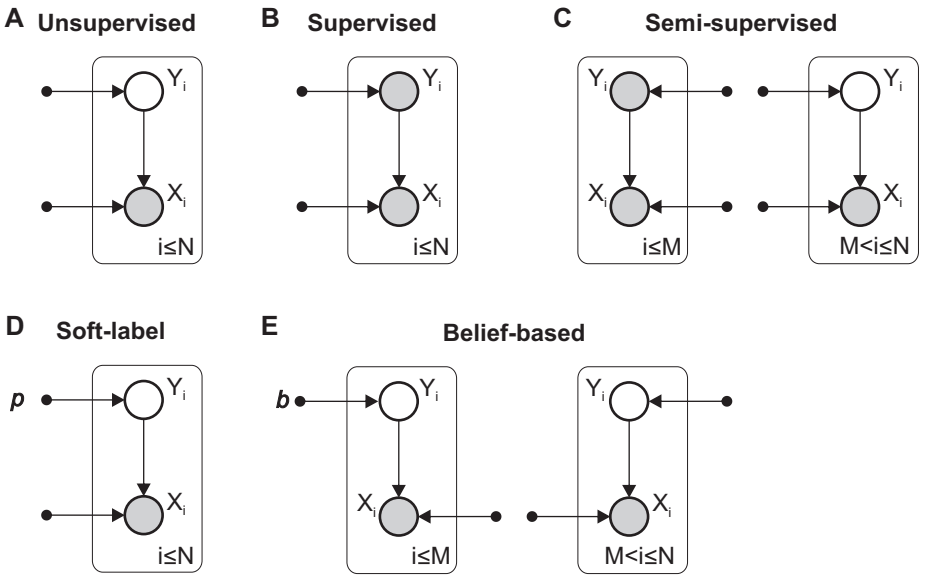


Figure 2.1: Graphical representation of mixture model variants discussed in this chapter. Graphical model representation [16] illustrates random variables (or sets of random variables) as open nodes, and parameters as small solid nodes. Here, $\theta = \{\theta_1, ..., \theta_K\}$ denotes the set of Gaussian parameters for all components, $\pi = \{\pi_1, ..., \pi_K\}$ denotes the set of mixing proportions, $b = \{b_1, ..., b_K\}$ the set of beliefs, and $p = \{p_1, ..., p_K\}$ the set of plausibilities. Apart from user-defined b and p, all parameters are estimated from the data. Directed edges point either from nodes corresponding to variables on which the distribution of the target node is conditioned, or from the parameters of the target node's distribution. Large rounded box called a *plate* denotes a set of nodes, with one of them shown explicitly. The set of nodes is defined with the running index indicated with a label in the lower part of the plate. Here, the index i always satisfies $i \geq 1$. Shaded nodes represent random variables that are set to their observed values. (**A**) The unsupervised mixture model, where all variables $\{Y_1, ..., Y_N\}$, representing cluster labels assigned to the data points, are not known. (**B**) The fully supervised mixture model, with all label variables set to their known values. (**C**) The semi-supervised mixture model, with the variables Y_i $(i \leq M)$, representing cluster labels assigned to the examples, set to their known values, and the remaining variables $(i > M)$ not known. (**D**) The soft-label mixture model, with all label variables not known, but with their prior weighted by the plausibilities. (**E**) Graphical representation of the belief-based mixture model, with all label variables not known, but with priors for the example label variables Y_i $(i \leq M)$ changed to their belief values.

$$\mu_k^{(q+1)} = \big(\sum_{i=1}^{N} x_i t_{ik}^{(q+1)}\big)/\big(\sum_{i=1}^{N} t_{ik}^{(q+1)}\big), \tag{2.4}$$

$$(\sigma^2)_k^{(q+1)} = \big(\sum_{i=1}^{N} t_{ik}^{(q+1)}(x_i - \mu_k^{(q+1)})^2\big)/\big(\sum_{i=1}^{N} t_{ik}^{(q+1)}\big). \tag{2.5}$$

For parameter initialization two procedures are applied. First, EM algorithm can be run many times with random initial parameters, possibly reaching different local incomplete likelihood maxima. Second, in the case of multivariate data, initial parameters can be computed from clusters obtained by hierarchical pre-clustering of the data. Univariate data can simply be divided into quantiles [137].

Fully supervised mixture modeling In the fully supervised variant, at input all observations have precise labels, as represented in a graphical form in Fig.2.1 B. The input dataset can be defined as $X^s = \{(x_1, z_1), ..., (x_N, z_N)\}$, where for observation x_i the function z_i, given as argument cluster k, returns value 1 if $Y_i = k$, and value 0 otherwise. We denote this value z_{ik}. Given the z_i functions, the log likelihood, called here the *complete data* likelihood, can be written as:

$$l(\Psi, X^s) = \sum_{i=1}^{N}\sum_{k=1}^{K} z_{ik} \log\big(\pi_k f(x_i; \theta_k)\big). \tag{2.6}$$

In the fully supervised case, it is easy to give the maximum likelihood estimates of the model parameters. For a mixture of Gaussians, we simply calculate the mean of all observations that are in each cluster k, $\bar{\mu}_k = \big(\sum_i x_i z_{ik}\big)/\big(\sum_i z_{ik}\big)$, their variance $\bar{\sigma}_k^2 = \big(\sum_i z_{ik}(x_i - \bar{\mu}_k)^2\big)/\big(\sum_i z_{ik}\big)$ and their number in proportion to the number of all observations $\bar{\pi}_k = \sum_i z_{ik}/N$. McLachlan and Peel [89] as well as Zhu and Goldberg [144] provide more details about the fully supervised mixture modeling.

Semi-supervised mixture modeling In the semi-supervised mixture modeling variant (Fig.2.1 C), we know the precise labels for the first M observations. Therefore the likelihood for the input set $X^{ss} = \{(x_1, z_1), ..., (x_M, z_M), x_{M+1}, ..., x_N\}$ is a mixture of the complete (Eq.2.6) and the incomplete (Eq.2.1) log likelihoods [144]:

$$\begin{aligned} l(\Psi, X^{ss}) &= \sum_{i=1}^{M}\sum_{k=1}^{K} z_{ik} \log\big(\pi_k f(x_i; \theta_k)\big) \\ &+ \sum_{i=M+1}^{N} \log\big(\sum_{k=1}^{K} \pi_k f(x_i; \theta_k)\big). \end{aligned} \tag{2.7}$$

Accordingly, in the E step of the EM algorithm, the posterior probabilities are obtained by setting $t_{ik} = z_{ik}$ for examples ($i \in \{1, ..., M\}$), and using Eq.(2.2) for the

remaining observations. Having this, the update equations in the M step are the same as in Eq.(2.3–2.5).

Soft-label mixture modeling Soft-label mixture modeling was recently introduced in machine learning by Come *et al.* [22] and shown to improve model-based clustering of general benchmark datasets. It formulates the given imprecise knowledge with belief functions [110]. In our application, each observation is labeled with a single cluster. In general, the soft-label method allows labels defined as subsets of clusters. Therefore, we consider only a particular case in their approach. In this case, the input dataset is defined as $X^p = \{(x_1, p_1), ..., (x_N, p_N)\}$, where for an example x_i $(i \leq M)$, a *plausibility* p_{ik} for each cluster k is given, satisfying $\sum_{k=1}^{K} p_{ik} = 1$. For the remaining observations $(i > M)$ it is assumed that this distribution is uniform, i.e., $p_{ik} = 1/K$. Come *et al.* use the plausibilities to weight the priors. This model variant is represented in Fig.2.1 D. In this case, the log likelihood for the input dataset reads:

$$l(\Psi, X^p) = \sum_{i=1}^{N} \log \big(\sum_{k=1}^{K} p_{ik} \pi_k f(x_i; \theta_k) \big). \tag{2.8}$$

Therefore, in the E step of the EM algorithm we compute:

$$t_{ik}^{(q+1)} = \frac{p_{ik} \pi_k^{(q)} f(x_i; \theta_k^{(q)})}{\sum_{k'=1}^{K} p_{ik'} \pi_{k'}^{(q)} f(x_i; \theta_{k'}^{(q)}).} \tag{2.9}$$

The update equation for the mixing proportion in the M step reads:

$$\pi_k^{(q+1)} = \sum_{i=1}^{N} t_{ik}^{(q+1)} / N, \tag{2.10}$$

i.e., the mixing proportions are computed based on the posterior probabilities of all data points, including the examples. The Gaussian parameters are updated as in Eq.(2.4) and Eq.(2.5).

2.2 Partially supervised belief-based mixture modeling

We propose our own partially supervised mixture modeling method that handles imprecise knowledge about the examples. The idea of the method is to set an equivalent of the prior π_k differently for each example x_i $(i \leq M)$ to the value of our *belief*, understood as the certainty about the example belonging to a particular cluster k. The belief is defined as a probability distribution over the clusters in $\mathcal{Y}$, given by a vector b_i, where $b_{ik} = P(Y_i = k)$, satisfying $\sum_{k=1}^{K} b_{ik} = 1$. The belief-based model variant is represented in Fig.2.1 E. The input set to our method is

$X^b = \{(x_1, b_1), ..., (x_M, b_M), x_{M+1}, ..., x_N\}$. Accordingly, the log likelihood for this dataset reads:

$$\begin{aligned} l(\Psi, X^b) &= \sum_{i=1}^{M} \log\big(\sum_{k=1}^{K} b_{ik} f(x_i; \theta_k)\big) \\ &+ \sum_{i=M+1}^{N} \log\big(\sum_{k=1}^{K} \pi_k f(x_i; \theta_k)\big). \end{aligned} \tag{2.11}$$

The maximum likelihood estimate of the parameters Ψ is obtained using the EM algorithm. In the E step of the $(q+1)$-th iteration the posterior probabilities are computed by:

$$t_{ik}^{(q+1)} = \begin{cases} b_{ik} f(x_i; \theta_k^{(q)}) / \sum_{k'=1}^{K} b_{ik'} f(x_i; \theta_{k'}^{(q)}),\ i \leq M \\ \\ \pi_k f(x_i; \theta_k^{(q)}) / \sum_{k'=1}^{K} \pi_{k'} f(x_i; \theta_{k'}^{(q)}),\ i > M. \end{cases} \tag{2.12}$$

In the M step, in contrast to soft-label modeling, the update equation for the mixing proportions does not depend on examples and reads:

$$\pi_k^{(q+1)} = \sum_{i=M+1}^{N} t_{ik}^{(q+1)} / (N - M), \tag{2.13}$$

The Gaussian parameters are updated using the equations Eq.(2.4) and Eq.(2.5).

Key differences to soft-label modeling The two partially supervised belief-based and soft-label methods differ in the way they incorporate imprecise knowledge. Belief values should be interpreted as the actual certainties with which the examples belong to each particular cluster. The plausibilities weight the mixing proportions, giving higher weights to more likely clusters. Consider a model with two components of equal proportions and variances, but different means (as on Fig.2.2 A). A belief value 0.5 for an example indicates that in the data this example lies exactly in the middle between the two means. The plausibility value 0.5 states that there is no certainty about the cluster which the example belongs to, and does not suggest any likely position for the corresponding data point.

The difference in mixing proportion estimation between the belief-based and soft-label modeling (Eq. 2.13 versus Eq. 2.10) has a crucial practical consequence. In the case of soft-label modeling, examples with high plausibilities have higher influence on the estimation than the remaining observations. In the case of belief-based modeling, only the remaining observations are used to estimate the mixing proportions. This implies that the soft-label method is susceptible to bias in the proportion of given examples, whereas belief-based modeling is susceptible to bias in the remaining observations'

proportions. Consider a dataset with two clusters of 1000 elements each (cluster size proportion 1:1, mixing proportions (0.5,0.5)). For very low example numbers it is easy to give biased example proportions affecting the soft-label model estimation. For instance, 10 examples for one and and 100 for the other cluster gives a 1:10 example proportion (and a 99:90 proportion between the remaining observations, close to the desired 1:1). On the other extreme, taking 990 and 900 known examples for the two clusters respectively, hampers the belief-based model estimation in two ways. First, the sample of remaining observations may be too small for proper estimation of the mixing proportions, and in turn, other model parameters in the EM iterations. Second, the remaining observations' proportion 10:1 is biased. Note here that when all examples for a given cluster are known, the belief-based method is not even applicable. To summarize, in comparison to soft-label modeling, belief-based modeling is tailored for the more realistic input sets where the number of examples is small, compared to the amount of unlabeled data required for robust estimation of mixing proportions. However, for high example numbers soft-label modeling should be applied.

Parameter initialization of the supervised methods The semi-supervised and the partially supervised methods take as input examples with cluster labels. Implicitly, they require that the user assumes an order on the clusters to be found in the data. The user labels each example with the number of its believed cluster in the assumed order. On the other hand, the EM algorithm estimates the model components (i.e., clusters) in the order of their initial parameters. Consequently, for the EM algorithm to utilize the examples properly, the initial parameters of each component k should correspond to the cluster labeled k by the user, $k \in \mathcal{Y}$. There are various ways of defining the initial parameters. We describe two of them.

One way is to compute the initial parameters from the examples. For a Gaussian mixture model component k one can compute the mean, variance and proportion of the examples labeled k. Automatically, the initial parameters of component k will correspond to examples from cluster k. However, initialization from examples is not always the best choice, especially when there are only a few of them. Also, for some clusters there might be no example available.

Another common way is to run the same initialization procedures as for unsupervised modeling (section 2.1), returning parameters for clusters in an order not necessarily the same as the one assumed by the user. Next, initial parameters for the EM algorithm are obtained from this clustering. Given any such initialization procedure (in this thesis initialization using quantiles for univariate data), we run the EM algorithm for all possible permutations of initial parameters, and the estimated model with the highest likelihood is returned.

2.3 Mixture model-based clustering

Re-clustering ability In mixture model-based clustering, once the model is estimated, each observation is assigned to its most probable cluster (from equally probable, one is chosen at random). Note that by this maximum a posteriori (MAP) criterion, semi-supervised modeling clusters the examples always in the same way as they are labeled in the input (see section 2.1). In contrast, the partially supervised methods are able to "re-cluster" the examples: an example, although assigned with the highest certainty to a particular cluster k, can have as a result of the EM algorithm the highest posterior probability to belong to a cluster $k' \neq k$. In the case of soft-label modeling, the posterior probability to belong to cluster k can be low for an example x_i if the mixing proportion π_k or the density function $f(x_i; \theta_k)$ are small, even if the plausibility p_{ik} is high (see Eq. 2.9). Belief-based modeling does not take into account the mixing proportions when deciding the cluster label. For a given example, the belief about the example "competes" only with the value of the density function (see Eq. 2.12). In summary, semi-supervised model estimation is most strongly influenced by the examples and, unlike the partially supervised methods, cannot correct for mislabeled examples. Thus, if the data group into clear clusters, the given examples are in ideal proportions and constitute a representative sample from each component, then the semi-supervised method is expected to perform best in estimating the true model. In the more realistic case the knowledge is imprecise and uncertain, and both belief-based and soft-label methods are applicable instead.

Evaluation of clustering accuracy Note, that after assigning to the most probable clusters the clustering is no longer probabilistic but partitional. Thus, when true clustering is available, we evaluate the model-based clustering using standard accuracy (number of correctly labeled observations over the number of all observations) or adjusted Rand index [55]. The latter measure takes values in the $(0, 1)$ interval, and for random clusterings gives values close to 0. High values of the Rand index indicate significant agreement of a given clustering with the correct clustering. Calculating the agreement on any pair of observations, the measure scores: (i) the fact that the observations are clustered together in both clusterings, and (ii) the fact that the observations are not clustered together in both clusterings.

2.4 Extant mixture modeling methods in application to differential expression analysis

High-throughput gene expression measurements provide for a comparison between two experimental conditions. After proper normalization, sets of up- or down-regulated genes (together: differentially expressed) can be determined. Established differential expression analysis tools are based on examining the fold-change of gene expression

level and/or performing a t-test [27, 113, 114]. Typically, a threshold cutting off the differentially expressed genes in the resulting ranked gene list is determined based on the false discovery rate (FDR, [122]). We do not cover the most standard differential analysis approaches as t-test, SAM [122], Cyber-T [7], and LIMMA [115].

Before we describe methods for differential expression analysis, which we either apply or compare to in this thesis, we note application of mixture modeling in related areas. Mixture modeling was widely used in multidimensional clustering of gene expression profiles [137, 90, 43, 31], proving that it is well suited for expression data. In this field of gene expression clustering several approaches extend mixture modeling to include prior knowledge. Costa *et al.* [24, 25] and Pan *et al.* [101] incorporate pairwise constraints known for a subset of the observations and perform penalized mixture modeling ensuring that the constraints are not violated. In a second paper, Wei Pan [99] takes into account a grouping of genes, defined by functional relations on top of the clustering. Alexandridis *et al.* [4] perform semi-supervised model-based clustering and tumor sample classification using tumor samples whose classes are known precisely. None of these methods, however, can easily be adapted to utilize imprecise examples in differential expression analysis. Below we provide details about extant differential expression analysis approaches, which all differ from our partially supervised methodology by the fact that they do not benefit from labeled data.

NorDi The Normal Discretization (NorDi) algorithm, proposed by Martinez *et al.* [87], identifies differential genes by normalizing and discretizing gene expression measures in a given experiment into *under-expressed*, *unexpressed* and *over-expressed* classes. This algorithm first fits the data to a single Gaussian component, iteratively removing outliers, and next calculates the under- and over-expressed thresholds. In each step of the iterative normalization procedure, outliers detected by the Grubbs outliers method [46] are removed from the data, and the Jarque-Bera normality test [61] is performed. The procedure runs until no significant outliers are detected or there is a lower goodness-of-fit with the normal distribution than in the previous iteration. The normality of the obtained distribution is assessed using the Lilliefors normality test [79]. Having the normalized data, and setting a $1 - \alpha$ confidence degree, the thresholds for under- and over-expression cutoffs for data discretization are defined using the lower and upper $\alpha/2$ quantiles. Finally, these cutoffs are used to discretize all values of the initial sample. NorDi is reviewed here and compared to mixture-model based methods in the result sections 2.7 and 2.9 becouse of its distinctive way of modeling the data. Mixture model-based approaches assume that the differentially expressed genes have a different distribution than the unchanged genes. In contrast, NorDi defines differentially expressed genes as those lying on the tails of a distribution common for all genes.

Unsupervised model-based clustering Here we cover the approach proposed by Pan *et al.* [100], which is one of many methods [40, 95, 30] that use unsupervised model-based clustering for the task of detecting genes differentially expressed between

two conditions. First, a two-sample t-statistic for each gene is computed. Next, for a given K, unsupervised mixture modeling of the obtained t-statistics into K components is performed (Eq.2.2–2.5 in section 2.1). Finally, the genes are clustered by their posterior probabilities to the most probable model components. The approach does not *a priori* set the number of model components K. Instead, to determine K, Pan *et al.* apply model selection criteria, namely the Akaike Information Criterion (AIC [1]) and the Bayeasian Information Criterion (BIC [107]):

$$\begin{aligned} AIC &= -2l(\Psi_K, X) + 2|\Psi_K| \\ BIC &= -2l(\Psi_K, X) + 2\log(|X|), \end{aligned}$$

where Ψ_K is the set of model parameters with the number of components fixed to K, X is the input data (here, the computed t-statistics), and $l(\Psi_K, X)$ is the incomplete log likelihood (Eq.2.1). To apply the criteria, first series of model estimations for different component numbers are performed, and next the K resulting in the least AIC or BIC is chosen. By freeing the number of clusters, Pan *et al.* may obtain a model, which better fits the underlying data, but is more difficult to interpret. In our results section we fix the number of clusters so that the results are comparable with our approach.

POE The Probability Of Expression (POE) method consists of a gene expression mixture model together with a Bayesian estimation approach, and is described in detail by Garret and Parmigiani [40]. Here we cover the basics of the mixture model. POE is applied to multiple-experiment data, with the assumption that the expression is different for different subsets of the experiments. Thus, the input data matrix X consists of G rows for the genes and E columns for the experiments. Matrix entry x_{ij} is the intensity of expression measurement of gene $i \in \{1, ..., G\}$ in experiment $j \in \{1, ..., E\}$, or a transformation of this entity, for example log expression ratio with respect to some control. The dataset X is assumed to be normalized and preprocessed. Three latent categories for x_{ij} are defined:

$$\begin{aligned} e_{ij} &= -1 \text{ if gene } i \text{ has abnormally low expression in experiment } j \\ e_{ij} &= 0 \text{ if gene } i \text{ has baseline expression in experiment } j \\ e_{ij} &= 1 \text{ if gene } i \text{ has abnormally high expression in experiment } j \end{aligned}$$

The baseline expression is identified by a large class of experiments with relatively low variability.

For each gene i the uniform distributions are used to model the "abnormal" expression and a normal is used to model the "baseline" expression:

$$\begin{aligned} P(x_{ij}|(e_{ij} = -1)) &\sim \mathcal{U}(-\kappa_i^- + \alpha_j + \mu_i, \alpha_j + \mu_i) \\ P(x_{ij}|(e_{ij} = 0)) &\sim \mathcal{N}(\alpha_j + \mu_i, \sigma_i^2) \\ P(x_{ij}|(e_{ij} = 1)) &\sim \mathcal{U}(\alpha_j + \mu_i, \alpha_j + \mu_i + \kappa_i^+), \end{aligned}$$

where $\alpha_j + \mu_i$ is the center of the baseline expression distribution for gene i in experiment j, with μ_i measuring the gene effect and α_j measuring the sample effect for normal expression levels of gene i in experiment j. The parameters κ_i^- and κ_i^+ denote the lower and upper limits for the abnormal distributions of gene i. A constraint is added that both $\kappa_i^- > r\sigma_i$ and $\kappa_i^+ > r\sigma_i$, for a user-defined r, ensuring that the uniform distributions are able to capture differential expression (in practice, r satisfies $r \geq 5$).

The model parameters are given hierarchical distributions (see Garret and Parmigiani [40] for the distribution functions) and the obtained Bayesian hierarchical model is estimated using a Metropolis-Hastings MCMC approach to obtain posterior distributions of the parameters.

The basic difference to our approach is that POE gains power from estimating the parameters using the entire data matrix over multiple experiments. It proved efficient in our application to large datasets of yeast knockout data (chapter 3). For a lower number of experiments, as the ATM pathway dataset in Human (chapter 4), we apply our partially supervised methods, gaining from known examples instead.

2.5 Partially supervised differential expression analysis

Input data Our approach takes as input data and imprecise examples of differential and unchanged genes. The data are log expression ratios computed for two conditions, referred to as treatment and control, respectively. When replicate experiments are available, log mean ratios or t-statistics should be analyzed. Negative observations refer to lower, while positive observations refer to higher expression values in treatment versus control. The differential genes comprise a small fraction of all genes and their observations are expected to lie on the extremes of the data range.

Analysis There are two analysis scenarios supported: first, clustering into two clusters of differential and of unchanged genes, and second, clustering into three clusters of down- , up-regulated and unchanged genes. Practically, in the first scenario, the differential cluster is defined as the one with the higher variance. In the second scenario, we sort the three estimated model components increasingly by their means. The down- and up-regulated clusters have the lowest and the highest mean, respectively. Our implementation in an R package bgmm (`http://bgmm.molgen.mpg.de`) provides support for fitting a mixture modeling method of choice in both scenarios. As a result, the estimated model parameters, probabilities of belonging to each cluster, and a label of the differential cluster are returned. Additionally, the user can plot the obtained model to verify whether the data clusters as expected. We use the first scenario of two clusters throughout this thesis.

2.6 Validation on synthetic data

In this section we validate the performance of our approach on synthetic data, where the true labels for all observations are known. We compare our partially supervised methodology to other methods in two different aspects: (i) accuracy of model-based clustering and (ii) differential expression analysis.

2.6.1 Evaluation of model-based clustering

First, we evaluate the accuracy of model-based clustering by three methods that utilize labeled data: the partially supervised belief-based and soft-label (section 2.2), as well as semi-supervised modeling (section 2.1).

Input data and examples We consider two different Gaussian mixture models (Model 1 and Model 2), with two components each (Fig.2.2 A, C). In both models the mixing proportions are equal, $\pi = (\pi_1, \pi_2) = (0.5, 0.5)$. The Gaussian model parameters are denoted $\theta = (\mu_1, \mu_2, \sigma_1^2, \sigma_2^2)$. We run three tests on 1000 random samples of 1000 observations each: first, assuming Model 1 and choosing a pool of 14 examples per component, second, Model 2 and 14 examples per component, and third, Model 2 and 450 examples per component. The examples are given belief/plausibility of belonging to their cluster equal to 0.95, and of belonging to the other cluster equal to 0.05. In each test, to generate one sample from the assumed model, we draw the number of observations in the first component from the binomial distribution $N_1 \sim \mathcal{B}(1000, \pi)$, and set the number in the second component to $N_2 = 1000 - N_1$. Next, we draw N_1 observations from the normal distribution $\mathcal{N}(\mu_1, \sigma_1^2)$ and N_2 from $\mathcal{N}(\mu_2, \sigma_2^2)$. For every observation in the sample its *true label* is derived: observations are assigned to the most probable cluster under the assumed model (either Model 1 or 2). Note, that a true label of a given data point is not the true component label, but the true cluster label. It does not necessarily agree with the original model component used to generate the data point. Instead, it agrees with the the cluster to which this point is assigned by the original model. The compared methods make their predictions of the true labels by first estimating the model of the data sample, given the examples, and next model-based clustering of the data. In each test, the accuracy of assigning true labels to observations that are not used as examples is averaged over the 1000 samples.

Advantage of partial supervision The first test (Fig.2.2 A, B) shows advantage of considering imprecise knowledge (discussed theoretically in section 2.3). Model 1 (Fig.2.2 A), with well separated components and sets of examples per component, is easy to estimate. Using all given examples correctly labeled, all methods find true cluster labels accurately (first three bars in Fig.2.2 B). In contrast to semi-supervised modeling, both partially supervised belief-based and soft-label methods

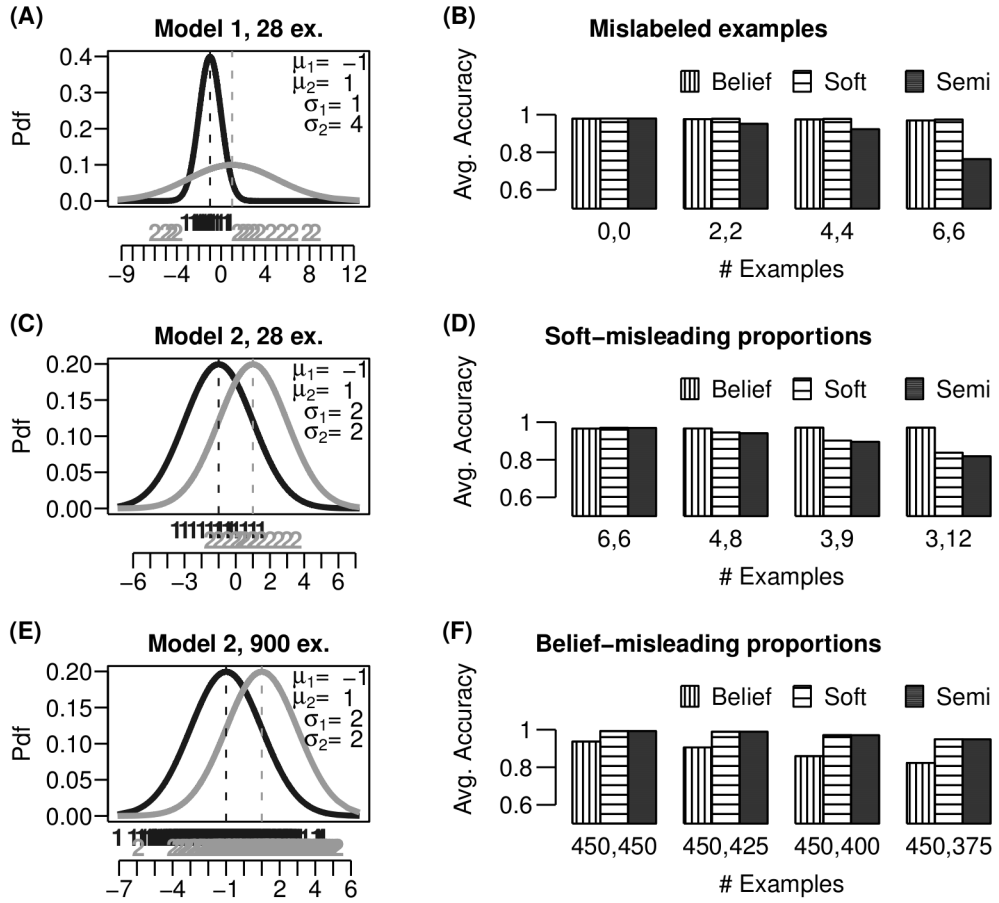


Figure 2.2: Partially supervised model-based clustering of simulated data. (**A**) Model 1 assumed in the first test, with two well separated components (drawn in black and gray), gaussian parameters as indicated on the plot, and separated sets of 14 examples per component (marked below). (**B**) y-axis: average accuracy of belief-based, soft-label and semi-supervised methods in putting data into the same clusters as the true model in **A**. x-axis: different accuracy bar plots for increasing number of examples that are mislabeled (out of the pool of 14 per component). Both partially supervised methods deal significantly better with mislabeled examples than the semi-supervised method. (**C**) Model 2 assumed in the second test, with overlapping components and small example sets (14 per component), plotted as in **A**. (**D**) The plot as in **B**, but the x-axis shows the numbers of examples, correctly labeled, used per component (from those indicated in **C**). The example numbers proportions (from left to right 1:1, 1:2, 1:3 and 1:4) are increasingly biased with respect to the model mixing proportions (1:1). Applied to cluster the data from the model in **C**, belief-based modeling is more resistant to such bias than both soft-label and semi-supervised modeling. (**E**) Model 2 with a large number of 450 examples per component assumed in the third test, ploted as in **C**. (**F**) The plot as in **D**, but here the increasing bias is introduced in the proportions of observations that are not used as examples (from left to right 1:1, 2:3, 1:2, 2:5). Applied to cluster the data from the model in **E** and given large example numbers, belief-based modeling less accurately estimates the model and is less resistant to such bias than both soft-label and semi-supervised modeling.

are highly accurate even when examples are mislabeled by switching their labels to other clusters (remaining bars in Fig.2.2 B).

Belief-based versus soft-label modeling Fig.2.2 C–F shows on Model 2 the differences in performance between the belief-based and soft-label modeling (discussed theoretically in section 2.2). The components of Model 2 largely overlap, and we use overlapping subsets of examples per component. In the second test, for small example numbers (Fig.2.2 C) and equal example proportions the model is well estimated by all methods (first three bars in Fig.2.2 D). However, when the example number proportions disagree with the assumed model mixing proportions, only belief-based modeling achieves high clustering accuracy (remaining bars in Fig.2.2 D). In the third test, with large example numbers (Fig.2.2 E) and equal example proportions, the belief-based method lacks enough observations to estimate the model as good as the soft-label and semi-supervised methods (first three bars in Fig.2.2 F). Additionally, the larger the bias in representation of observations not used as examples, the poorer the accuracy of the belief-based method (remaining bars in Fig.2.2 F). In both cases soft-label modeling behaves similarly to semi-supervised modeling.

2.6.2 Partially supervised differential expression analysis

Next, we show the improvement obtained by using our partially supervised approach in differential expression analysis.

Input data We generated 100 datasets, each simulating expression of 200 differential and 1800 unchanged genes in the control and treatment conditions. Each dataset consists of two data matrices, control and treatment, both with three columns (experimental repeats) and 2000 rows (genes). The basal gene log intensity values in the control matrix are drawn from a normal distribution $\mathcal{N}(10, 1)$. The values in the treatment matrix for the unchanged genes come from the same basal distribution, whereas for the differential genes are drawn from $\mathcal{N}(10, 16)$. This reflects the biological reality where the differentially expressed genes change their expression between the control and treatment condition, but each to a different extent.

Compared methods On these synthetic datasets we compare the partially supervised and semi-supervised modeling with standard differential analysis methods: t-test, SAM [122], Cyber-T [7], and LIMMA [115]. Additionally, we run unsupervised mixture model-based clustering of t-statistic, proposed by Pan *et al.* (section 2.4). The standard differential analysis approaches are applied directly to the simulated control and treatment matrices and return p-values of differential expression. Next, we set the commonly applied p-value thresholds 0.01 and 0.05 to define the differentially expressed genes. The unsupervised clustering is applied to the t-statistic computed using LIMMA. The partially supervised and semi-supervised methods are

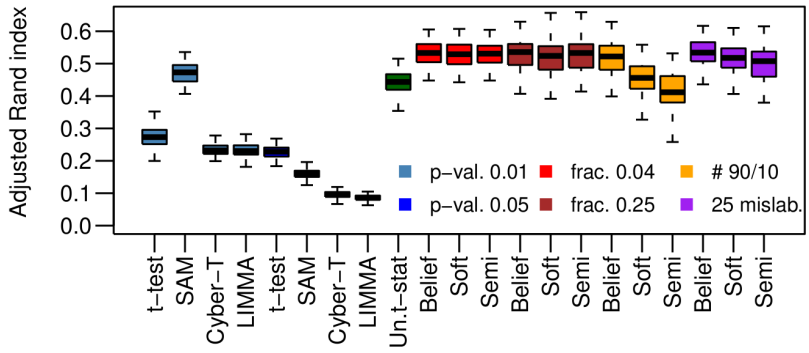


Figure 2.3: Partially supervised differential expression analysis on synthetic data. Given 8 examples of differential and 72 examples of unchanged genes (a 0.04 fraction of all elements in each cluster), the partially supervised belief-based and soft-label methods, as well as semi-supervised modeling achieve superior accuracy (red boxplots) over the standard differential analysis approaches (light blue for the 0.01 p-value cut-off and dark blue for the 0.05 cut-off). Increasing the number of examples used by the supervised methods to 50 and 450 (a 0.25 fraction; brown boxplots) yields similar results. Belief-based method maintains high performance also when the known examples are given in reversed proportion 9:1 (orange boxplots), or are mislabeled (25 examples switched between the 50 differential and 450 unchanged genes, respectively; violet boxplots).

applied to log mean treatment versus control intensity ratios (section 2.5). Application of those methods to the t-statistic yielded the same results and is thus not reported. Examples for the supervised methods are uniformly drawn at random from the set of differential and unchanged genes and assigned belief/plausibility values of belonging to their true clusters equal 0.95.

Accuracy of differential expression analysis We evaluate the compared methods by their accuracy (measured with the adjusted Rand index, section 2.3) of identifying the true differential and unchanged genes. Fig.2.3 shows the adjusted Rand index distributions obtained over the 100 synthetic datasets. Given correct examples in true proportions, the partially supervised and semi-supervised methods most accurately classify the differential and unchanged genes by their simulated expression values. Proportional increase in the number of given examples did not change the results; we show performance with 0.04 (8 for the differential and 72 for the unchanged genes) and 0.25 (50 and 450) of all elements in a cluster used as examples. The unsupervised clustering of the t-statistic performs worse, showing the improvement gained with incorporating knowledge in the analysis. Recall that the model-based methods perform MAP clustering (section 2.3) and do not require setting cut-off thresholds. In contrast, the accuracy of the standard methods depends on p-value cut-off used. For example, the accuracy obtained by SAM with a p-value cut-off 0.01 is the highest among standard approaches, but it drops dramatically for the p-value 0.05. Finally, we show two extreme cases of misleading input example settings that hamper the accuracy of the soft-label, and to a higher extent, the semi-supervised method (section 2.2). First, we give the examples in proportion 9:1, inverted with

respect to the actual proportion of cluster sizes. Second, we again give 50 and 450 examples for the differential and unchanged genes (a 0.25 fraction), but we mislabel 25 of them by switching their labels to the other clusters. The belief-based method proves robust to both misleading input settings.

2.7 Finding Ste12 target genes

Next, we apply the partially supervised approach (section 2.5) to identify pheromone environment-specific target genes of Ste12, a transcription factor in yeast.

Input data We use expression data from four types of cells: untreated wild-type and *Ste12* mutants, as well as wild-type and *Ste12* mutants treated with 50nM of α-factor treatment for 30min [104]. To focus on transcriptional changes triggered by pheromone stimulation, we limit the analysis to 602 genes that show a 1.5 fold up- or down-regulation upon pheromone treatment of wild-type cells. The analyzed data consists of $\log_2$ expression ratios, pheromone-treated *Ste12* mutants versus pheromone-treated wild-type cells. In this dataset, we seek to distinguish the set of *differential* genes from a set of genes that remain *unchanged*.

Input examples We utilize high-throughput experiments to define examples from the set of differential genes: we take 42 genes that have their promoter bound by Ste12 in pheromone environment with a p-value of 0.0001 [49], and that are at least two-fold up-regulated upon pheromone treatment as compared to wild type [104]. We further use the significance of Ste12-DNA binding to reflect the level of certainty about those examples in the belief/plausibility values. The Ste12-DNA binding p-values of the example genes correlate with the logarithm of the changes in expression upon Ste12 knockout in pheromone environment (Pearson correlation coefficient 0.42, p-value 0.0045). We set the belief/plausibility of belonging to the set of differential genes accordingly: the belief values lie in the $(0.5, 0.95)$ interval and are proportional to the log binding p-values. We do not use any examples for the second cluster of unchanged genes.

Compared methods For a comparison to the partially supervised belief-based (section 2.2) and soft-label modeling, we test also the semi-supervised and unsupervised mixture modeling (section 2.1). All these methods are initialized using quantiles (section 2.2) and applied to find two clusters: one for the differential genes, and one for the unchanged. Additionally, we compare to the single-Gaussian NorDi algorithm (section 2.4). To compare to the traditional differential expression analysis, we use the p-values for the genes provided by Roberts *et al.* [104]. Based on the p-values, we define two sets of differential genes, first with the common p-value threshold 0.01, and second with the threshold 0.05. Using each threshold, we first select only genes that

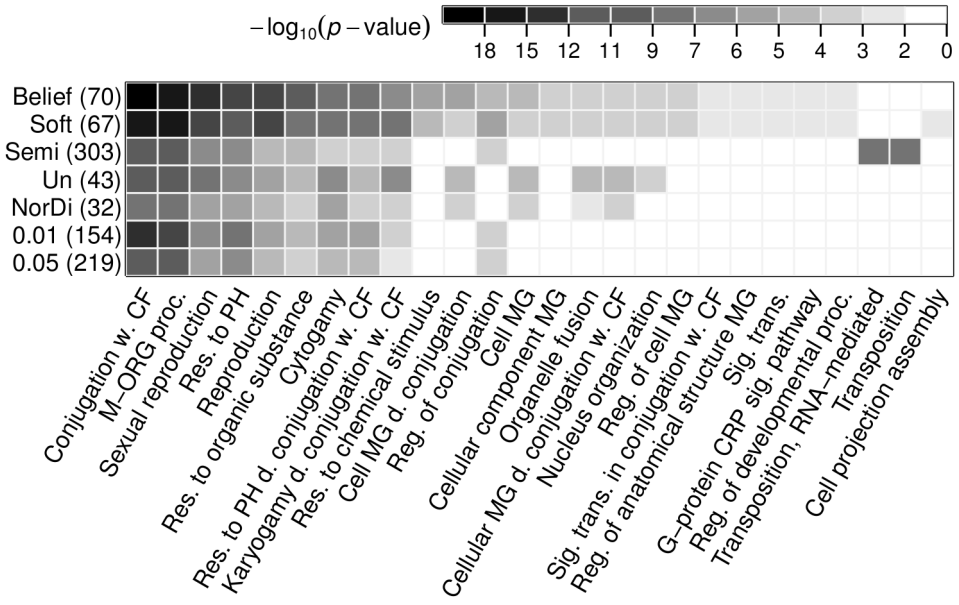


Figure 2.4: Biological validation of identified Ste12 targets. Enrichment p-values (shades of gray) of the sets of Ste12 targets identified by the compared methods (matrix rows; 0.01 and 0.05 denote cut-offs applied to differential expression p-values provided by Roberts *et al.* [104]; set sizes are given in brackets) in GO biological process terms (columns). Each presented term is enriched in at least one Ste12 target gene set with a p-value < 0.01 and FDR< 0.01. Significant enrichment represents distinct behavior of the target genes compared with the rest of all genes. The belief mixture modeling identified a set of Ste12 target genes with the lowest product of all p-values. Abbreviations: Un, unsupervised; CF, cellular fusion; M-ORG, multi-organism, Res., response; PH, pheromone; MG, morphogenesis; Reg.; regulation; CRP, coupled receptor protein; Sig. trans., signal transduction; w. with; d., during.

are differential under pheromone treatment in wild-type cells. Next, from those we select genes that are differential under Ste12 knockout in pheromone-treated cells.

Accuracy of identifying Ste12 targets We define the set of *Ste12 targets* identified by each method as those genes from the obtained set of differential genes, which are down-regulated in the *Ste12* mutants (Ste12 is a transcriptional activator [69]). We evaluate the identified sets of Ste12 targets by testing whether the proteins encoded by the targets take part in Ste12-dependent processes induced by pheromone (Fig.2.4). To this end, for each target set we computed the p-values for its enrichment in Gene Ontology annotations (GO [5]), using the TermFinder tool by Boyle *et al.* [17].

The set of Ste12 targets identified by the belief-based modeling method has the highest enrichment in the GO annotations related to Ste12 activity upon pheromone stimulation [50]: mating and conjugation with cellular fusion. Similarly strong evidence for the same functionality is shown for the set of Ste12 targets of comparable size, identified by the soft label modeling method. Unsupervised mixture modeling and the NorDi algorithm identify Ste12 target sets that are smaller than the sets iden-

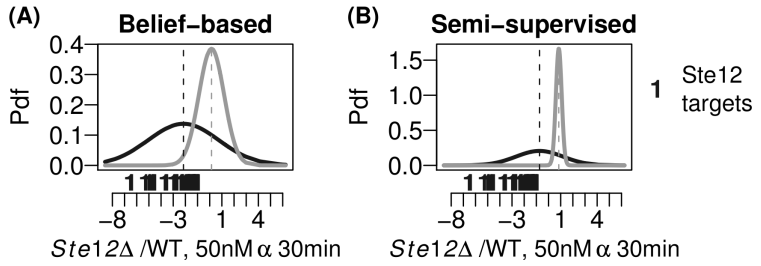


Figure 2.5: Different impact of examples on the models estimated by different supervised methods. Model estimated by the partially supervised belief-based (**A**), and by the semi-supervised mixture modeling (**B**). The plots as in Fig.2.2 A.

tified by the two partially supervised methods, leaving out many genes that are functionally related to the pheromone-triggered and Ste12-dependent processes (Fig.2.4). Semi-supervised modeling, in contrast, includes all given examples in the cluster of differential genes. As opposed to belief-based modeling, the semi-supervised method shifts this cluster towards low change in expression upon Ste12 knockout (Fig.2.5). Therefore, its set of identified Ste12 targets contains half of all analyzed genes, and incorporates most superfluous genes, e.g. genes taking part in the transposition process. Also relatively big, the sets of Ste12 targets identified using the two p-value cut-offs have better enrichment scores than the set identified by semi-supervised modeling, but worse than the sets identified by the partially supervised methods (Fig.2.4).

2.8 Distinguishing miR-1 from miR-124 targets

To further evaluate the partially supervised mixture modeling methods, we check their accuracy of distinguishing miR-1 from miR-124 target genes in human, based on two expression datasets from transfections of these microRNAs (shortly, miRNAs [81]) and knowledge from computational miRNA target predictions.

Input data and examples We use the subset of the genes measured by Lim *et al.* [81], which can be divided into two distinct clusters with rigorous experimental verification: 90 miR-1 targets [108, 141], and 35 miR-124 targets [127, 75, 63]. Among them, we use as examples 16 miR-1 and 11 miR-124 target genes that have computationally predicted binding sites of miR-1 and miR-124, respectively. We take only the examples that are predicted as respective targets by both computational methods that we used: MirTarget2 [126, 125] and miRanda [13]. The belief/plausibility values for examples to belong to their clusters are set to 0.95.

Accuracy of distinguishing miR-1 from miR-124 target genes In both transfection datasets, we expect to see down-regulation of one miRNA's target genes (e.g., miR-1 targets upon miR-1 transfection) and the other target genes unchanged by the

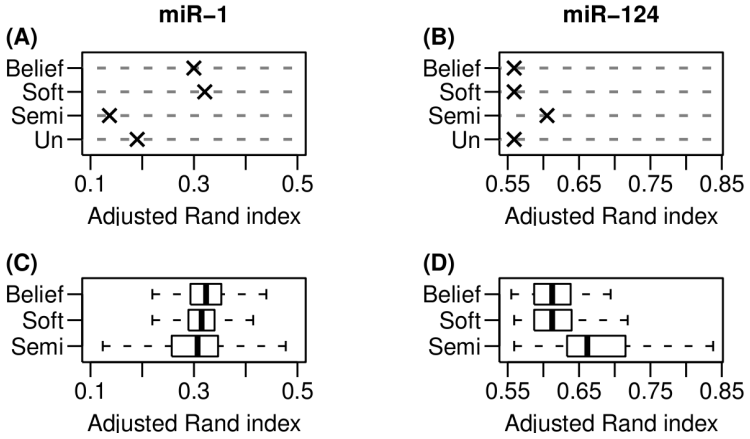


Figure 2.6: Accuracy of distinguishing miR-1 from miR-124 targets. (**A**) The adjusted Rand index (x-axis) indicates whether the different mixture modeling methods (y-axis) clustered the data correctly into true groups of known miR-1 and miR-124 targets. Analyzed expression data come from the miR-1 transfection experiment. The semi- and partially supervised methods utilized 16 computationally predicted examples of miR-1, and 11 of miR-124 targets. (**B**) Plot as in **A**, but for the data obtained under the miR-124 transfection. (**C**) Box-plots show the adjusted Rand index distribution (x-axis), obtained by the methods (y-axis) in 1000 tests, where 16 examples were drawn from all miR-1 targets, and 11 drawn from all miR-124 targets at random, and the data came from miR-1 transfection. (**D**) Plot as in **C**, but for the data from miR-124 transfection.

transfection. Therefore, for each dataset we apply the partially supervised modeling methods and, for comparison, the remaining mixture modeling methods to find two clusters. The obtained clusterings are validated with the two true clusters of miR-1 and miR-124 target genes using the adjusted Rand index (section 2.3). The examples are not included in computing the index.

The measurements from the miR-124 transfection are easier to cluster than the measurements from the miR-1 transfection (for miR-124 the clusters are more separated; data not shown). Accordingly, the estimations of the model are less accurate for the miR-1 transfection data (Fig.2.6 A versus B). As expected (section 2.3), in the easier case of miR-124 transfection, the semi-supervised modeling achieves better results than others. On the contrary, in the more difficult case of the miR-1 transfection, the semi-supervised method performs worst, and the partially supervised methods achieve the highest accuracy. The same is observed when randomly chosen sets of examples are used instead of the computationally predicted ones (Fig.2.6 C, D).

2.9 Clustering cell cycle gene profiles

Finally, we make use of partially supervised mixture modeling in the task of clustering cell cycle gene expression profiles [20].

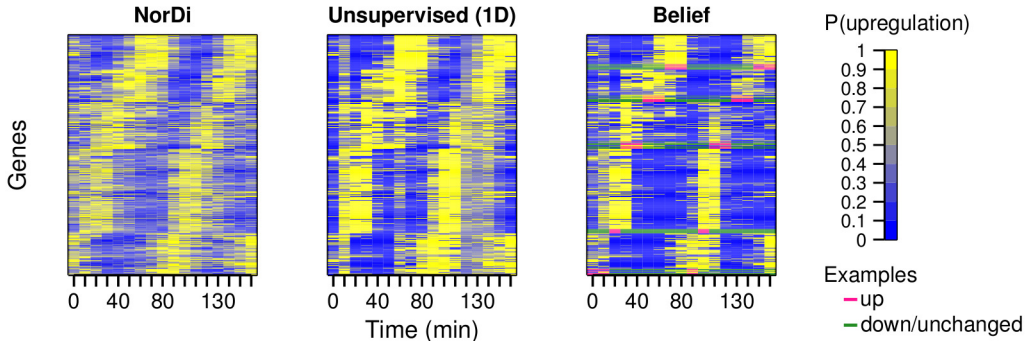


Figure 2.7: Cell cycle gene clustering. The probability of up-regulation estimated for each cell cycle gene (rows; ordered by their true cluster labels), in each time-point (columns) by three methods: NorDi, as well as unsupervised and belief-based mixture modeling, applied to each time point data separately. Belief-based mixture modeling, which uses examples of up-regulated and of unchanged genes in each time-point (marked in pink and green), achieves most clearly visible distinct gene expression profiles, characteristic for five cell cycle phase clusters.

Input data and examples Based on expression measurements over 17 time points, which cover two cell cycles, 384 genes fall into five disjoint clusters. Each cluster contains genes peaking at a particular cell cycle phase: early G_1, late G_1, S, G_2, or M [20]. Following Yeung *et al.* [137] we take this five-phase criterion as the true clustering of genes in this dataset. For each phase cluster we take seven examples of genes known to be active in this phase (first seven listed for that cluster in *et al.* [20] Tab.1, excluding genes active in more than one phase), together 35 examples.

Clustering procedure The partially supervised, unsupervised, semi-supervised mixture modeling, as well as NorDi are applied to cluster the 384 genes in a two-step procedure:

1. *Clustering of data from each time point into two clusters.* In the data from each time point t separately, find two clusters, one of which corresponds to the up-regulated genes. Use seven genes known to be active in the phase corresponding to this time point as examples for the up-regulated cluster, with belief/plausibility values 0.95. Similarly, use the remaining 28 examples for the second cluster of genes that are unchanged or down-regulated. Output the probability p_g^t of each gene g to belong to the cluster of up-regulated genes (the posterior probability for the mixture modeling methods, and one minus the p-value of differential expression for the NorDi algorithm).
2. *Clustering of genes into five clusters.* For each cell cycle phase cluster construct a binary profile reflecting the known default "activity" of genes from this cluster over the 17 time points. The activity profile $\vec{v}_c$ of a phase cluster c has a value 1 in entry t if genes from this cluster peak in the time point t. Otherwise the entries are 0. For each gene g, take the vector of its estimated up-regulation probabilities

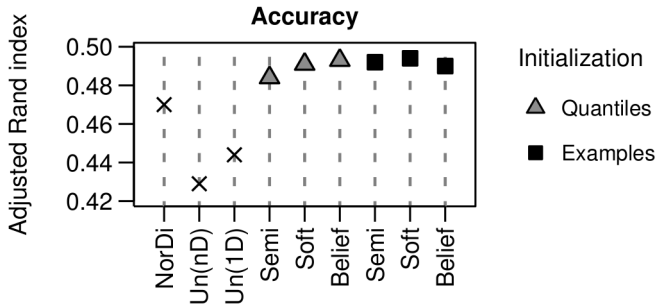


Figure 2.8: The accuracy of cell cycle gene clustering. From all compared methods, the partially supervised have higher accuracy (measured by adjusted Rand index, y-axis) in grouping genes into five cell-cycle gene clusters, than the semi-supervised and unsupervised methods. The partially supervised modeling methods were initialized in two ways: either quantile or example-based (section 2.2).

$\vec{p_g} = (p_g^1, ..., p_g^{17})$ from step 1, and assign g to the cluster with the most similar activity profile. Formally, we assign gene g to cluster

$$c^* = \arg\max_c \left(\vec{v_c}^T \vec{p_g} + (\mathbf{1} - \vec{v_c})^T (\mathbf{1} - \vec{p_g})\right),$$

where $\mathbf{1}$ denotes a vector of length 17 filled with 1s.

Advantage of supervised methods Fig.2.7 compares the clusterings obtained in the first step by the unsupervised algorithms, to the clusterings obtained by our belief-based modeling. The examples used by our method help to clearly distinguish patterns of genes from each phase cycle peaking at their characteristic time points.

Fig.2.8 shows that all supervised modeling methods, regardless of the parameter initialization, outperform the unsupervised methods in clustering cell cycle gene profiles using the two-step procedure. For comparison, we applied also a one step analysis with multidimensional Gaussian mixture modeling ([137], denoted $Un(nD)$) to separate the entire dataset at once into five clusters. Interestingly, multidimensional clustering obtained the least accuracy, measured with the adjusted Rand index (section 2.3). Best results are obtained for the two-step procedure, using either belief-based or soft-label modeling in the first step.

2.10 Discussion

Mixture modeling is an established technique in machine learning, and proved successful in the field of gene expression analysis. The two partially supervised methods presented in this paper extend mixture model-based clustering, adding the ability to utilize imprecise examples. In contrast to other mixture modeling methods that

incorporate knowledge, both belief-based and soft-label modeling can be customized for differential expression analysis guided by examples of genes that are believed to be up, down or unchanged. The known examples usually constitute only a small subset of all genes and are themselves not 100% certain. The presented applications show a rich variety of possible knowledge sources for examples: high-throughput TF-DNA binding experiments, computational predictions of miRNA targets, and literature knowledge of genes active in different cell cycle phases. The known examples are traditionally used to verify experimental outcome *after* it is defined by differential expression analysis. Our partially supervised methodology incorporates such prior biological knowledge *into* the analysis itself, making the outcome more reliable.

The methodology enables confronting available uncertain knowledge with the data. On the one hand, the partially supervised methods profit from the examples to better cluster the remaining data. On the other hand, they use the entire data to verify the knowledge about the examples. For instance, the signal in the data may contradict the prior belief about a gene to be up-regulated in a knockout experiment. Both partially supervised methods may "re-cluster" the examples with such improbable initial cluster labels. In this way, they are more flexible than semi-supervised mixture modeling, which assumes that the example labels are fixed.

The application of the proposed methodology to the problem of differential analysis imposes two natural restrictions, which could easily be abandoned for the needs of different applications. First, here we analyze only one-dimensional data, but in general the approach can as well be extended to multidimensional clustering given examples with imprecise cluster labels. Similarly, we restrict ourselves only to consider two- or three-component models, although it is common to use tools of model selection to choose out of models with arbitrary numbers of clusters. Here it is also dictated by the nature of the problem: we assume the clusters to be interpreted, and the known examples to be assigned to each of the clusters. Intuitively, we expect examples of differential or unchanged genes (two clusters), alternatively, of up-, down-regulated, or unchanged genes (three clusters). It would be difficult to assign those examples to clusters in a model with more than three components.

Chapter 3

Elucidating Gene Regulation With Informative Experiments

This chapter puts forward a framework for elucidating gene regulation downstream of a given signaling pathway using an optimal set of experiments. The framework, introduced in sections 3.1–3.7, benefits from prior knowledge about the pathway formalized in a logical model (section 3.2). The model predictions are utilized by our experimental design (ED) algorithm called MEED (sections 3.3–3.6). Section 3.7 describes how the measurements in the experiments proposed by MEED are matched with the model predictions in order to elucidate the regulator-target relations downstream of the modeled pathway. In section 3.8.2 we cover alternative ED approaches, which we compare to our MEED framework on synthetic data (section 3.9) and in application to a signaling pathway in yeast (section 3.10).

3.1 The MEED framework

The proposed framework consists of three components: modeling of the studied signaling pathway, an experimental design algorithm MEED and an expansion procedure. The framework aims to discover regulatory relations and mechanisms of transcriptional control downstream of the given signaling pathway using an optimal set of perturbation experiments (see sections 1.1 and 1.2 for biological semantics of these notions). Software implementing our framework is freely available from `http://meed.molgen.mpg.de/`.

The first component, the model, formalizes prior knowledge about signaling relations between the molecules in the given pathway. The model is predictive: For a given experiment (i.e., extracellular stimulation and genetic perturbation), the model predicts the activation states of the regulators in the pathway. Here, we assume that both the signaling and regulatory relations are discrete logical functions and that the model describes the steady state of the system after exposure to the experiment. In addition, we predefine a repertoire of logical functions that formalize regulation mechanisms, such as activation or repression (regulation by only a single pathway

component is considered). By applying all predefined logical functions to the model-predicted state of a given regulator under a given experiment, we obtain predictions about all possible readouts of the regulator's target genes. This is done for all regulators and all candidate experiments. In this way, we calculate predicted expression profiles for all potential targets of the regulators. The modeling formalism applied here was introduced by Gat-Viks *et al.* [42]. The expansion procedure utilizing a given signaling pathway model for predicting expression profiles was proposed by Gat-Viks and Shamir [41].

Our contribution consists of proposing a model expansion experimental design (MEED) algorithm, and embedding it in the general framework. The MEED algorithm aims to select from the set of candidate experiments optimizing two objectives: (i) to minimize the number of selected experiments and (ii) to maximize diversity between the predicted expression profiles. The second condition aims to avoid an ambiguous situation in which two genes with distinct regulatory mechanisms attain the same expression profile under the suggested experiments. Only in the case in which the two genes have two distinct expression profiles, it is possible to distinguish their regulatory mechanisms. Next, the chosen experiments should be carried out in a lab and used to identify regulator–target relations. To this end, the expansion procedure matches the model-predicted expression profiles of putative targets for the set of experiments selected by MEED with real expression measurements observed under the same experiments. The building blocks of our framework are illustrated in Fig.3.1 and described in sections 3.1–3.7.

3.2 Predictive logical model of Gat-Viks et al.

First, we formalize the available qualitative information about a given pathway in a logical model (Fig.3.1 A, left) with discrete variables, proposed by Gat-Viks *et al.* [42]. A model $\mathcal{M}$ consists of a set V of variables, a set $U = \{u_1, ..., u_k\}$ of discrete states that the variables may attain, and a set of discrete logical *regulation functions* $f_v : U^{|Pa(v)|} \rightarrow U$, for each $v \in V$. f_v defines the state of variable v as a function of the states of a subset $Pa(v)$ of all variables, referred to as the set of *parents of* v. A graphical representation of the model is a digraph $G = (V, A)$, where each node $v \in V$ is connected by incoming edges in A with the nodes in $Pa(v)$, i.e., $(w, v) \in A$ iff $w \in Pa(v)$. The set of *stimulators* I includes all variables with zero indegree.

Biological semantics of the model The model formalizes signaling relations in a given pathway. The set of model stimulators I represents the environmental signals, which trigger the pathway. Remaining model variables correspond to the signaling molecules. In our analysis (sections 3.9–3.10.4), we assume that all variables may have three states: 1 (*activated*), -1 (*deactivated*) or 0 (*unchanged*).

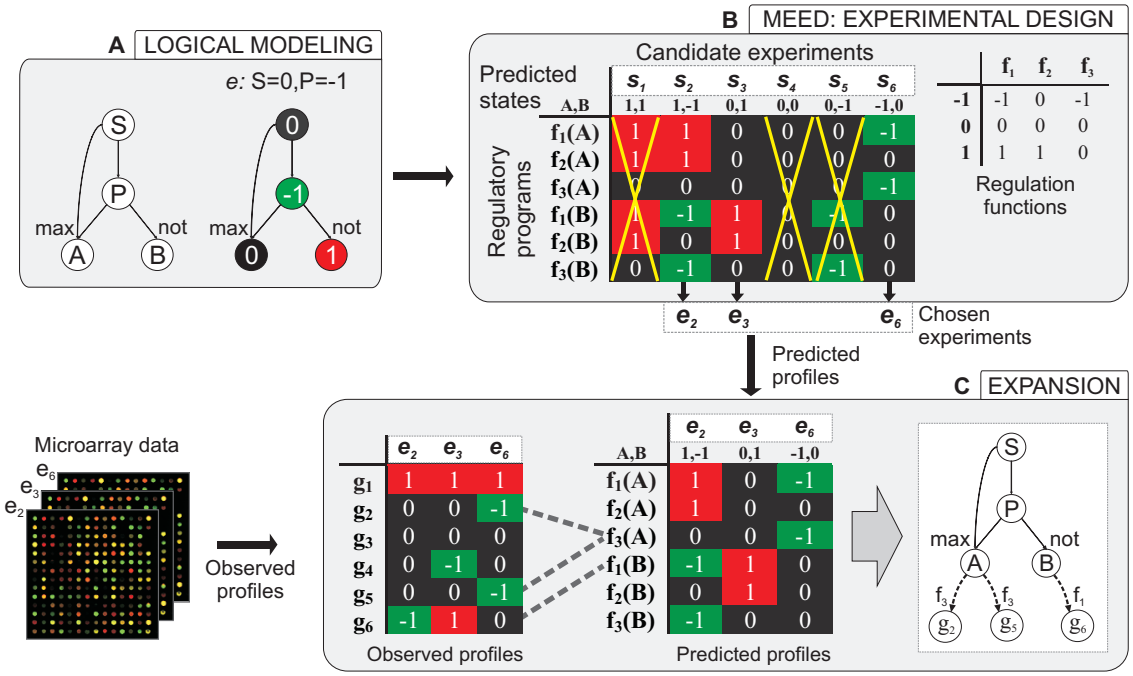


Figure 3.1: (**A**) Logical modeling. Left: a toy model. S – a stimulator variable representing the environmental signal, P – a variable representing a signaling molecule. A, B – regulators representing transcription factors. max, not: regulation functions. Right: prediction of regulator states. e – experiment, in which environmental signal is medium (stimulation $S = 0$) and the signaling molecule is knocked out (perturbed variable P, perturbation state -1). (**B**) Our MEED algorithm. Right: three exemplary regulation functions, f_1, f_2 and f_3, represented by a truth table, in which the first column contains the states of a regulator, and each other column i contains the predicted responses of a target gene controlled by the regulator using f_i. For example, f_1 determines that activated (state $= 1$) regulator up-regulates (state $= 1$) its target, and deactivated (state $= -1$) regulator down-regulates (state $= -1$) its target. Left: matrix of predicted responses. Rows – regulatory programs, each represents a chosen regulator (A or B) acting on a target gene through a chosen regulation function. Columns – predicted model states s_1–s_6 in the set of candidate experiments given as input to MEED. The predicted states of regulators A and B appear below. For example, the third column corresponds to a model state s_3 predicted for experiment e_3, with predicted states $A = 0$ and $B = 1$. A matrix entry – a predicted response of a potential target gene assuming it is regulated by its row's regulatory program in its column's experiment. Hence, each row of the matrix is a predicted profile for a given regulatory program. If the predicted profiles are different, they are referred to as distinguished. MEED aims to find the smallest subset of candidate experiments, which distinguishes between the same pairs of regulatory programs as the full set of candidate experiments. Here, MEED chooses three out of the candidate experiments: e_2, e_3, and e_6, which distinguish all regulatory programs (the remaining ones are marked as deleted). (**C**) The expansion procedure. The experiments proposed by MEED are carried out and the measurements are used in the expansion procedure. Left: the measurements of gene expression in the chosen experiments are referred to as observed profiles of the genes (rows). Middle: a matrix as in B, including only experiments chosen by MEED. The expansion procedure identifies regulatory programs for the genes by matching of predicted and observed profiles (marked as dashed gray lines). Right: genes matching identical regulatory programs constitute regulatory modules. Here, two regulatory modules are found: the regulatory program $f_3(A)$ controls the module of g_2 and g_5, and regulatory program $f_1(B)$ controls g_6.

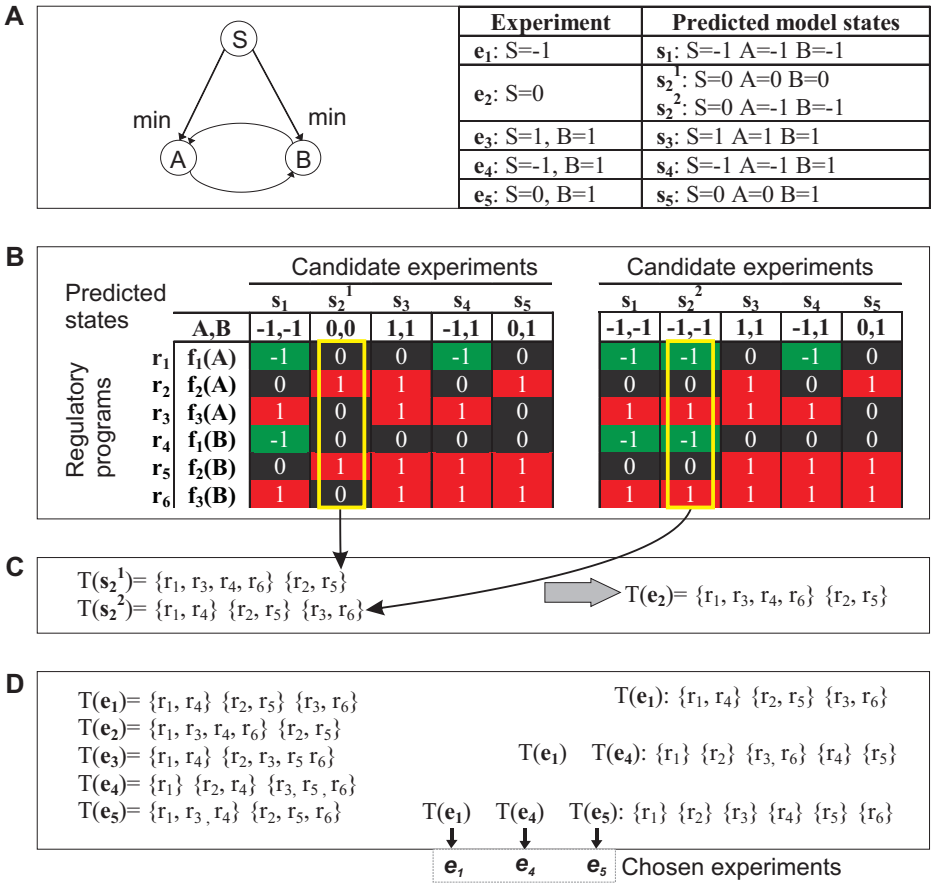


Figure 3.2: (**A**) Left: A toy logical model with cycles. Right: Predicted model states for exemplary experiments. There are two model states, s_2^1 and s_2^2, predicted for experiment e_2. (**B**) Matrices of predicted responses, as in Fig.3.1 B. Each matrix corresponds to one model state predicted for e_2 (in the left matrix, the second column corresponds to s_2^1, in the right matrix to s_2^2). The regulators, their predicted states as well as the candidate experiments are taken from **A**. Regulation functions are the same as in Fig.3.1 B. Note that in this example, each regulatory program has two possible predicted profiles, one for each matrix. (**C**) Each predicted model state induces a partition on the set of regulatory programs according to their predicted response. Left: the partitions $T(s_2^1)$ and $T(s_2^2)$. $T(s_2^1)$ corresponds to a column marked in yellow in the left matrix in **B**. It divides regulatory programs into two blocks $\{r_1, r_3, r_4, r_6\}$, and $\{r_2, r_5\}$, which have predicted response 0 and 1, respectively. Therefore, r_1 and r_4 are not distinguished by s_2^1, as opposed to r_2 and r_6. The partition for an experiment is a supremum over partitions for its predicted model states. Here, $T(e_2)$ is given on the right. (**D**) The partition for a set of experiments is the intersection of the partitions for each of the experiments. Left: partitions for candidate experiments e_1–e_6. The partition $T(e_x)$ for each experiment except e_2 ($x \in \{1, 3, 4, 5\}$) equals the partition for its predicted model state $T(s_x)$. Right: partitions obtained in each step of MEED. In the first step, MEED considers partitions for all candidate experiments and chooses the one that has the highest entropy score (in this example, e_1). In the second step, it extends the list of chosen experiments by adding an experiment that provides the highest entropy gain (e_4). The joint ability of experiments e_1 and e_4 to distinguish between regulatory programs is represented by an intersection of their partitions. Adding e_5 further partitions the block $\{r_3, r_6\}$ into single-variable blocks. The partition for the chosen experiments e_1, e_4, and e_5 has only single-variable blocks. Therefore, in this example it is possible to distinguish all regulatory programs using the chosen set of experiments.

Representation of experiments in the model An *experiment* on the model $\mathcal{M}$ is formalized by defining: (i) *stimulation* – an assignment of states to all model stimulators in I (fixed according to the levels of environmental signals applied in the experiment); (ii) *perturbed variables* – a set $P \subseteq V \setminus I$ of model variables that are subject to perturbation; and (iii) *perturbation states* – fixed states of the perturbed variables, which represent the type of experimental manipulation, such as knockout (perturbation state is -1) or over-expression (perturbation state is 1). In this thesis, we consider only experiments, in which either none or exactly one variable is perturbed. Assuming in general that k perturbation states are possible for each variable, having $|I|$ stimulators and $|P|$ variables that can be perturbed in the model $\mathcal{M}$, the number of all possible experiments on $\mathcal{M}$ is $k^{|I|}(|P|k+1)$. There are $k^{|I|}$ possible stimulation states, $|P|k$ ways of perturbing one variable, and one where no variable is perturbed.

Model states A model state s (illustrated in Fig.3.1 A right), is an assignment of states to each of the variables in the model, $s : V \to U$. We say that s *agrees* with the model $\mathcal{M}$ on variable v if the state $s(v)$ of variable v equals the output of its regulation function when applied to its parents' states, $f_v(s(Pa(v))) = s(v)$. For a given experiment e, we call a model state s_e *predicted for experiment* e if (i) the stimulators in I are assigned their stimulation and the perturbed variables in P are assigned their perturbation state as defined by e, and (ii) the state $s_e(v)$ of each variable $v \in U \setminus I \setminus P$ agrees with $\mathcal{M}$ on v. The state $s_e(v)$, assigned to each variable $v \in V$ by a given model state s predicted for e, is called an *e-predicted state* of the variable v. A predicted model state for a given experiment is a prediction about a steady state of the biological system, describing the states of the pathway components under this experiment.

In the case when the model $\mathcal{M}$ is acyclic, each experiment e on $\mathcal{M}$ has exactly one possible model state s_e predicted for e, defining a unique e-predicted state for each variable. In this case, the unique model state s_e can easily be computed (claim 1 in Gat-Viks *et al.* [42]). s_e is calculated iterating over the variables $v \in V \setminus I \setminus P$ in a topological order defined by the acyclic model graph, and obtaining the state of each v by applying f_v to its parents' states. The first variables will have their parents' states fixed according to stimulation and perturbation states defined by the experiment e. The running time of this procedure is linear in the number of nodes plus the number of edges: $O(|V| + |A|)$, which is the time required to compute the topological order [23].

However, for a model whose graph contains feedback loops, it is possible to obtain zero, one or several possible predicted model states (see Fig.3.2 A for an example). In this case, in order to compute the predicted model states, the cyclic model $\mathcal{M}$ is first transformed into an acyclic model $\mathcal{M}_F$ using its *feedback set*. A feedback set in a directed graph is a set of nodes whose removal renders the graph acyclic [35]. To obtain $\mathcal{M}_F$, given a feedback set F in $\mathcal{M}$, the regulation functions of the variables in F are changed to null, and the edges incoming to F in the model graph are removed

accordingly. Note that for a given experiment e, if a state s_e of the new model $\mathcal{M}_F$ predicted for e agrees with $\mathcal{M}$ on every variable $v \in F$, s_e is also a predicted state of the model $\mathcal{M}$. Given a state of the model $\mathcal{M}_F$, it is easy to check the agreement by calculating f_v for each $v \in F$. Following Gat-Viks *et al.* [42], we provide the procedure for computing the states of $\mathcal{M}$ predicted for a given experiment e, using its feedback set F and a topological ordering on the model graph.
Generate each possible state assignment to F. For each assignment $s_F : F \to S$:

- Generate an experiment e' for $\mathcal{M}_F$ by joining the stimulation in e with s_F.
- Use a topological ordering to compute a (unique) model state s_e predicted for e'.
- If s_e agrees with $\mathcal{M}$ on every $v \in F$, add it to the set of states of $\mathcal{M}$ predicted for e.

The procedure runs in $O(k^{|F|}(|V| + |A|))$ time, since it requires checking $k^{|F|}$ state assignments to F, and for each assignment computing a unique model state as described before. Thus, the restrictive element for the efficiency of this procedure is the size of the feedback set. Computing a minimal feedback set is NP hard [64], and Gat-Viks *et al.* [42] use known heuristics [109] for this task.

Discrepancy between the predicted and observed model states For a given experiment e on model $\mathcal{M}$, Gat-Viks *et al.* [42] define an *observed partial state* e_S as a set of measurements for some of the model variables during the experiment, where $e_S(v) = \text{null}$ for those variables v that were not observed. The model $\mathcal{M}$ can be used to compare the possible model states predicted for e with the observed partial state. In the case when more than one predicted model state exists, we expect the correct one to be the most similar to the observed partial state. To assess this similarity, a *discrepancy* $D(s_e, e)$ between the experiment e and a given model state s_e predicted for e is measured: $D(s_e, e) = \sum_{v \in V, e_S(v) \neq \text{null}} (s_e(v) - e_S(v))^2$. The state with the smallest discrepancy is considered the unique model state predicted for the experiment e, with the assumption that this state corresponds to the steady state reached by the biological system when e is performed.

3.3 Regulatory programs and their predicted profiles

Discrepancy compares predicted and observed states of the variables in the model. Here, we introduce a tool to connect predicted model states with observed states of variables that represent target genes and are not included in the model, but are potentially transcriptionally regulated by the modeled pathway components. Below we formalize the particular mechanisms of this regulation.

Regulators and a repertoire of regulation functions First, we define a set V' of *regulators*, which is the subset of all model variables, $V' \subseteq V$ and represents proteins

having transcriptional control over response of target genes. Next, we predefine a set of regulation functions $\mathcal{F}$ that describe biologically relevant logical relationships between a subset of regulators and its target. Here, we consider transcriptional control by only a single regulator, which can attain one of three possible states $u \in \{-1, 0, 1\}$. Thus, we adapt six biologically relevant functions introduced by Yeang and Jaakkola [136] to define a repertoire $\mathcal{F} = \{a_N, a_S, a_B, i_N, i_S, i_B\}$ of six one-argument regulation functions with the following formulas and biological semantics:

- *Necessary activation*

$$a_N(u) = \begin{cases} -1, \text{ iff } u = -1 \\ \\ 0 \text{ otherwise.} \end{cases} \tag{3.1}$$

  Deactivation of the regulator forces down-regulation of the target. The target remains unchanged upon the activation of the regulator.

- *Sufficient activation*

$$a_S(u) = \begin{cases} 1, \text{ iff } u = 1 \\ 0 \text{ otherwise.} \end{cases} \tag{3.2}$$

  Activation of the regulator forces up-regulation of the target. The target remains unchanged upon the deactivation of the regulator.

- *Activation both*

$$a_B(u) = u \tag{3.3}$$

  Both the deactivation of the regulator forces down-regulation of the target, and the activation of the regulator forces up-regulation of the target.

- *Necessary inhibition*

$$i_N(u) = \begin{cases} 1, \text{ iff } u = -1 \\ \\ 0 \text{ otherwise.} \end{cases} \tag{3.4}$$

  Deactivation of the regulator forces up-regulation of the target. The target remains unchanged upon the activation of the regulator.

- *Sufficient inhibition*

$$i_S(u) = \begin{cases} -1, \text{ iff } u = 1 \\ 0 \text{ otherwise.} \end{cases} \tag{3.5}$$

  Activation of the regulator forces up-regulation of the target. The target remains unchanged upon the deactivation of the regulator.

- *Inhibition both*

$$i_B(u) = -u \tag{3.6}$$

  Both the deactivation of the regulator forces up-regulation of the target, and the activation of the regulator forces down-regulation of the target.

Regulatory programs A *regulatory program* $r = (v, f)$ consists of a regulator $v \in V'$ from the model and a regulation function chosen from the repertoire $\mathcal{F}$. The regulator tells "who" regulates, whereas the regulation function formalizes the

regulatory mechanism and tells "how" (e.g., Fig.3.1 B). For a given model state s_e predicted for an experiment e, the value $f(s_e(v))$ defines an (e, r)-*predicted response* of a potential target gene to the regulatory program r in experiment e. Biologically, the predicted response specifies whether the target gene is in state 1 (*up-regulated*), -1 (*down-regulated*) or 0 (*unchanged*) in the experiment e. Finally, a vector of predicted responses in a given ordered set of experiments E defines an (E, r)-*predicted profile*. Assuming the model is correct, the predicted profile reflects the transcriptional response of a potential target gene controlled by the program r in the given set of experiments E. Given the set of regulators V' and the repertoire of regulation functions $\mathcal{F}$, the set of predicted profiles R contains $V' \times \mathcal{F}$ elements. Fig.3.1 B left presents the set of predicted profiles as rows of a matrix, referred to as a *matrix of predicted responses*. The columns of this matrix correspond to predicted model states for a set of experiments, and entries to predicted responses.

In a general case, one experiment may define several predicted model states, giving several predicted states per regulator. Given that there are at most m model states predicted for each experiment in E, there are pessimistically $m^{|E|}$ different predicted profiles for each regulatory program. These predicted profiles are determined by the different combinations of model states predicted for experiments in E. Fig.3.2 B illustrates two predicted profile sets as two matrices of predicted responses, one per each of the two model states predicted for experiment e_2 (the remaining experiments induce unique model states).

3.3.1 Requirements for the experiments

For a given set of experiments E and a set of regulatory programs R, the expansion procedure (section 3.7) matches the (E, r)-predicted profiles, for all $r \in R$, with the observed profiles of the measured genes. For the matching to proceed unambiguously, two requirements need to be satisfied:

1. *A single predicted profile per regulatory program* Before designing and carrying out experiments, we cannot anticipate which combination of their predicted model states will fit best the steady states reached in the biological system. In our framework, this problem is overcome by the MEED algorithm by taking into account all possible sets of predicted profiles (section 3.3.2). However, the expansion procedure requires a single set of predicted profiles. In the input to the expansion procedure the measurements in the designed experiments are already available. The required single profile set can be obtained using the experimental measurements to choose the predicted model states with the least discrepancy with the observed states.
2. *All pairs of regulatory programs, in all combinations of predicted model states, should have different predicted profiles.* We deal with this requirement using the MEED algorithm (section 3.5). To this end, below (section 3.3.2) we introduce auxiliary notions of *distinguishability* between two regulatory programs.

3.3.2 Distinguishing regulatory programs

We start by defining how a given pair of regulatory programs is distinguished by a predicted model state, next we extend the definition for a single experiment (which might induce several predicted model states), and finally, we generalize by stating how two programs are distinguished by a set of experiments.

Distinguishing by a predicted model state Recall that a model state s_e predicted for a given experiment e assigns to each regulator its e-predicted state (section 3.2). From this state we can compute the (e, r)-predicted response for each regulatory program $r \in R$. In this way, state s_e induces a natural partition of R. The partition contains two regulatory programs $r_1 = (v_1, f_1)$ and $r_2 = (v_2, f_2)$ in the same block if and only if $f_1(s_e(v_1)) = f_2(s_e(v_2))$, i.e., they have the same e-predicted responses. Regulatory programs contained in different blocks of this partition are said to be *distinguished by the predicted model state* s_e (exemplified in Fig.3.2 C).

Distinguishing by an experiment An experiment e may in general define a number of model states predicted for e (section 3.2). We consider a partition $T(e)$ of the set of regulatory programs R induced by an experiment e as the supremum over the set of partitions induced by its predicted model states. To compute the supremum, we first generate a relation, where two regulatory programs are related when they are contained in a common block of at least one partition. Next, we compute a transitive closure of this relation. The resulting equivalence relation has equivalence classes which define the blocks of the supremum partition. The regulatory programs contained in different blocks of $T(e)$ are called *distinguished by the experiment* e. By the definition of a supremum over partitions, the following holds:

- If two regulatory programs are distinguished by an experiment, they are distinguished by all its predicted model states. For example, in Fig.3.2 C, $T(e_2)$ contains r_2 and r_6 in separate blocks, implying that they are distinguished by e_2 and by both model states predicted for e_2. This fact is essential for the correctness of our approach (see "Importance of distinguishing regulatory programs in our framework" below).
- The fact that two regulatory programs are distinguished by all predicted model states of an experiment does not always imply that they are also distinguished by the experiment itself. For example, consider a set of regulatory programs $\{x, y, z\}$, and partitions for hypothetic predicted model states $T_1 = \{x, y\}\{z\}$ and $T_2 = \{x, z\}\{y\}$. z and y are distinguished by both these predicted model states. The supremum over T_1 and T_2 is $\{x, y, z\}$ and it contains z and y in a common block. From the second implication it follows, that our MEED algorithm (section 3.5), which selects experiments that are needed to distinguish between pairs of regulatory programs, may choose experiments that are superfluous for the requirement of differentiating between all their possible predicted profiles (section 3.3.1). This may happen only in the case of cyclic models, which can generate

more than one predicted model state for an experiment.

If an experiment has no predicted model states, it is not informative and its partition includes only one block containing all regulatory programs.

Distinguishing by a set of experiments We call a pair of regulatory programs *distinguished by a set of experiments* $E = \{e_1, ..., e_n\}$ if and only if they are distinguished by at least one of its experiments. Equivalently, we say that E distinguishes between regulatory programs that are contained in separate blocks of a partition $S(E) = T(e_1) \cap ... \cap T(e_n)$ (exemplified in Fig.3.2 D). The partition for an empty set of experiments is a full, one-block partition containing all regulatory programs. Regulatory programs contained in the same block of the partition $S(E)$ are not distinguished by any of the experiments, whereas regulatory programs in different blocks are distinguished by at least one experiment. We say that an experiment set E *distinguishes all* regulatory programs, if its corresponding partition $S(E)$ contains only single-element blocks.

Importance of distinguishing regulatory programs in our framework Note that if E distinguishes between two regulatory programs $r_1 = (v_1, f_1)$ and $r_2 = (v_2, f_2)$, their predicted profiles will be different (i.e., have at least one different predicted response) in all possible combinations of model states predicted for experiments in E. Indeed, by the definition of distinguishing by a set of experiments, there exists an experiment $e \in E$, which distinguishes between r_1 and r_2. Take any possible model state s_e predicted for e. By the definition of distinguishing by an experiment $f_1(s_e(v_1)) \neq f_2(s_e(v_2))$. Therefore, the predicted profiles for r_1 and r_2 are also different in any combination of predicted model states (since they always differ by at least one component, which corresponds to the experiment e). This essential fact ensures that the predicted profiles of any two regulatory programs, which are distinguished by a given set of experiments E, are also different for the predicted model states used by the expansion procedure (chosen by the least discrepancy and corresponding to the steady states the biological system has reached under the experiments; see sections 3.2 and 3.3.1). Thus, by maximizing the number of distinguished regulatory programs, MEED (section 3.5) maximizes also the diversity of predicted profiles used in the expansion procedure.

The FUP score In sections 3.9–3.10.2 we report the performance of a set of experiments E with the *fraction of undistinguished pairs (FUP)*. The score is given by the proportion of regulatory program pairs undistinguished by E out of all possible pairs of regulatory programs:

$$FUP(E) = \frac{\sum_c n_c(n_c - 1)}{|R|(|R| - 1)}$$

where n_c is the size of the c-th block of the corresponding partition $S(E)$ of the set of regulatory programs R. $FUP(E)$ attains values between 0 (all regulatory programs

are distinguished) and 1 (no pair of regulatory programs is distinguished). The more pairs of regulatory programs distinguished by a given set of experiments, the smaller its FUP score. Unlike the ambiguity score (section 3.7), which evaluates the results of expansion utilizing experimental data, FUP evaluates a given ED method based only on model predictions.

3.4 The Experimental Design problem

The task for our algorithm is to select an economical subset of a given set of candidate experiments E that under a given pathway model $\mathcal{M}$ yields different (E, r_1)- and (E, r_2)-predicted profiles for each pair of regulatory programs r_1 and r_2 in a given set R, regardless the combination of model states predicted for the experiments in E. The candidate experiments contain a full set of experiments to choose from; for example, only those experiments that can be conducted in a lab.

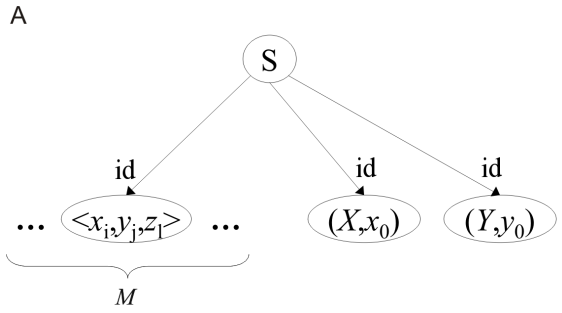
Formulation of the ED problem To fulfill the task, MEED aims to select the smallest subset of the candidate experiments, which can distinguish all regulatory programs in R. From the previous section 3.3.2 we know that for cyclic models such a subset of experiments may be superfluous, but it guarantees that predicted profiles of two different regulatory programs will be different in any combination of predicted model states. In the case when the candidate experiments themselves cannot distinguish all regulatory programs, the identified subset should distinguish between the same pairs of regulatory programs as the full candidate set. We formalize this problem and show that its decision version is NP-complete. To be more general, we relax the setup kept in our analysis. We allow all model variables to have a number of states, which can be different than three, and all regulatory programs to have several (i.e., also more than one) regulators.

Problem 1. ED($\mathcal{M}, R, E, l$)
INSTANCE: A logical model $\mathcal{M}$, a set of regulatory programs R with regulators from the model, a set of candidate experiments E on $\mathcal{M}$, and a number $l \leq |E|$.
QUESTION: Is there a subset of size l of E which distinguishes all regulatory programs in R.

Proposition 1. The ED problem is NP-complete.

Proof. It is easy to see that ED$\in$NP, since a nondeterministic algorithm would only need to guess a subset $E' \subseteq E$ of size l, in polynomial time construct a partition $S(E')$ (section 3.3.2) and verify whether it is an identity partition on R.

We show that the ED problem is NP-complete by reduction of the 3-DIMENSIONAL MATCHING (3DM) problem [38]. An instance of 3DM is defined by a set $M \subseteq X \times Y \times Z$, where $X \cap Y = X \cap Z = Y \cap Z = \emptyset$ and $|X| = |Y| = |Z| = m$. A



B

REGULATORY PROGRAMS	
x_i	$S \vee (X,x_0) \vee <x_i,\bullet,\bullet>$
x_0	$S \vee (X,x_0)$
y_j	$S \vee (Y,y_0) \vee <\bullet,y_j,\bullet>$
y_0	$S \vee (Y,y_0)$
z_l	$S \vee <\bullet,\bullet,z_l>$
z_0	S

Figure 3.3: Construction of the ED problem for the 3DM problem. (A) The model. The nodes correspond to variables and arrows to regulation functions. S – the stimulator variable. There are triplet variables corresponding to the triplets in $\mathcal{M}$ and two additional variables (X, x_0) and (Y, y_0). id – the identity function. **(B) The regulatory programs.** Left column of the table – identifiers of the programs, right column – their regulation functions and regulators (specified as function arguments). $< x_i, \bullet, \bullet >$ denotes all (possibly several) triplet variables with x_i on the first coordinate.

solution to 3DM is a *matching* for M of size m, i.e., a subset $M' \subseteq M$ such that no elements of M' agree on any coordinate. Let $X = \{x_1, ..., x_m\}$, $Y = \{y_1, ..., y_m\}$ and $Z = \{z_1, ..., z_m\}$. For a given 3DM problem we define an instance $(\mathcal{M}, R, E, m + 2)$ of the ED problem and prove that there exists a matching of size m if and only if there exists a set of $m + 2$ experiments in E for $\mathcal{M}$ which distinguish all regulatory programs in R.

We now define an instance $\mathcal{M}, R, E, m + 2$ of the ED problem for a given instance M of the 3DM problem. The set of variables V in model $\mathcal{M}$ contains stimulator variable S, a set of $|M|$ variables corresponding to the triplets in M (referred to as the *triplet* variables) and two additional variables, labeled (X, x_0) and (Y, y_0), $V = M \cup \{S, (X, x_0), (Y, y_0)\}$. All variables can be in one of the two states: 0 and 1. The regulation functions are defined in Boolean logic. The states of the stimulator S determine the states of all remaining variables through an identity function, i.e., for a given model state s, $f_v(s(S)) = s(S)$, for each $v \in V \setminus \{S\}$ (Fig.3.3 A). We define the set of regulators to be all variables in the model.

Next, we define a set R of $3m + 3$ regulatory programs for the model $\mathcal{M}$, each with the same regulation function: the Boolean alternative (logical "$\vee$"). Intuitively, the regulatory programs correspond to the elements in the sets X, Y, and Z and three additional entities denoted x_0, y_0 and z_0. Each regulatory program denoted x_i, $1 \leq i \leq m$, has a set of regulators: S, (X, x_0) and all triplet variables, which have x_i on the first coordinate. The regulatory program x_0 has two regulators: S and (X, x_0). Each regulatory program denoted y_j, $1 \leq j \leq m$, has the regulators: S, (Y, y_0) and all triplet variables, which have y_j on the second coordinate. The regulatory program y_0 has two regulators: S and (Y, y_0). Finally, each regulatory program denoted z_l, $1 \leq l \leq m$ has the regulators: S, and all triplet variables, which have z_l on the third coordinate. The regulatory program z_0 has one regulator S (Fig.3.3 B).

Finally, we define the set of candidate experiments E on the model $\mathcal{M}$. Recall from section 3.2 that an experiment is given by the stimulation, perturbed variable and its perturbation state. We assume that the set $P = V \setminus \{S\}$ of all non-stimulator variables can be perturbed. Since S determines by identity the state of all other model variables and is an alternative regulator in all regulatory programs, the following experiments do not distinguish between any pairs of regulatory programs: for $v \in P$ and $i \in \{0, 1\}$, the experiments of the form: $S = i, \emptyset$ (an experiment with any stimulation i, where no variable is perturbed), as well as $S = 1, v = i$ (an experiment where the stimulation is high, with any perturbed variable v and any perturbation state i), and $S = 0, v = 0$. Thus, from all possible experiments, it suffices to consider only the set of candidates $E = \{e^v : v \in P\}$, where e^v denotes an experiment of the form: $S = 0, v = 1$.

Assume there exists a 3D-matching $M' = \{t_1, ..., t_m\}$ for $\mathcal{M}$. We show that in this case the set of $m + 2$ experiments $E' = \{e^{t_1}, ..., e^{t_m}, e^{(X,x_0)}, e^{(Y,y_0)}\}$ on the model $\mathcal{M}$ distinguishes all regulatory programs in R. Indeed, the experiments $e^{(X,x_0)}$ and $e^{(Y,y_0)}$ together induce a partition $S(\{e^{(X,x_0)}, e^{(Y,y_0)}\})$ on R into three blocks: $\{x_0, x_1, ...x_m\}$, $\{y_0, y_1, ..., y_m\}$, and $\{z_0, z_1, ..., z_m\}$. Since M' is a 3D-matching, for each $x_i \in X$ it contains a single triplet $t =< x_i, y, z >\in M'$ with x_i on the first coordinate. The corresponding experiment $e^t \in E'$ induces a partition $T(e^t)$ on R into two blocks: $\{x_i, y, z\}$ and $R \setminus \{x_i, y, z\}$. Intersecting $T(e^t)$ with $S(\{e^{(X,x_0)}, e^{(Y,y_0)}\})$ results in a partition which contains a singleton $\{x_i\}$. Since all regulatory programs with labels from X have their corresponding experiments in E', they are distinguished by E' from all other regulatory programs, in particular from x_0. Similar argument holds for any regulatory program with labels from Y or Z.

Assume there exists a set of $m + 2$ experiments $E' \subseteq E$ distinguishing all regulatory programs in R. We show that in this case there exists a matching of size m for the $3DM$ problem. Note that $e^{(X,x_0)} \in E'$ and $e^{(Y,y_0)} \in E'$. Otherwise either the regulatory program x_0 would not be distinguished from z_0 or y_0 would not be distinguished from z_0. In the remaining m experiments the triplet variables are perturbed. Denote the set of m perturbed triplets as $M' = \{t \in M : e^t \in E' \setminus \{e^{(X,x_0)}, e^{(Y,y_0)}\}\}$. Suppose that for some $x \in X$ there exist no triplet in M' with x on the first coordinate. Then the regulatory programs x and x_0 are not distinguished by E', which contradicts our assumption. Similar argument holds for any $y \in Y$ and $z \in Z$ on the second and third coordinate, respectively. Thus M' contains m triplets that together have all m elements from X on the first, all elements of Y on the second, and all elements of Z on the third coordinate. Thus, the triplets in M' do not agree on any coordinate, and M' is a solution to the given $3DM$.

□

3.5 The MEED algorithm

Notions of entropy used in the algorithm To evaluate the ability of a set of experiments to distinguish regulatory programs, MEED uses an *entropy score*. Let E be a given set of experiments and R a set of regulatory programs. Assume that E induces a partition $S(E)$ of R into C disjoint blocks (section 3.3.2). The score is defined as

$$H(E) = -\sum_{c=1}^{C} \frac{n_c}{|R|} \log\left(\frac{n_c}{|R|}\right),$$

where n_c is the number of regulatory programs in block c, $1 \leq c \leq C$. If all regulatory programs are distinguished by the set of experiments E, then $C = |R|$ and the corresponding score is $H(E) = log(|R|)$. If all regulatory programs are undistinguished by E, there is only one block in the partition, $C = 1$ and $H(E) = 0$. Intuitively, the higher the entropy score, the higher the ability of the set of experiments to distinguish between the regulatory programs. Accordingly, an *entropy gain* $H(e|E)$ is given by $H(e|E) = H(E \cup e) - H(E)$, where $S(E \cup e) = S(E) \cap T(e)$ (i.e., the additional experiment introduces a finer partition of the set of regulatory programs). Entropy gain evaluates how much the joint ability to distinguish between regulatory programs improves when the experiment e is added to the set of experiments E.

The algorithm To obtain a practical solution for the untractable ED problem, MEED implements a greedy approximation algorithm.

```
MEED(E, R)
1  E* ← ∅
2  while H(E*) < log(|R|)
3  do e ← arg max_{e∈E} H(e|E*)
4      if H(e|E*) = 0
5        then break
6      E* ← E* ∪ e
7  return E*
```

The algorithm takes as input a set of candidate experiments E to select from, and the set of regulatory programs R that it has to distinguish (indicated in the heading). The procedure starts from an empty set of experiments E^*, corresponding to a full partition on the set of regulatory programs (line 1), and keeps choosing new experiments one by one. In each greedy step, experiment e that maximizes the entropy gain for the current experiment list E^* is chosen (line 3) and added to E^* (line 6; equivalently, its corresponding partition is intersected with the partition for E^*). The iteration finishes either when all regulatory programs are distinguished (line 2)

or when no additional experiment can give improvement (line 4). The latter case means that the selected list E^* does not distinguish all regulatory programs, but the same as the candidate experiments in E. The output of MEED is the ordered subset $E^* \subseteq E$ of chosen experiments.

The upper bound computational cost of this algorithm is $O(|R| * |E|^2)$. This estimation assumes a pessimistic scenario that the iteration does not end until all experiments from E are added one by one to E^*, and that in each step all unused experiments are tried to select the best one. The factor $|R|$ comes from the fact that in order to compute the entropy gain, intersection of two partitions of the set R needs to be generated, which can be implemented in $O(|R|)$.

3.6 Approximation factor of the MEED algorithm

In this section we prove the correctness of our algorithm MEED and derive its approximation factor. Below we refer to the optimization version of the experimental design problem as ED$(\mathcal{M}, R, E)$.

Proposition 2. For a given instance $(\mathcal{M}, R, E)$ of the ED optimization problem, the MEED algorithm returns a set $E^* \subseteq E$ of the candidate experiments, which distinguishes the same pairs of regulatory programs as E. Moreover, $|E^*| \leq |E^{opt}|(1+\ln(|R|) + \ln(\log(k)))$, where E^{opt} is the smallest-size solution of the ED problem and k is the number of states each variable can have.

By Proposition 2 it holds that if E distinguishes all regulatory programs in R, also E^* will distinguish all regulatory programs in R.

To prepare grounds for the proof of Proposition 2, we define an auxiliary optimization problem on partitions. Adapting the reasoning schema of Konwar *et al.* [74], we propose a generic heuristic GENERIC-GREEDY solving the partition problem and derive its approximation factor. Our proof of Proposition 2 relies on the fact that ED can be viewed as the problem on partitions and that the MEED algorithm implements this generic heuristic.

Notation and basic notions Let $\mathcal{P}$ denote a set of partitions of a given set R and let $\mathcal{P}^+$ be the closure of $\mathcal{P}$ under finite intersections. The intersection of all partitions from a set $\mathcal{P}$ is denoted by $\bigcap \mathcal{P}$. We keep the following convention: T, T', T'' range over $\mathcal{P}$ and S, S', S'' range over $\mathcal{P}^+$. Full (one block) partition of R is denoted $\{R\}$ and the identity partition into singletons is denoted id_R. For two partitions T' and T'' we write $T'' \leq T'$ and say T'' is *included* in T' if for any block $T''_i \in T''$, there exists a block $T'_j \in T'$ such that $T''_i \subseteq T'_j$ (see example illustrated in Fig.3.4). Note that, for any pair of partitions S and T, $S \cap T \leq S$.

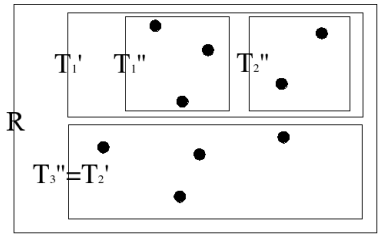


Figure 3.4: Partition inclusion. Example. T' and T'' are partitions of set R satisfying $T'' \leq T'$.

Let $\Phi : \mathcal{P}^+ \to \{x \in \mathbb{R} : x \geq 0\}$ be a given strictly increasing function, which satisfies the following conditions:
(A0) $\Phi(S) = 0 \Leftrightarrow S = id_R$.
(A1) For each $T \in \mathcal{P}$ and $S \in \mathcal{P}^+$, let $\Delta_T(S) = \Phi(S) - \Phi(S \cap T)$ be a *gain* function determined by Φ. If $\Delta_T(S) > 0$, then $\Delta_T(S) \geq 1$.
Note that by definition Δ_T is strictly increasing.

An auxiliary problem on partitions

We now define an auxiliary optimization problem on partitions.

Problem 2. PARTITION($\mathcal{P}, R$)
INSTANCE: A set $\mathcal{P}$ of partitions on a set R.
QUESTION: Find a subset $\mathcal{P}^{opt} \subseteq \mathcal{P}$ of minimal size such that $\bigcap \mathcal{P}^{opt} = \bigcap \mathcal{P}$.

Of course, a solution $\mathcal{P}^{opt}$ of a given instance $(\mathcal{P}, R)$ of the PARTITION problem satisfies $\bigcap \mathcal{P}^{opt} = id_R$ if and only if $\bigcap \mathcal{P} = id_R$.

A greedy heuristic for solving the PARTITION problem

Below we present a generic greedy heuristic for solving the PARTITION problem for a given set $\mathcal{P}$ of partitions of a set R.

```
GENERIC-GREEDY(P, R)
1  P* ← ∅
2  S ← {R}
3  while Φ(S) > 0
4  do T ← arg max_{T∈P} Δ_T(S)
5      if Δ_T(S) = 0
6        then break
7      S ← S ∩ T
8      P* ← P* ∪ {T}
9  return P*
```

Lemma 1. Given an instance $(\mathcal{P}, R)$ of the PARTITION problem, the GENERIC-GREEDY algorithm finds a set of partitions $\mathcal{P}^* \subseteq \mathcal{P}$, which satisfies: (i) $\bigcap \mathcal{P}^* = \bigcap \mathcal{P}$ and (ii) $|\mathcal{P}^*| \leq 1 + \ln(\Delta^{max})$, where $\mathcal{P}^{opt}$ is the smallest-size solution of the PARTITION problem and $\Delta^{max} = \max_{T \in \mathcal{P}} \Delta_T(\{R\})$.

Proof. **(i)** We show that GENERIC-GREEDY stops and returns a set of partitions $\mathcal{P}^*$ with $\bigcap \mathcal{P}^* = \bigcap \mathcal{P}$. The case of $|R| = 1$ is trivial: the **while** loop of the algorithm is never entered since at the entry $\Phi(S) = \Phi(\{R\}) = 0$ and $\bigcap \mathcal{P}^* = \bigcap \mathcal{P} = \{R\}$. Consider $|R| > 1$. By condition (A0), in this case $\Phi(S) > 0$ and the loop is entered. Assume first, that the condition in line 5 is never satisfied. In this case, in each greedy step a partition $T \in \mathcal{P}$ giving a positive non zero $\Delta_T(S)$ for the current partition S can be found (line 4; if several T maximize $\Delta_T(S)$, the algorithm chooses one at random). Intersection of the current S with the found T will give a new S that is included in the previous one (line 7). Thus by the monotonicity of Φ the value of $\Phi(S)$ decreases. Therefore the **while** loop terminates in a finite number of steps, and upon termination we have $\Phi(S) = 0$. By (A0) we know that then $\bigcap \mathcal{P}^* = S = id_R$. Since $\mathcal{P}^* \subseteq \mathcal{P}$, then also $\bigcap \mathcal{P} = id_R$ and (i) holds. Next, assume otherwise, that at some point of the iteration the condition in line 5 is satisfied and the **break** statement is executed. With this assumption for any $T \in \mathcal{P}$ it holds that $\Phi(\bigcap \mathcal{P}^*) - \Phi(\bigcap \mathcal{P}^* \cap T) = \Delta_T(\bigcap \mathcal{P}^*) = 0$. Since $\bigcap \mathcal{P}^* \cap T \leq \bigcap \mathcal{P}^*$ and Φ is strictly increasing then $\bigcap \mathcal{P}^* \cap T = \bigcap \mathcal{P}^*$. Thus we have $\bigcap \mathcal{P} = \bigcap \mathcal{P}^* \cap (\mathcal{P} \setminus \mathcal{P}^*) = \bigcap \mathcal{P}^*$ and (i) holds.

(ii) To prove the approximation factor for the GENERIC-GREEDY algorithm, we first introduce notions needed to evaluate the cost of the optimal and greedy solutions $\mathcal{P}^{opt}$ and $\mathcal{P}^*$. Let $|\mathcal{P}^{opt}| = n$ and $|\mathcal{P}^*| = m$. For $1 \leq g \leq m$, T_g denotes the partition selected in the g-th step of GENERIC-GREEDY, whereas T_i^{opt}, $1 \leq i \leq n$, is the i-th element of the optimal solution, taking any order on $\mathcal{P}^{opt}$. For $1 \leq g \leq m$ we consider the intersections of g partitions selected by the greedy algorithm, denoted by $S_g = T_1 \cap ... \cap T_g$. Similarly, for the optimal partitions we define $S_i^{opt} = T_1^{opt} \cap ... \cap T_i^{opt}$, where $1 \leq i \leq n$. We set $S_0 = S_0^{opt} = \{R\}$. We denote the gain for the greedy choice of T_g in the g-th step of GENERIC-GREEDY towards a given intersection of greedy and optimal partitions by $\Delta_i^g = \Delta_{T_g}(S_{g-1} \cap S_i^{opt})$, where $1 \leq g \leq m$ and $0 \leq i \leq n$. Similarly, let the gain for the optimal choice of T_i^{opt} towards the intersection of the greedy and optimal partitions be denoted $\delta_i^g = \Delta_{T_i^{opt}}(S_g \cap S_{i-1}^{opt})$, where $1 \leq i \leq n$, and $0 \leq g \leq m$.

Note that $\Delta_0^g \geq \Delta_1^g \geq ... \geq \Delta_n^g = 0$ for $1 \leq g \leq m$. Indeed, for every such g and $0 \leq i \leq n-1$, by the fact that $S_{g-1} \cap S_{i+1}^{opt} \leq S_{g-1} \cap S_i^{opt}$, and by the monotonicity of Δ we have $\Delta_{i+1}^g = \Delta_{T_g}(S_{g-1} \cap S_{i+1}^{opt}) \leq \Delta_{T_g}(S_{g-1} \cap S_i^{opt}) = \Delta_i^g$.

Similarly, $\delta_i^0 \geq ... \geq \delta_i^m = 0$ for every $1 \leq i \leq n$.

Moreover, the following inequalities hold:

$$\Delta_0^g \geq \Delta_{T_i^{opt}}(S_{g-1}) \geq \Delta_{T_i^{opt}}(S_{g-1} \cap S_{i-1}^{opt}) = \delta_i^{g-1} \tag{3.7}$$

for every $1 \leq g \leq m$ and $1 \leq i \leq n$. The first inequality follows from the fact that $\Delta_0^g = \Delta_{T_g}(S_{g-1})$ is obtained by the greedy partition selection in the g-th step of the GENERIC-GREEDY algorithm and thus must be at least as high as $\Delta_T(S_{g-1})$ for any $T \in \mathcal{P}$, also T_i^{opt}. The second inequality follows from the monotonicity of Δ.

For the analysis of the size of the greedy solution as compared to the optimal one, we assign a cost to each pair of optimal and greedy partitions T_i^{opt}, T_g:

$$c_i^g = \begin{cases} \ln(\delta_i^{g-1}) - \ln(\delta_i^g) & \text{if } \delta_i^{g-1} \geq \delta_i^g > 0 \\ \ln(\delta_i^{g-1}) + 1 & \text{if } \delta_i^{g-1} > \delta_i^g = 0 \\ 0 & \text{if } \delta_i^{g-1} = \delta_i^g = 0 \end{cases}$$

By definition every δ when positive, is higher than 1 (condition (A1)), and thus every cost is nonnegative.

From now on, our reasoning can follow exactly the one of Konwar *et al.* [74]. Since $\delta_i^m = 0$, the total cost assigned to a given optimal partition T_i^{opt} is a telescopic sum $\sum_{g=1}^m c_i^g = 1 + \ln(\delta_i^0) \leq 1 + \ln(\Delta^{max})$. The cost of all partitions of the optimal solution is the same, so the overall cost of the optimal solution is at most $n(1 + \ln(\Delta^{max}))$. We show that at the same time the cost of each greedily chosen T_g is at least equal to 1. By Eq.(3.7) for every $1 \leq i \leq n$ it holds that $c_i^g \geq (\delta_i^{g-1} - \delta_i^g)/\Delta_0^g$ (refer to Figure 3 by Konwar *et al.* [74] for a graphical explanation). By $\delta_i^{g-1} - \delta_i^g = \Delta_{i-1}^g - \Delta_i^g$ and $\Delta_n^g = 0$, we have $\sum_{i=1}^n c_i^g \geq \sum_{i=1}^n (\Delta_{i-1}^g - \Delta_i^g)/\Delta_0^g = 1$.

Thus, $m \cdot 1 \leq \sum_{g=1}^m \sum_{i=1}^n c_i^g = \sum_{i=1}^n \sum_{g=1}^m c_i^g \leq n(1 + \ln(\Delta^{max}))$, which completes the proof.

□

An example function satisfying conditions (A0) and (A1) In the following, we will use a function $\rho : \mathcal{P}^+ \to \{x \in \mathbb{R} : x \geq 0\}$, defined as $\rho(S) = \sum_{c=1}^C n_c \log(n_c)$, where partition S contains C blocks and n_c is the number of elements in block c ($1 \leq c \leq C$).

Proposition 3. For every instance $(\mathcal{P}, R)$ of the PARTITION problem the function ρ satisfies (A0) and (A1).

Proof. The function ρ satisfies (A0) and (A1) for any $\mathcal{P}$. ρ is related to an entropy measure $H(T)$ for a partition of the set R into C blocks:

$$H(T) = -\sum_{c=1}^C \frac{n_c}{|R|} \log\left(\frac{n_c}{|R|}\right) = \log(|R|) - \frac{1}{|R|}\rho(T). \tag{3.8}$$

Thus, small values of ρ are equivalent to high partition entropy. We base on the properties of entropy (see Cover and Thomas [26]) to prove the properties of ρ.

For any partitions T and T', such that $T < T'$, the properties of entropy imply that $H(T) > H(T')$, and by Eq.(3.8) $\rho(T) < |R|\log(|R|) - |R|H(T') = \rho(T')$, which proves that ρ is strictly increasing. For the identity partition we obtain $\rho(id_R) = \sum_{c=1}^{|R|}\log(1) = 0$. For any partition T, if $\rho(T) = 0$, then the entropy obtains its maximum value $H(T) = \log(|R|)$ and this is true only for $T = id_R$. Therefore, ρ satisfies (A0).

By monotonicity of ρ, functions Δ_T determined by ρ assume only nonnegative values. Let $S \in \mathcal{P}^+$ be such that $\rho(S) > 0$. For any $T \in \mathcal{P}$ if $S = T$ then $\Delta_T(S) = 0$. For $S \neq T$, $\Delta_T(S) > 0$. The gain $\Delta_T(S)$ can be obtained by a sum of the gains on the separate blocks of S when intersected with T. Thus the minimal non-zero gain is obtained when only one block of S contributes the smallest possible nonzero gain when intersected with $T \neq S$. Let c be the index of such a block of S of size n_c which by the intersection is divided into B blocks of size n_b each, $1 \leq b \leq B$ and $\sum_{b=1}^{B} n_b = n_c$. Note that such a division into blocks defines a partition T^c on a set of elements in the c-th block. Then the minimal nonzero gain

$$\min \Delta_{T \neq S}(S) = n_c \log(n_c) - \sum_{b=1}^{B} n_b \log(n_b) = -n_c \sum_{b=1}^{B} \frac{n_b}{n_c} \log\left(\frac{n_b}{n_c}\right) = n_c H(T^c).$$

By relation of entropy to the maximum probability (Property 1.24 in the book by Tanieja [119]) $\frac{1}{2}H(T^c) > 1 - p_{max} \geq 1/n_c$. The second inequality holds since in our case p_{max} can at most be equal to $(n_c - 1)/n_c$ (the ratio of the size largest possible block of the partition T^c to the size of the whole c-th block of S). Therefore $\min \Delta_{T \neq S}(S) > 2$ which proves that ρ satisfies the condition (A1).

□

MEED implements GENERIC-GREEDY Having Lemma 1 and Proposition 3, we prove Proposition 2.

Proof. **Proposition 2.** Recall from section 3.3.2 that every experiment e on a given model defines a partition $T(e)$ on the set of regulatory programs R and a set of experiments E defines a partition $S(E)$. Therefore a given instance $(\mathcal{M}, R, E)$ of the ED problem, where $E = \{e_1, ..., e_n\}$, defines an instance $(\mathcal{P}, R)$ of the PARTITION problem, where $\mathcal{P} = \{T(e_1), ..., T(e_n)\}$.

We show line by line that the MEED algorithm (section 3.5) implements the GENERIC-GREEDY algorithm. The set E^* returned by MEED corresponds to $\mathcal{P}^*$ returned by GENERIC-GREEDY and $S(E^*)$ corresponds to S. With $S(\emptyset) = \{R\}$, line 1 in the former algorithm implements lines 1 and 2 in the latter. As shown in the proof of Proposition 3, ρ belongs to the class of functions Φ. First, by definition, the entropy score (section 3.5) satisfies $H(E^*) = \log(|R|) - \frac{1}{|R|}\rho(S(E^*))$, i.e., H is only a linear function of ρ. Therefore the conditions of the **while** statements in the two algorithms are equivalent: $H(E^*)$ reaches its maximum value $\log(|R|)$ if and only if

$\rho(S(E^*)) = 0$. Note also, that $H(e|E^*) = \frac{1}{|R|}\Delta_{T(e)}(S(E))$, where the function $\Delta_{T(e)}$ is determined by ρ. Therefore the lines 3–5 of the MEED algorithm implement the lines 4–6 in GENERIC-GREEDY. Line 6 in our algorithm implements lines 7 and 8 in the generic heuristic: adding e to E^* corresponds to $\mathcal{P}^* \cup T(e)$ and $S(E^*) \cap T(e)$.

Therefore, by Proposition 3 and Lemma 1, MEED stops and finds an approximate solution. To derive the approximation factor, we calculate $\Delta^{max} = \max_T \Delta_T(\{R\})$, determined by the function ρ, knowing that $T = T(e)$ for some experiment $e \in E$. Recall that each of the regulators in a given model can have k possible states. Therefore also for each regulatory program there can be k possible responses of its potential target and for any experiment e, its partition $T(e)$ can divide R into at most k blocks. By the relation to entropy, for a given e, $\rho(T(e))$ obtains minimal value for partition $T(e)$ having maximum entropy. By the maximality property of entropy (Property 1.15 in the book by Tanieja [119]) $\Delta^{max} = \rho(\{R\}) - \rho(T(e))$, where $T(e)$ is a partition into k sets of equal sizes. Thus $\Delta^{max} = |R|\log(|R|) - |R|\log(|R|/k) = |R|\log(k)$ and from Lemma 1, $1 + \ln(|R|\log(k))$ is the approximation factor for our algorithm. □

3.7 Expansion procedure

The *expansion procedure* aims to detect which of the genes measured in a given set of experiments E could be regulated by the predefined regulatory programs in a set R. The procedure applies probabilistic matching between each (E, r)-predicted profile ($r \in R$), and the observed expression profiles of the measured genes. Let $P(\vec{x}_g = \vec{y})$ denote the probability of a match between a given observed profile $\vec{x}_g$ of a gene g and a given (E, r)-predicted profile $\vec{y}$ of a certain regulatory program r in a set of experiments E. This probability depends on probabilities of a match between the observed and predicted responses to experiment e in E:

$$P(\vec{x}_g = \vec{y}) = \prod_{e \in E} P(\vec{x}_g(e) = \vec{y}(e)),$$

where $\vec{x}_g(e)$ and $\vec{y}(e)$ denote the observed and the predicted response to experiment e, respectively. For the expansion procedure, we can assume that there is a unique predicted response to each experiment. The response is defined by the model state with the least discrepancy to the observed state (sections 3.2 and 3.3.1).

The probabilities $P(\vec{x}_g(e) = \vec{y}(e))$ are estimated using the probabilities of differential expression in experiment e, given by the POE method (section 2.4). The outcome of applying POE to the data are the probabilities of low, baseline and high expression of each gene g in experiment e, which we directly translate to probabilities of g being: *down-regulated* in e (denoted $p^{-1}_{g,e}$), *unchanged* ($p^0_{g,e}$), and *up-regulated* ($p^1_{g,e}$), respectively. The probabilities satisfy $\min(p^{-1}_{g,e}, p^1_{g,e}) = 0$ and $p^0_{g,e} = 1 - \max(p^{-1}_{g,e}, p^1_{g,e})$. Having this, we set $P(\vec{x}_g(e) = \vec{y}(e)) = p^{\vec{y}(e)}_{g,e}$.

For each gene g, we find the (E, r)-predicted profile $\vec{y}$ that matches its observed profile $\vec{x}_g$ with the highest probability $P(\vec{x}_g = \vec{y})$. We conclude that the gene is controlled by r if this probability exceeds a threshold defined as $p^{|E|}$, where p is a user-defined parameter that corresponds to the cut-off for matching probability of each of the responses and $|E|$ is the profile length (in our analysis, we set the parameter $p = 0.7$). In such a case, we say that the regulatory program *matches* the gene. A group of genes that match the same regulatory program constitutes a *regulatory module* (Fig.3.1 C). Hence, a regulatory module corresponds to a cluster of genes that are co-expressed and are predicted to be co-regulated by the same regulator in the model and through a common regulatory mechanism.

Of course, matching of profiles in the expansion procedure can be hampered. If any two regulatory programs cannot be distinguished by the input experiments (section 3.3.2), their predicted profiles are identical. In such a case, a single observed profile of genes in a regulatory module could match more than one predicted profile, making it impossible to identify a unique regulatory program for this module. Such regulatory module is called an *ambiguous module*. To evaluate expansion quality, we use an *ambiguity score* reporting the average number of regulatory programs that were identified for each gene. Intuitively, the more regulatory programs matching each ambiguous module, and the more genes it contains, the higher the overall ambiguity score.

In our framework, to uniquely identify regulatory modules downstream of a given model, the expansion procedure uses experiments E^* suggested by MEED (Fig.3.2 B, C). Both MEED and the expansion procedure utilize the same model and regulatory programs. Therefore, if experiments in E^* distinguish between all pairs of regulatory programs, all identified regulatory modules are unambiguous.

3.8 Alternative ED approaches

In this section we first cover alternative ideas for selecting experiments based only on the prior model of a given signaling pathway. Next, we compare our ED framework to extant ED approaches, which build models of the system and utilize perturbation data.

3.8.1 Alternative ways of model-based ED

Independent entropy-based experiment scoring Recall that MEED scores a subset of experiments according to their joint ability to distinguish between regulatory programs. In the results sections 3.9–3.10.2 we compare our algorithm to independent experiment scoring, referred to as INDEP. INDEP ranks the experiments according to the same score as MEED (the entropy score, section 3.5), but the score is assigned

only to each experiment independently and experiments are ordered by decreasing score. For example, in Fig.3.1 B experiment e_2 has the best (i.e., the highest) entropy score, whereas experiments e_1, e_3, e_5 and e_6 all have equal entropy scores, lower than the entropy score of e_2 but higher than of e_4. Here, INDEP prioritizes the experiments in the following way: e_2, next e_1, e_3, e_5 and e_6 in random order, and last, e_4. Note that experiment e_5, with the second best score, is dispensable given the first chosen experiment e_2: the pairs of regulatory programs distinguished by e_5 are a subset of pairs distinguished by e_2 (compare the predicted responses in respecting columns for e_5 and e_2 in Fig.3.1 B). In this way, INDEP prioritizes highly informative experiments, but several of them might be redundant and distinguish between the same pairs of regulatory programs. In contrast, after choosing e_2 as the first experiment, MEED discards e_5 since it gives no entropy gain when added to e_2 (i.e., no improvement in distinguishing regulatory programs, section 3.3).

Network-based ED methods. MEED is also compared to *network-based* methods, which prioritize the perturbed variables according to key topological features of the model graph: in- and out-degree, total number of connections, topological and reverse topological order (referred to as IN-DEGREE, OUT-DEGREE, CONNECTIONS, TOPOL and REV-TOPOL, respectively). In each step, an experiment chosen in a given order includes exactly one perturbed variable and there are no two experiments with the same perturbed variable. In case of ties (e.g., in IN-DEGREE, if two nodes have the same number of incoming edges and therefore the same ranking), the variables are chosen at random. The topological order on a cyclic graph is computed following the standard zero-indegree algorithm for topological sorting on directed acyclic graphs [23], with the exception that when a cycle is detected (i.e., there are no zero-indegree nodes), a randomly chosen node from the cycle is first added to the order, and next removed from the graph together with all its adjacent edges, and the standard iteration is continued.

In each step of experiment selection, the order on model variables assumed by a given network-based method defines only the variable to be perturbed in this step. To fully define the experiment in this step, the stimulation and perturbation states need to be fixed. With this respect, we divide the network-based methods into two types:

- *random network-based methods* assign the perturbed variable a random perturbation state and pick random stimulation,
- *hybrid network-based methods* follow the reasoning and scoring of our algorithm MEED: first, a set of all those experiments that perturb the specified variable is collected. From this candidate set, the hybrid methods choose the experiment that gives the highest entropy gain when added to the previous experiments.

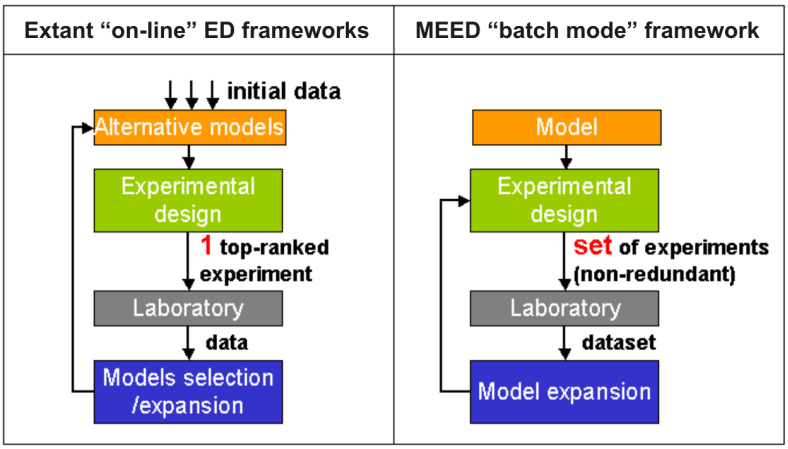


Figure 3.5: MEED framework versus extant ED frameworks

3.8.2 Comparison to extant ED approaches

Here, we discuss the differences between our MEED framework and the approaches of Barrett and Pallson [10], Ideker *et al.* [59], Yeang *et al.* [136], and Vatcheva *et al.* [124].

An "on-line" framework of the extant ED approaches versus a "batch mode" of the MEED framework Fig.3.5 summarizes the general differences between our MEED framework and the solutions common to the extant ED frameworks. In a standard systems biology framework, readily available data is gathered to generate initial alternative models of the biological system (for the method of Barrett and Pallson (2006), there is only one initial model). Next, an experiment selection method (specific for each approach) is used to rank each individual candidate experiment. The common framework is an iterative "on-line" learning process: The top ranked experiment is chosen to be performed in a lab and the outcome is utilized to re-compute the model(s). The MEED framework extends this common approach by taking advantage of a prior model. MEED can design experiments solely based on the model predictions without access to previous experimental data. Most importantly, the algorithm works in a "batch mode": it provides a whole set of non-redundant, informative experiments that can be performed together in a lab and finally utilized by the expansion procedure. Results of the expansion can be iteratively improved by applying MEED to choose from a different set of candidate experiments to complement those that were already performed.

Differences in the ED algorithms Tab. 3.1 summarizes in detail the basic principles of MEED and the extant ED procedures. The main methodological difference between the MEED algorithm and other ED approaches that also use the notion of entropy (all reviewed here but Barrett and Palsson, 2006) is the following: MEED

Method	"Batch mode"	Prior model	Model	Prior data	Selecting next experiment e	Stop criterion
MEED	+	+	Logical	-	e maximizes entropy gain of the current set of experiments, increasing the diversity of states predicted by the model for potential target genes.	Chosen set of experiments has the maximum entropy score, /no unused experiment gives entropy gain, /all experiments used.
Barrett and Palsson, 2006	-	+	ODEs, rFBA, Boolean	B, E	e is a change of environment and a knockout of a group of TFs that are most interconnected and differentially activated in these environments.	All unused experiments involve environments activating TFs without predicted target genes, /all experiments used.
Ideker et al., 2000	-	-	Boolean	E	e gives maximum expected decrease in entropy of alternative models. Computed based on model states predicted for e on each of the models.	No unused experiment gives entropy decrease, /all experiments used.
Yeang et al., 2005	-	-	Physical network models	B, P, E	Expected information gain: e maximizes difference in entropy of alternative models before and after performing e. Calculated based on model predictions of change of model states upon e.	Minimum entropy on models: One model has high, whereas other models have zero probability, /no unused experiment gives entropy reduction, /all experiments used.
Vatcheva et al., 2006	-	-	Semi–quantitative	V	Expected information increment: e maximizes expected difference in entropy of alternative models before and after performing e. Calculated based on model e. predictions for the temporal evolution of the system state.	Minimum entropy on models: One model reaches cut-off, whereas others have probability zero, /no unused experiment gives entropy reduction, /all experiments used.

Table 3.1: Comparison table of MEED and extant the ED algorithms. The table summarizes the methods by the following features: **Batch mode** – taking into account dependencies between the experiments and the ability to provide a non–redundant set of experiments on output, **Prior model** – utilizing a prior model of the studied system in the decision process, **Model** – the modeling mathematical formalism (rFBA – regulated Flux Balance Analysis, ODEs – Ordinary Differential Equations), **Prior data** – experimental data (B – binding, E – expression, P – protein-protein interactions, V – varies depending on the application of the theoretical approach; in the original case study the data consisted of amount of phytoplankton biomass and concentration of the remaining substrate) required prior to ED procedure and for each next experiment to be selected, **Selecting an experiment** e – a criterion for selecting the next experiment, and a **Stop criterion** for the algorithm.

pre-calculates a set of possible regulatory programs and requires from the new experiment to increase the entropy of the set of experiments (so that the extended set of experiments imposes the most "even" partition on the set of regulatory programs), while the other approaches work with an ensemble of alternative models and aim to decrease the entropy of this ensemble (so that, with the new experiment, one or few models have significantly higher probability than the remaining ones).

Implementation of the methods by Ideker et al. (2000) and by Barrett and Pallson (2006). To perform comparative analysis in section 3.10.2 we implemented two extant ED methods, introduced by Ideker *et al.* (2000) and by Barrett and Pallson (2006). The two methods have their own modeling approach and differ in the way they utilize prior experimental data and measurements from each selected experiment. To focus the comparison exclusively on experimental design (Tab.3.1, "Selection of next experiment e" and "Stop criterion"), we utilized the expansion procedure to unify the modeling part of the two methods. First, to give initial information required by both methods, we took the regulatory modules identified by model expansion using the four highest priority experiments proposed by MEED. Second, the regulator-target relations required in each "on-line"iteration were re-calculated by the expansion procedure after selecting each experiment and given as input to the next design iteration. Note that the additional information coming from expansion at each step is not utilized by MEED, which designs the whole set of experiments at once, solely based on the prior model of the signaling pathway (Fig.3.5).

3.8.3 Future work: solution by integer programming

Recall from section 3.3, that for a given set of candidate experiments E and a given set of regulators R, predicted profiles for the regulators in R can be presented in matrices of predicted responses. Each matrix is defined by one combination of predicted model states for the experiments in E and corresponds to a set of predicted profiles that need to be distinguished. For those matrices, the requirement for experiments in section 3.3.1 can be formulated in terms of selecting the smallest-size common subset of columns (i.e., experiments). Consider matrices with columns restricted to the selected subset. Each such matrix is required to have all rows pairwise different (assuring that the predicted profiles for each combination of predicted model states are different).

Simple ED problem For a simple case, when there is a unique model state predicted for each experiment in E, there is only one matrix of predicted responses. Denote this matrix $\mathbf{P}$. We now formally define the requirement in this simple case.

Problem 3. SIMPLE ED(P)
INSTANCE: An integer matrix $\mathbf{P}$ with columns indexed by the elements of a set E and rows with the elements of a set R.

QUESTION: Find a minimal subset $E^* \subseteq E$ such that in a matrix with the columns restricted to E^* each pair of rows is different.

Let M be an $m \times n$ matrix, where $m = \binom{|R|}{2}$ and $n = |E|$. Denote the i-th row of M by M_i $(1 \leq i \leq m)$. The rows of M are the absolute values of the differences of rows of the matrix $\mathbf{P}$. That is, for each pair $\mathbf{P}_k, \mathbf{P}_l$ of rows of $\mathbf{P}$ $((k,l) \in R \times R)$, there exists a row M_i in M such that $M_i = |\mathbf{P}_k - \mathbf{P}_l|$. Let $x \in R^n$. A straightforward integer linear programming (ILP) formulation of the SIMPLE ED problem in a canonical form [102] is defined by:

$$\begin{aligned} &\min \textstyle\sum_e x_e, \\ &Mx \geq 1, \\ &x \geq 0, \\ &x \text{ integer} \end{aligned} \tag{3.9}$$

We are looking for solutions where x is a binary vector with entries x_e equal to 1 if and only if $e \in E^*$ $(1 \leq e \leq n)$. The inequalities (3.9) assure that each pair of rows of the matrix $\mathbf{P}$ differs by at least one entry. Such formulated ILP problem can be solved by the branch and bound strategy [93].

Known problems similar to SIMPLE ED The SIMPLE ED problem resembles a general group of *test set* problems, defined by Berman *et al.* [12]. One such problem is the classical MINIMUM TEST SET described by Garey and Johnson [38]. Informally, the test set problems deal with an universe of objects, a group of subsets ("tests") of the universe and a notion of "distinguishability" of pairs of objects of the universe by a set of the tests. The goal is to select a minimum size subset of the tests, which distinguishes every pair of elements of the universe. Note that the instance of a test set problem can be represented as a matrix, with rows corresponding to the universe and columns to the tests. The entries of this matrix denote inclusion of objects in the tests. Unlike the matrix $\mathbf{P}$ given as input to our problem, this matrix is binary. Berman *et al.* [12] review the test set problems which arise in Bioinformatics. For example, Karp *et al.* [65] challenge experimental design (with an experimental design objective that is different than ours), and formulate a subproblem, which we here describe in matrix terms:

Problem 4. CONDITION COVER$(\mathbf{P}, c)$
INSTANCE: A binary matrix $\mathbf{P}$ with columns annotated by the elements of a set E and rows with the elements of a set R, and a cost function $c : E \to \mathcal{N}$, assigning a natural number or 0 to each column.
QUESTION: Find a minimal cost subset $E^* \subseteq E$, such that in a matrix with the

columns restricted to E^*, for each pair of rows, there is (i) at least one entry for which they disagree and (ii) at least one entry for which they agree.

The problem of Karp *et al.* [65] is an extension of ILP for the SIMPLE ED problem, with two inequalities per each pair of rows: inequality identical to (3.9) for the requirement (i) and an additional inequality for requirement (ii). To solve the problem they use the branch and bound approach.

In addition to the problems reviewed by Berman *et al.* [12], Klau *et al.* [71, 72] deal with a problem of probe selection. They first consider an easier version of the problem, which we refer to as SIMPLE PROBE SELECTION and which also belongs to the class of the test set problems. We formulate this problem abstracting from its biological meaning:

Problem 5. SIMPLE PROBE SELECTION$(\mathbf{P}, c, h)$
INSTANCE: A binary matrix $\mathbf{P}$ with columns annotated by the elements of a set E and rows with the elements of a set R, parameters c and h.
QUESTION: Find a minimal subset $E^* \subseteq E$, such that in a matrix with the columns restricted to E^*, (i) for each pair of rows there are at least d entries for which they disagree, and (ii) each row has at least c entries equal to 1.

Here, we do not compare to the full PROBE SELECTION problem, which not only requires that each pair of rows is different on d entries, but also that pairs of small groups of rows are different.

The ILP of Klau *et al.* [71, 72] minimizes the same objective function as our program and includes two sets of inequalities. To satisfy requirement (i), a set of inequalities for all pairs of rows of $\mathbf{P}$ is similar to (3.9) and given by:

$$Mx \geq \mathbf{d},$$

where M is the same as defined for our ILP, and $d \in R^m$ is a vector with all entries equal to the distance parameter $\mathbf{d}$. For requirement (ii), a set of R additional inequalities is added. Klau *et al.* [71, 72] solve the full PROBE SELECTION problem using a branch-and-cut strategy [132], which can shortly be described as a branch and bound with dynamic adding of violated inequalities.

Future work: the full ED problem In the general case, there are several matrices of predicted responses. The matrices have different entries but the same row and column annotations (the elements of a set E for rows and the elements of a set R for columns). The ED problem is then to find the smallest-size common subset of columns, which solves the SIMPLE ED problem for each matrix. The number of matrices is pessimistically exponential in the number of experiments. Due to this major obstacle we do not discuss integer programming-based solution to this problem, leaving it as an open challenge.

3.9 Experimental design validated on synthetic data

To assess the performance of our algorithm, we first compare it with alternative ED methods in four tests on 1000 synthetic inputs each (Fig 3.6).

Synthetic input To define the input for the ED methods in each test, we generate a random model, a repertoire of regulation functions, a set of regulators, and a set of candidate experiments on the model (section 3.3):

- *Random model* Construction of a model requires a definition of its structure and regulation functions. First, to assure that the topologies of the randomly generated models have realistic biological properties, we obtain them based on the graph of the canonical human Tumor Necrosis Factor (TNF) pathway [45]. Each topology is generated by a hundred edge-switching operations on the TNF pathway graph in such a way that the nodes preserve their degrees [6, 88]). Each resulting model graph has one stimulator node, which corresponds to the node in the TNF pathway that does not have incoming edges. All variables are assumed to have three possible states. Next, for each variable in the randomized model, its regulatory function is drawn at random from a uniformly distributed set of possible logical functions. We use only functions whose number of inputs exactly equals to the in-degree of the variable's node. The logical functions were restricted to ensure that the output is dependent on each input.
- *A repertoire of regulation functions* The repertoire of regulation functions for the regulatory programs contains only the *activation both* function a_B (Eq.3.3).
- *A set of regulators* We assume that all variables are regulators, except for the stimulator variables.
- *A set of candidate experiments on the model* For each generated model we take all the possible experiments as the set of candidates. Given that a model generated based on the TNF pathway graph contains nineteen nodes, has one stimulator, and that perturbation can be done on any regulator, but at most one at a time, there are in total 165 possible experiments on this model (section 3.2).

Evaluation of the proposed experiments The experiments proposed by the analyzed methods were evaluated with respect to their efficiency in distinguishing between regulatory programs using the FUP score (section 3.3). Preferably, a given ED method not only proposes an experiment list containing a small number of experiments with the minimal FUP score, but also obtains low FUP scores for each number of only highest priority experiments from this list. This overall performance, referred to as cumulative FUP, is evaluated as the sum of FUP scores over all groups of the highest priority experiments.

Performance of MEED Fig 3.6 A, B presents the FUP score averaged over all random models, obtained by experiments suggested by MEED and the alternative ED

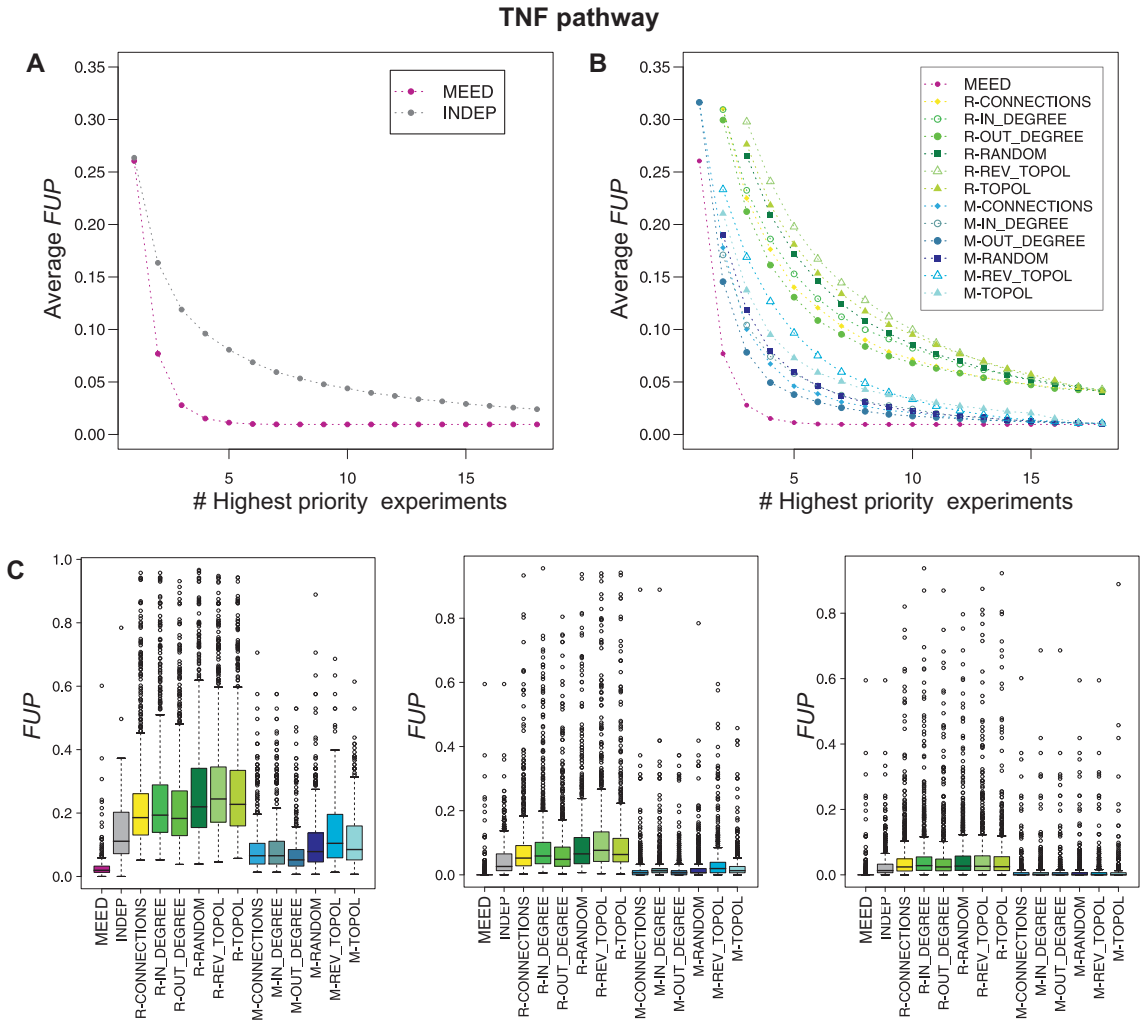


Figure 3.6: Comparative performance analysis on random models. The comparison is carried out on 1000 cyclic models generated by random reshuffling of the TNF canonical human signaling pathway. (**A**, **B**) x-axis: the number of highest priority experiments used from the compared experiment lists to distinguish between regulatory programs, y-axis: the FUP score averaged over the 1000 random models (only the results with average $FUP < 0.35$ are reported). The lower the averaged cumulative FUP, the higher the performance of a given ED method. (**A**) Comparison with the INDEP method. Our MEED algorithm has significant advantage over independent experiment scoring. (**B**) Comparison with the network-based methods. The network-based methods choose the perturbed variables according to key features of the structure, whereas stimulations and perturbation states are chosen either at random (the random methods, R-prefixed, green shaded) or following our MEED algorithm (the hybrid methods, M-prefixed, blue shaded). (**C**) Box plots of the FUP scores (y-axis) for groups of 3, 9 and 15 highest priority experiments from the experiment lists proposed by all analyzed methods (x-axis). The results show that MEED consistently outperforms other methods on the tested random models. In general, the hybrid methods have a better performance than the random methods. This evident tendency implies that even allowing MEED to decide only on stimulations and perturbation states, regardless the way the perturbed variables were chosen, can still provide significant improvement.

methods. MEED proposes only a few experiments on average and obtains the lowest average cumulative *FUP* (the area under curve). Next, we investigate the distribution of *FUP* scores over the 1000 synthetic inputs (Fig 3.6 C). With increasing number of highest priority experiments used from the list proposed by MEED to distinguish regulatory programs, its *FUP* variance quickly declines and becomes negligible.

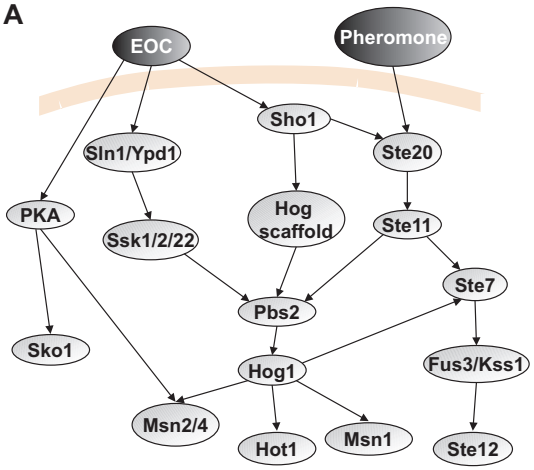
Advantage of MEED over INDEP The first compared method, called INDEP (section 3.8), applies the same measure as MEED, but the score is assigned to each experiment independently, ignoring potential dependencies between experiments. In contrast to INDEP, each consecutive experiment designed by MEED radically increases the number of distinguished regulatory program pairs. With this ability, MEED significantly outperforms INDEP, showing the importance of scoring a set of experiments together rather than each experiment independently (Fig 3.6 A, C).

Advantage of MEED over network-based methods Next, MEED is compared with network-based ED methods, which choose the perturbed variables according to key topological features of the model structure (section 3.8). These methods are divided into random (prefix R) and hybrid (prefix M), according to how they determine stimulation and perturbation states for a predefined perturbed variable: either at random, or with the use of reasoning and scoring of our MEED algorithm, respectively. Fig 3.6 B, C shows the advantage of our algorithm over all network-based methods, indicating that MEED reduces the amount of experimental effort required to distinguish between regulatory programs. Notably, the hybrid methods perform better than the random methods (e.g., both the hybrid method M-TOPOL and the random method R-TOPOL prioritize the variables to be perturbed based on topological order, but M-TOPOL has better performance). Hence, even having predetermined specific molecules to be perturbed, the experimenter can still gain from consulting MEED regarding the type of perturbation and the level of stimulation.

3.10 The MEED framework applied to a yeast signaling model

A yeast signaling model In this section we utilize our framework for the investigation of the yeast cellular response to hyperosmotic and pheromone triggers. The response is mediated by signaling cascades that involve the PKA pathway, as well as the HOG and mating/pseudohyphal growth pathways. The model of the system (based on Gat-Viks and Shamir [41]) is referred to as the *yeast signaling model* or, in short, the *yeast model*. Fig 3.7 shows the model graph and Appendix Fig. .1 presents its regulation functions. The model contains two stimulators: environmental osmotic concentration (EOC) and pheromone. In this study, we focus on the regulation of the immediate response, exploring only the system state before the potential feedback

A



B

#	Stimulation	Perturbation
1.	0.5M KCl 40min	—
2.	Control	*sko1*
3.	0.5M KCl 40min	*pbs2*
4.	50aF 30min	*fus3D*
5.	Control	*sln1*
6.	0.5M KCl 40min	*hog1*
7.	0.5M KCl 40min	*ssk1*
8.	0.5M KCl 40min	*sho1*
9.	Control	*STE12*
10.	0.5M KCl 40min	*ste11*
11.	Control	*MSN2*

Figure 3.7: Experiment list proposed by MEED for the yeast signaling model. The model is depicted on the left as a network with nodes (ovals) corresponding to environmental conditions (dark gray) and signaling components (light gray). Arrows represent signaling relations. The list of experiments designed by MEED is given in a table on the right, listing stimulation (Control – YPD) and perturbation (green: knock-out and red: over-expression).

mechanisms affect the signaling pathway. Therefore, the model does not contain several possible mechanisms of feedback control (e.g., Hog1 protein phosphatases whose production is stimulated after the osmotic shock, or glycerol production that leads to restoration of turgor pressure and stops further activation of the HOG pathway [52]) and we utilize only measurements that were made shortly after stimulation.

Regulators and regulatory functions We consider only transcriptional control by single regulators. With this restriction, there are 27 (3^3) possible one-argument regulation functions reflecting different means of regulation. To avoid the problem of overfitting (described by Gat-Viks and Shamir [41]), we limit ourselves to six biologically relevant regulation functions (section 3.3). We take all variables (apart from the Hog-scaffold variable; altogether fifteen variables) as the set of regulators. In total, we consider 90 regulatory programs (six for each regulator).

3.10.1 Experimental design on the yeast model

Candidate experiments To have access to experimental data for expansion, we restricted all analyzed ED methods to choose only from candidate experiments that are available in microarray databases. Our candidate set of experiments consists of 25 genome-wide profiles that are reported in five publications [104, 47, 92, 97, 21], and listed in Appendix Fig. .2.

Selected experiments For the yeast model, MEED proposes a list of 11 out of 25 candidate experiments (Fig 3.7). Fig 3.8 A and B shows that, similar to the results obtained for random pathways, MEED distinguished regulatory programs more efficiently than INDEP and the network-based methods (section 3.8). For the yeast model, M-TOPOL performs best from the network-based approaches. The set of all 25 candidate experiments (therefore, also the experiments selected by MEED) cannot distinguish between pairs of regulatory programs within five groups. Each of the five groups contains three regulatory programs, with the same three regulators: Hog1, Msn1, and Hot1. The regulation functions are the same for all regulatory programs within each group: *necessary activation* in the first group, *activation both* in the next, as well as *sufficient inhibition*, *sufficient activation*, and *necessary inhibition* in the remaining groups. Accordingly, adding more experiments from this candidate set to the experiment list designed by MEED does not enable to distinguish between more regulatory programs.

3.10.2 Expansion of the yeast signaling model

To test our framework in practice, we performed expansion of the yeast model using the measurements from the 11 experiments chosen by MEED. In the expansion procedure, genes were assigned to regulatory modules by a probabilistic matching of the observed profiles of the genes to the predicted profiles of the regulatory programs (section 3.7).

Expansion using experiments suggested by alternative ED approaches For comparison, we repeated the expansion procedure using experiments selected by independent experiment scoring (INDEP), the best-performing network-based method (M-TOPOL; Fig. 3.8 B), as well as two extant ED methods, introduced by Ideker *et al.* (2000) and by Barrett and Palsson (2006) (section 3.8). Unlike MEED, the two extant methods take as input high-throughput measurements (gene expression or binding data) to build initial pathway models, and apply an "on-line" procedure, that is, they use the data from each chosen experiment to propose the next one. Both the initial and "on-line" data come from expansion. INDEP, MEED and M-TOPOL were applied to choose from the same set of 25 candidate experiments. The methods of Ideker *et al.* (2000) and Barrett and Palsson (2006) choose only from 21 candidate experiments, since four are used to provide data for their initialization (see section 3.8).

Advantage over the alternative ED approaches For the yeast model, MEED achieves better performance than the extant methods in distinguishing regulatory programs (measured with *FUP* score, see section 3.3; Fig 3.8 A). The method of Ideker *et al.* (2000) reaches its stop criterion already after choosing three experiments.

As reported in Fig 3.8 C, using the 11 experiments proposed by MEED, the expansion procedure identifies 26 regulatory modules controlled by the yeast signaling pathway.

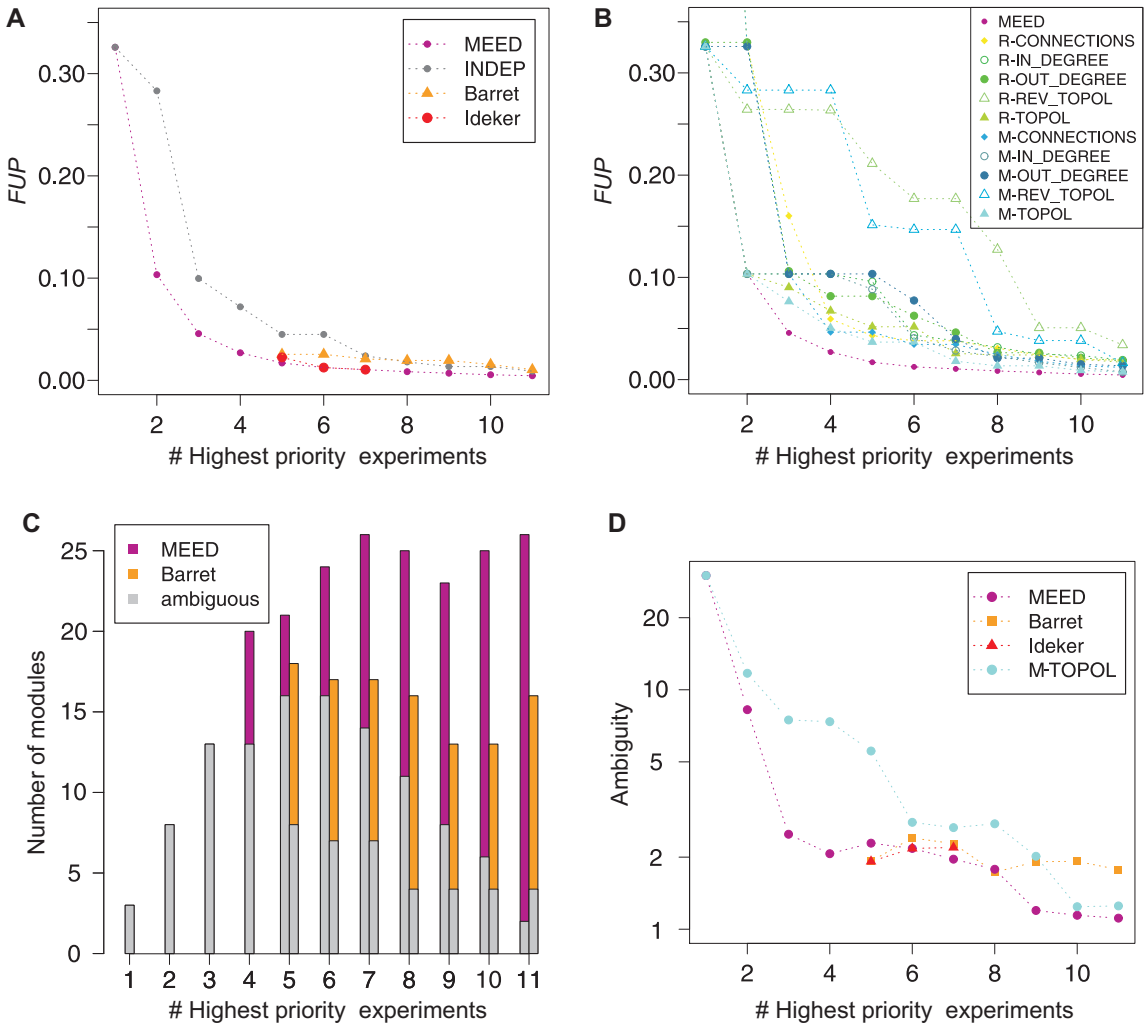


Figure 3.8: Comparative performance on the yeast signaling model: FUP scores and ambiguity of expansion. MEED (plotted in magenta) is compared to INDEP (gray), network-based methods, as well as two extant ED approaches (Barret and Pallson (2006) - orange; Ideker et al. (2000) - red). Since the two extant methods take as input results of expansion using the first four experiments proposed by MEED, their report starts from the fifth experiment. The method of Ideker et al. (2000) reaches its stop criterion already after choosing three experiments (fifth to seventh experiment). x-axis in all plots (**A**–**D**): the number of highest priority experiments. For comparison with MEED, we present up to eleven experiments chosen by the other methods. (**A**,**B**) FUP scores. y-axis: the FUP score measuring the ability of the experiments to distinguish between regulatory programs (only the results for $FUP < 0.35$ are reported). With the lowest FUP for every number of highest priority experiments, MEED outperforms all alternative methods. The best performing of the network-based methods is M-TOPOL. (**C**) Regulatory modules. y-axis: the number of modules identified in expansion. The proportion of ambiguous modules is marked in gray. In comparison with the method of Barret and Pallson (2006), more modules are obtained using the same number of highest priority experiments proposed by MEED (the results for the method of Ideker et al. (2000) are similar to the results of MEED and are not plotted for clarity). (**D**) Ambiguity of expansion. y-axis: ambiguity score (i.e., the average number of regulatory programs per gene; plotted in log scale). With lower ambiguity score for most numbers of highest priority experiments, MEED outperforms M-TOPOL and the method of Barret and Pallson (2006) on the yeast model.

More regulatory modules are identified using any number of highest priority experiments proposed by MEED than the same number of experiments proposed by the method of Barrett and Palsson (2006). Moreover, the eleven experiments chosen by MEED enable lower percentage (2 out 26) of ambiguous modules (modules that were matched to more than one regulatory program). The method of Ideker *et al.* (2000) achieves similar results as MEED (not shown).

The quality of expansion is further evaluated by the ambiguity score (section 3.7), reporting the average number of regulatory programs that were identified for each gene. Unlike the FUP score, which evaluates a given ED method based only on model predictions, the ambiguity score evaluates results of expansion, which utilizes experimental data. Fig 3.8 D indicates that MEED outperforms M-TOPOL and (except when the six highest priority experiments are used) the extant methods with respect to ambiguity scores. Taken together, the presented results indicate practical applicability as a strong advantage of MEED, which performs comparably or better than the extant approaches although it does not require the data from each chosen experiment to propose the next one (section 3.8).

Model specificity of MEED In Fig. 3.9, we use the ambiguity score to show the specificity of the set of experiments chosen by MEED for the particular yeast model. To this end, we compare the ambiguity of the regulatory modules obtained in expansion of the original yeast model with the expansion of its randomized version using the same experiment list proposed by MEED for the original model. Structure and regulation functions of the randomized model were obtained as in section 3.9. The nodes of the randomized model are the same as in the original model and the set of regulatory programs is also identical. Expansion of the original model results in strikingly less ambiguous regulatory modules as compared to the modules identified for the randomized model. This result indicates high specificity of our algorithm in choosing experiments for a particular model. The regulatory modules identified for the original model obtain better ambiguity scores than the regulatory modules found for the randomized model. This tendency can be explained by the fact that the randomized model does not represent the true signaling pathway. For all experiments, the randomized model predicts states of regulators in the pathway that are not "compatible" with the actual measured gene response. As a consequence, there is a poor match between the model-dependent predicted profiles and the real observed profiles, resulting in ambiguous regulatory modules. Moreover, the experiments proposed by MEED for the original model cannot be expected to distinguish the regulatory programs in the randomized model.

Stability of model expansion Next, we validate the expansion of the yeast pathway by conducting expansion with additional experiments on top of the eleven experiments suggested by MEED. In this way, we test the stability of gene assignment, that is, whether with more experiments there is a dramatic rearrangement of genes between regulatory modules, or whether the genes are added to or removed from the modules.

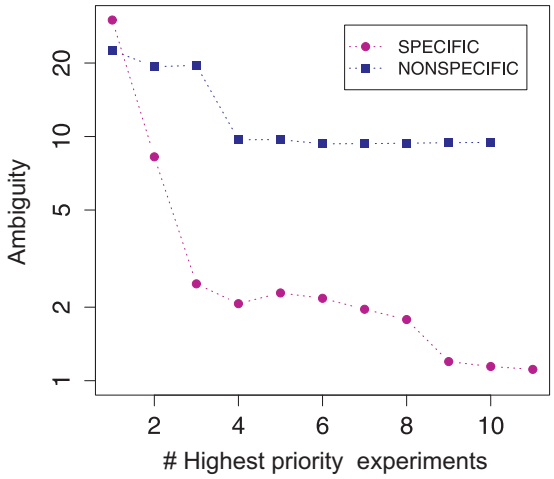


Figure 3.9: Model-specificity of MEED. Ambiguity of the regulatory modules obtained in expansion of the original yeast model (SPECIFIC, magenta circles) and its randomized version (NONSPECIFIC, blue squares,) using the same experiment list proposed by MEED for the original model. x-axis: the number of highest priority experiments from the list used in the expansion procedure. y-axis: ambiguity score, measuring the average number of regulatory programs predicted for each gene (plotted in log scale).

Model expansion using experiments designed by MEED is guaranteed to be correct and perfectly stable under several ideal assumptions. Thus, changes in gene assignment upon adding new experiments in expansion would indicate a violation of these assumptions. First, MEED needs to be given as input all biologically relevant regulatory programs. Otherwise, it is possible that the given programs will not be distinguished from the ones that were left out. In this case, genes regulated by a program that was left out could erroneously be assigned to one of the identified modules. Using a proper additional experiment in expansion would cause those genes to be removed from the false module. Second, the additional experiments may reveal mistakes in the model or low quality of the measurements.

We tested the stability of gene assignment by applying expansion procedure to the yeast pathway model using increasing experiment lists up to all 25 experiments from the candidate set. The first eleven experiments were those proposed by MEED, and were added to the list in the order they were chosen by the algorithm. This test was repeated ten times; each time the remaining fourteen experiments were added to the list in a different random order. Next, the results were averaged over the ten random orders. Fig. 3.10 A shows the total number of genes assigned to modules across different numbers of utilized experiments. The initial five highest priority experiments filter out majority of genes. After the 11 experiments proposed using MEED, using additional ones in expansion only slightly decreases the total number of assigned genes. A large fraction of those genes, which are assigned using the experiments proposed by MEED and remain assigned using extended experiment lists, is assigned to the same regulatory modules (Fig. 3.10 B). Therefore, there is

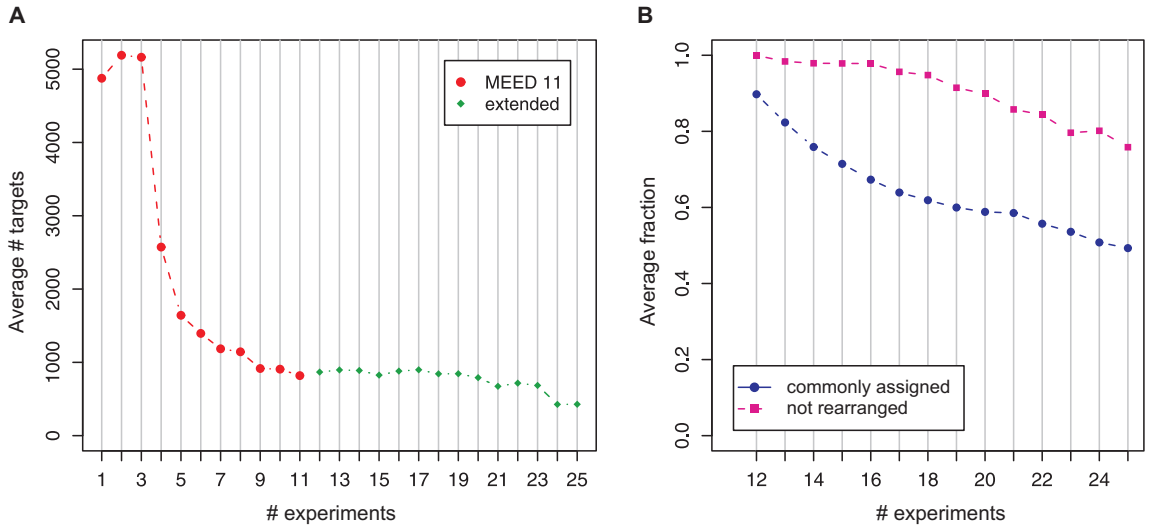


Figure 3.10: Stability of assignment of target genes to regulatory modules. (**A**) Number of target genes assigned to any regulatory module (red: using experiment lists up to the eleven experiments chosen by MEED; green: using extended experiment lists, average over ten random orders of adding extra experiments). x-axis: the number of experiments used in the expansion procedure. y-axis: the number of identified target genes. Using only a few experiments, expansion procedure identifies a large fraction of genes as regulatory targets. The number of assigned target genes decreases for the chosen eleven experiments, and stays in the same order of magnitude, but not on an equal level, when additional experiments are added. Therefore the numbers of assigned targets do not change drastically once reasonable length (more than just a few) experiment lists are used. (**B**) Rearrangement of genes upon addition of experiments. We define *commonly assigned* genes as all genes assigned to regulatory modules both with the extended experiment list and with eleven experiments suggested by MEED. x-axis: the number of experiments used in the expansion procedure. y-axis, blue: the fraction of commonly assigned genes out of all genes assigned to any regulatory module with the extended experiment list (average over the ten random orders). The fraction of commonly assigned genes goes down when more experiments are used, which indicates that some new genes are added to the modules and some genes are removed. y-axis, violet: out of the commonly assigned genes, the fraction of genes that are assigned to the same regulatory module both with eleven and with more experiments (average over the ten random orders, referred to as *not rearranged*). Remarkably, a large fraction of commonly assigned genes is not rearranged between the modules when adding additional experiments.

only little rearrangement between the modules when more experiments are used.

3.10.3 Regulatory modules in the yeast signaling model

To assess the biological findings resulting from application of our framework to the yeast signaling model, we focused further analysis on the obtained regulatory modules. As small modules could have been generated at random, given the large number

of potential regulatory programs, we restricted the analysis to fourteen modules containing at least seven genes. Fig.3.11 presents a map of the expansion, including the identified regulatory modules, their regulatory programs, predicted profiles and the expression matrices of the target genes. The map clearly shows high agreement between predicted profiles and observed profiles. Cases of disagreement (e.g., observed and predicted responses to the second experiment, *sko1* mutant, in two regulatory modules, inhibited by Kss1/Fus3 or Ste12, respectively) show faults in our understanding and incompleteness of the yeast signaling pathway model.

The identified versus known regulatory programs The expansion analysis provides hypotheses regarding the regulatory relations and their mechanisms downstream of the yeast signaling model. Tab.3.2 summarizes a detailed comparison between the identified regulatory programs to known programs based on a comprehensive review [52]. The known regulatory relations include four cases of transcriptional control by a single regulator and four combinatorial regulations (not considered in this study). All four single-regulator programs were detected by our analysis (activation by Msn2/4, activation by Ste12, inhibition by Sko1 and activation by Hot1–here ambiguous with Msn1 and Hog1), confirming the quality of our predictions. In a number of cases, well-characterized target genes were identified by the expansion analysis, thereby serving as positive controls. For example, our analysis indicates that CTT1 and HSP12 are activated by Msn2/4, and FUS1, FUS3 and FIG1 are activated by Ste12, both consistent with the known transcriptional control of these target genes. In total, out of sixteen target genes, known to be regulated by a single TF, eight genes have been assigned correctly and no gene has been assigned to a wrong regulatory module. As combinatorial regulation was not taken into consideration in our analysis, we expect that target genes with more than one known regulator will not be assigned to any of the regulatory modules. Indeed, all six combinatorially regulated target genes did not match any of the regulatory programs.

Indirect gene regulation by kinases in the modeled yeast pathway Interestingly, four kinases, including Kss1/Fus3, PKA, Sho1 and Ste7, were identified as gene regulators (Fig. 3.11). The hypothesized regulation might be explained by an indirect influence on the target genes through alternative signaling pathways and downstream TFs that are not part of the model. Several such alternative pathways are known but were omitted from the model. For example, PKA regulates transcription through the TFs Msn2/4 and Sko1 (part of the model) or through Adr1, Rap1 and Crz1 [52, 138] (not modeled), Kss1/Fus3 mediates transcription through the Far1 kinase independently of Ste12 [94], and the Sln1/Ypd1 kinases regulate an alternative hypo-osmotic stress pathway through the TF Skn7 [52] (not modeled). There is no known alternative pathway downstream the signaling molecules Sho1 and Ste7. Our results suggest that these signaling molecules have an indirect effect on gene expression through an additional pathway, independent of the model.

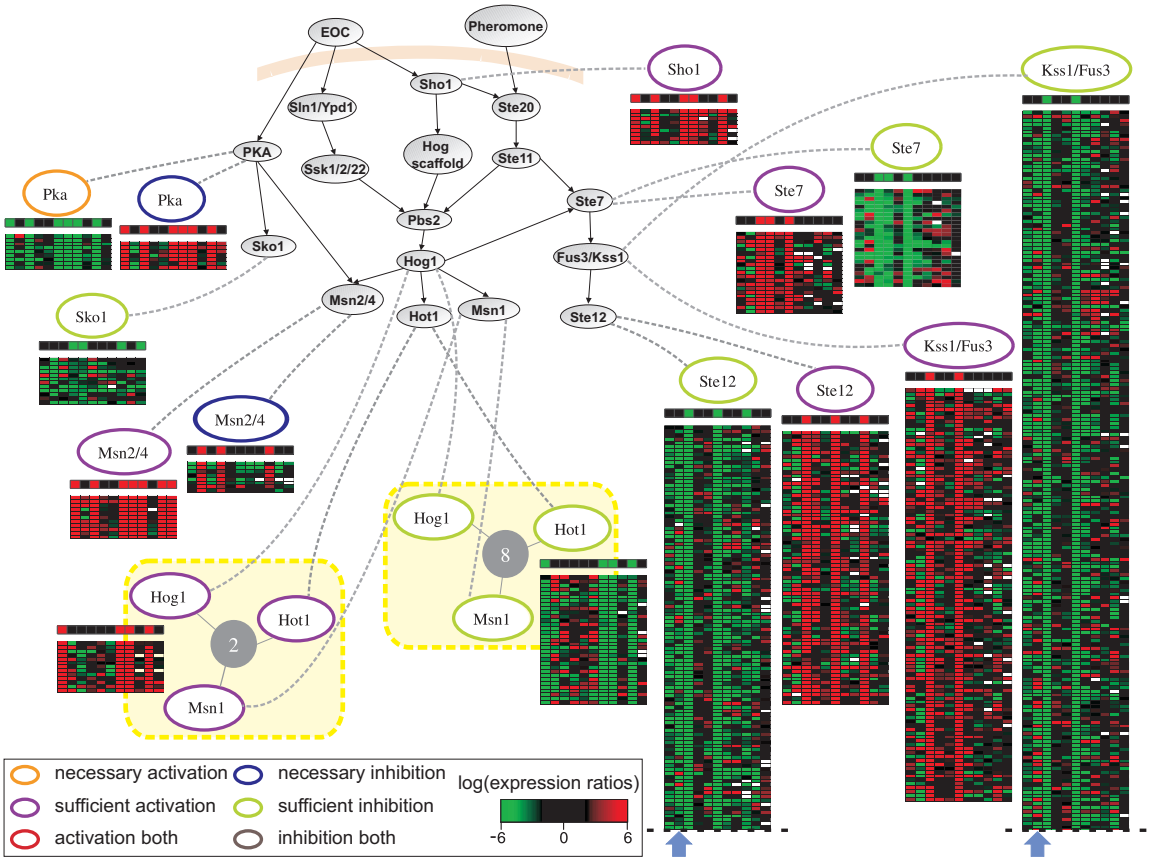


Figure 3.11: Expansion of the yeast signaling model using the experiments proposed by MEED. The yeast model is depicted in the center of the figure. The identified modules are presented, with additional dashed edges connecting the regulators in the pathway to their regulatory programs (nodes labeled with regulators and having a boundary color-coded according to their regulation function). The ambiguous modules, highlighted with dashed yellow rectangles, are presented as gray-filled nodes, labeled with their size and connected by edges to all their matching regulatory program nodes. The two ambiguous modules were subject to an additional MEED iteration, which succeeds to distinguish their regulatory programs using only two additional experiments. A matrix showing the expression measurements of target genes (rows) across the eleven experiments proposed by MEED (columns) is presented only for the modules that contain at least seven genes. The columns of the expression matrices are ordered from left to right according to the order proposed by MEED. For clarity, only subsets of the large Ste12 and Kss1/Fus3 matrices are shown. The predicted profiles appear as a separate row above the matrix. For most modules, the expression profiles agree well with the predicted profiles. Blue arrows exemplify experiments where all module genes jointly disagree with the predicted profile.

Type	Known regulatory program	Predicted regulatory program	Known gene target	Predicted target
single regulator	activation by Hot1	correct	CHA1	not predicted
			PHO84	not predicted
			YGR043C	correct**
			YGR052W	not predicted
			YHR087W	not predicted
	activation by Msn2/4	correct	CTT1	correct
			HSP12	correct
			HSP26	not predicted
			DCS2	correct
			GCY1	not predicted
	inhibition by Sko1	correct	AHP1	not predicted
			HAL1	not predicted
	activation by Ste12 (without Tec1)	correct	FUS3	correct
			FUS1	correct
			FIG1	correct
			YPL192C	correct
combinatorial regulation	Sko1+Crz1*		ENA1	not predicted
	activation by Msn1+Hot1		STL1	not predicted
	activation by Msn2/4+Msn1+Hot1		GPD1	not predicted
			GPP2/HOR2	not predicted
	inhibition by Sko1 together with Hog1		GRE2	not predicted
			GLR1	not predicted

Table 3.2: Identified versus known regulatory programs. The table summarizes the programs reviewed by Hohmann [52], reporting whether they were rediscovered by our MEED framework. **Type** – states whether the regulation is combinatorial or not. **Known regulatory program** – the known regulator(s) and the regulatory mechanism, **Predicted regulatory program** – states whether the known regulatory program was correctly identified (does not refer to combinatorial regulation, which was not a subject of our analysis), **Known gene target** – known target genes of the regulatory program, **Predicted target** – states whether the known target gene was correctly included in the regulatory module assigned to the regulatory program (genes marked as not predicted were not assigned to any module). *Crz1 is downstream to the PKA variable, therefore represents indirect regulation of PKA independently of Sko1. **assigned to an ambiguous module Hot1-Msn1-Hog1.

Biological evaluation of the regulatory modules We evaluated all fourteen modules to test whether the proteins encoded by the target genes had a related function or a shared transcriptional regulation. To that end, we scored each module according to its enrichment in GO annotations (using the Ontologizer tool designed by Bauer *et al.*, 2008 [11]) and sets of transcription targets identified by protein-DNA binding experiments [49, 103, 139] (computed using a hypergeometric test). Out of the four large modules (containing at least 100 target genes), three modules obtained enrichments below a p-value threshold 0.001 (Bonferroni corrected; Fig.3.12). All other modules did not obtain significant enrichment, probably because of their small size (each of these modules contains less than 26 genes, including genes that were not annotated yet).

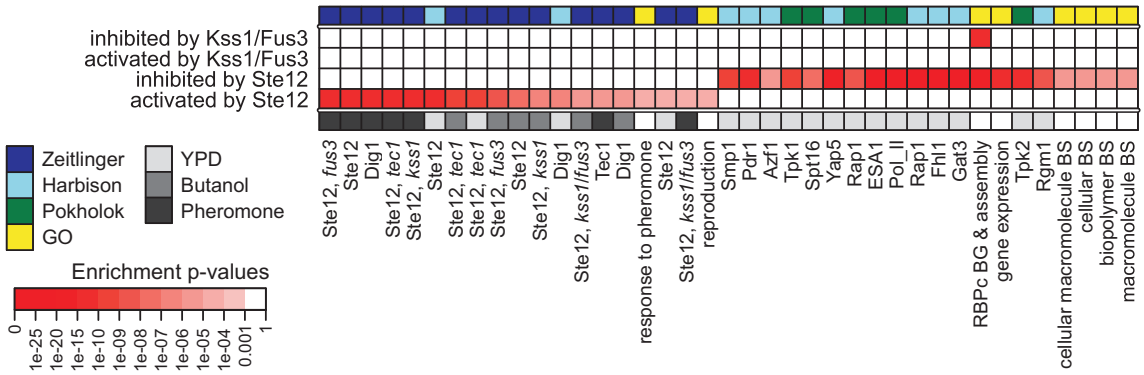


Figure 3.12: Functional coherence of identified regulatory modules. Enrichment of the target genes from each of four large identified modules (rows) in various experiments (columns). Significant enrichment (Bonferroni-corrected hyper-geometric p-value; indicated by shades of red) represents distinct behavior of the genes in a module compared to the rest of the genome. Enrichment p-values in TF-DNA binding targets [49, 103, 139] and gene ontology annotation (GO [5]) are reported. The different data sets and experiments' environmental conditions are color-coded above and below the matrix, respectively. The profiles used for the enrichment tests were not part of our original dataset. RPBc – Ribonucleoprotein complex, BG – biogenesis, BS – biosynthesis.

The enrichment analysis supports and provides insights into the identified modules. For example, it justifies the division of the genes downstream of the mating pathway into two activation modules: a module activated by the transcription factor Ste12 and a module activated by the kinases Kss1/Fus3. According to our enrichment analysis, the genes activated by Ste12 are characterized by several annotations, which are all related to the known functionality of Ste12 as a key TF of the mating pathway (Fig.3.12). However, the Kss1/Fus3 targets are not enriched in any of these annotations, confirming that Ste12 does not control those targets. To provide additional evidence that these two transcriptional modules are distinct, we performed promoter sequence analysis using the Amadeus tool [82]. The known binding motif of Ste12 was highly enriched in the module under sufficient activation by Ste12 (p-value $< 10^{-12}$), whereas the module under sufficient activation by Kss1/Fus3 was not enriched in this motif. Taken together, our analysis provides evidence for transcriptional regulation by Kss1/Fus3, independently of Ste12 control.

We next asked what is the regulatory pathway mediating the regulatory program of *sufficient activation* by Kss1/Fus3 on its gene targets. Kss1 and Fus3 have no preferential binding to the promoters of the Kss1/Fus3 module [103] (data not shown), ruling out the possibility that Kss1/Fus3 have a direct effect on their targets. One potential indirect transcriptional control by Kss1/Fus3 is mediated through the kinase Far1, which mediates cell-cycle arrest in response to pheromone, independently of Ste12. However, our module is not enriched in cell-cycle annotations (Fig.3.12), indicating that Far1 is unlikely to mediate the observed gene activation downstream of Kss1/Fus3. As more experimental investigations of the pathway connectivity become

available, the mechanisms by which Kss1/Fus3 control its targets should be further revealed.

3.10.4 Ambiguity networks and iterative experimental design

Ambiguity network To facilitate the inspection of ambiguous modules in a given expanded model, we devised the concept of an *ambiguity network*. Recall that an unambiguous module matches exactly one regulatory program, and an ambiguous module matches strictly more than one program (section 3.7). We define an ambiguity network as a graph whose nodes represent regulatory programs that matched one of the regulatory modules. One additional node is added for each ambiguous module, labeled by the number of genes it contains. There are edges between the ambiguous module nodes and their matching regulatory program nodes. In this way, the ambiguity network highlights the ambiguous modules and provides details on their size and the alternative regulation hypotheses.

Ambiguity network versus ambiguity score Fig.3.13 compares two ambiguity networks for two sets of regulatory modules that differ significantly in their ambiguity score. The networks were generated based on the yeast model expansion using two groups of five and six highest priority experiments from the experiment list proposed by M-TOPOL. Adding the sixth experiment (knockout of Pbs2 in high osmotic stress) lowers the ambiguity of the identified regulatory modules (compare Fig 3.8 D). Recall that the ambiguity score is proportional both to the number of regulatory programs matching each ambiguous module and to its size. Therefore, such a strong drop of ambiguity score can be explained by the fact that with the added experiment, the ambiguous modules either: (i) match fewer regulatory programs, or (ii) contain fewer genes. As an example of the former case, using the five highest priority experiments, the expansion procedure identifies one of the ambiguous modules to be controlled by seven regulatory programs. With the sixth experiment added, this module is replaced by two, matching four and three regulatory programs, respectively (Fig.3.13, red rectangles). As an example of the latter case, consider the largest ambiguous module containing 3233 genes in expansion performed using five experiments. With the sixth experiment added, this module is replaced by two smaller modules. These modules match three regulatory programs each and contain only 307 and 677 genes (Fig.3.13, blue rectangles).

Iterative ED in our framework Our framework can be used in iterations of the MEED algorithm and expansion procedure. Experiments chosen by MEED from the restricted set of 25 candidate experiments do not distinguish all regulatory programs in the yeast model. Five groups of regulatory modules remain undistinguished (listed in section 3.10.1). Accordingly, expansion performed using these experiments generates two ambiguous modules (the remaining three groups of regulatory programs are

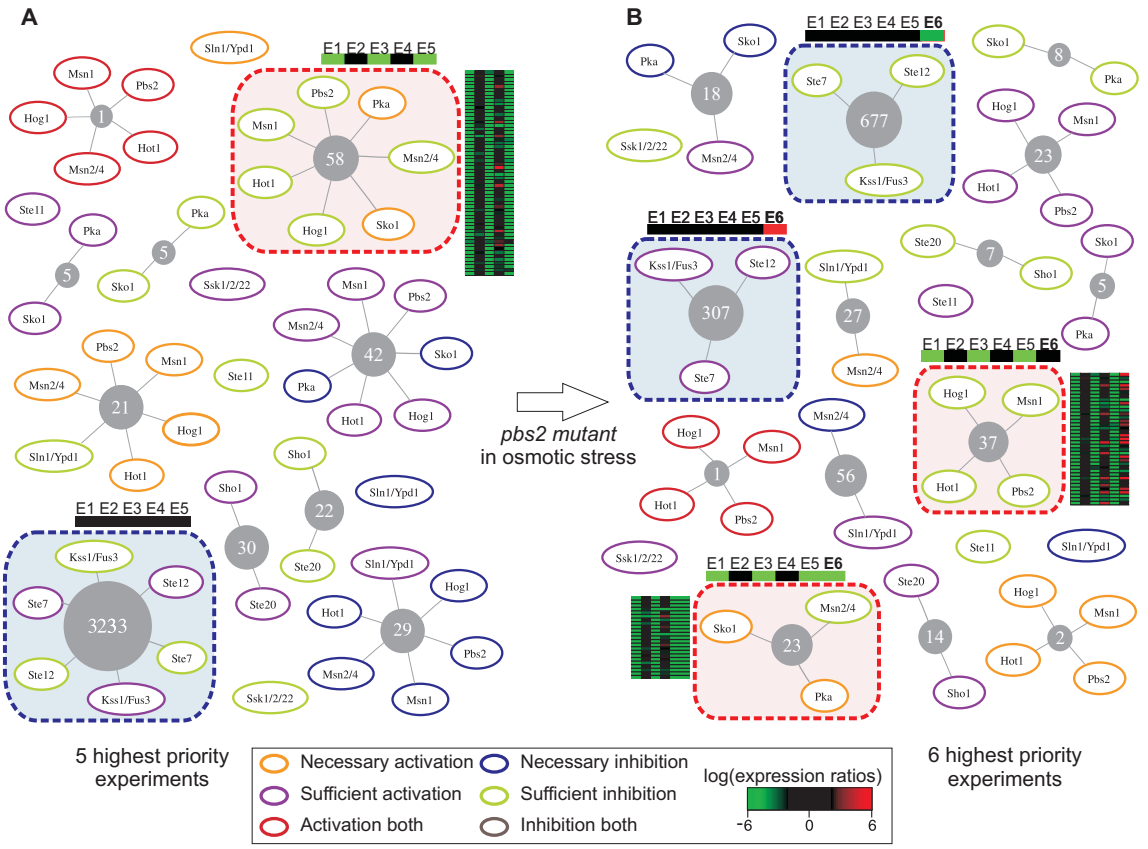


Figure 3.13: Illustrating expansion results with ambiguity networks. Ambiguity networks for regulatory modules obtained in expansion of the yeast model using the first five (**A**) and six (**B**) experiments on the list proposed by M-TOPOL (i.e., **A** and **B** differ only by one additional sixth experiment from the list). The ambiguity network provides a detailed insight into the ambiguous modules. Each white-filled node represents a regulatory program matching one of the identified modules. It is labeled with its regulator, and has a boundary color-coded according to its regulation function. Unambiguous modules are presented only by their unique matching regulatory program, without indicating their size. Ambiguous modules are presented as gray-filled nodes, labeled with their size and connected by edges to all their matching regulatory program nodes. Exemplary modules (highlighted with dashed rectangles) are shown together with their predicted profile (colored vector above the rectangle). Red rectangles: an ambiguous module controlled by seven regulatory programs containing a large set of genes in **A** is replaced in **B** by two smaller ambiguous modules controlled by four and three regulatory programs, respectively. The two modules differ in the gene response to the additional sixth experiment. Matrices showing expression profiles of the target genes (rows) across the experiments (columns) are plotted next to the modules. Blue rectangles: A large ambiguous module whose genes did not respond in any of the first five experiments (the corresponding predicted profile is filled with black in **A**). Using the sixth experiment, the large module is replaced by two smaller ones in **B**. One module contains genes that were down-regulated in the sixth experiment, whereas another contains genes that were up-regulated (can be seen in green vs. red entries in the predicted profiles of the modules). A large group of genes, whose expression has not changed in the sixth experiment, does not match any profile and therefore is not contained in any regulatory module.

not predicted to control any modules). The ambiguous modules match three regulatory programs each (the regulators Hog1, Msn1 and Hot1 as sufficient inhibitors and the same regulators as sufficient activators, shown in (Fig.3.11)).

3.11 Discussion

This chapter presents a general framework for discovering regulatory modules downstream of a studied signaling pathway. The framework guides the choice of experiments in research on a particular signaling pathway and investigating the regulation of the pathway's target genes. The pathway is formalized in a logical model. Based on the model's predictions, the MEED algorithm chooses the experiments from a set of candidates. The expansion procedure reconciles the predictions with the data from the suggested experiments to identify regulatory modules downstream of the modeled pathway.

If the candidate experiments distinguish all regulatory programs, using the experiments selected by MEED in expansion will result in a set of unambiguous modules. Ambiguous modules can be obtained in the case when only part of the experiment list suggested by MEED is used in expansion or when the candidate experiments do not distinguish all regulatory programs. In such a case, it is possible to analyze the ambiguity network and specify ambiguous modules that are subject to additional MEED iterations (section 3.10.4). This follows the widely accepted iterative framework for biological discovery in systems biology [58, 70], with the specific application of experimental design for discovering transcriptional regulation downstream of a given pathway.

MEED does not suggest all experiments necessary for high-confidence assignment of genes to regulatory modules. Rather, it tries to minimize the number of experiments required to distinguish the input list of regulatory programs. Therefore, in practice, model expansion will benefit not only from utilizing extra biological and technical repeats of the suggested experiments, but also from extending the economical list provided by MEED with additional available experiments. First, the new experiments will bring new evidence to refine the assignment of genes to modules. Second, they can be used to validate expansion results. In our study, upon adding experiments beyond the eleven proposed by MEED, the total number of assigned genes remains on the same order of magnitude. Moreover, only a small fraction of the genes is rearranged between the modules (section 3.10.2). This provides strong support for the robustness of the assignment of genes to modules downstream of the yeast signaling network.

Our modeling formalism was chosen to fit the available biological data and knowledge. In contrast to detailed modeling approaches (e.g., ODE modeling), the logical model does not require setting a large amount of parameters, which are unknown for most signaling reactions. Other semiqualitative/qualitative modeling methods, e.g., Boolean networks [44, 66], or qualitative differential equations [28], are dynamic

modeling approaches that require time-course data. Here, unlike these approaches, we assume that the regulatory relations are discrete logical functions and the model describes the steady state of the system, thereby enabling to utilize single time point expression measurements.

In the proposed framework, there is a distinction between the model-based experimental design and data-based expansion procedure: The MEED algorithm selects the experiments independent of the data and relies only on the non-stochastic model predictions of discrete states reflecting responses of putative regulatory targets. The stochastic nature of the data is considered only in the expansion, once the measurements from the experiments proposed by MEED are available. We expect that based on the proposed framework, it will be possible to develop techniques handling stochastic model formalisms, such as a Bayesian network model, which represent the prior belief in the logical functions (as implemented in Gat-Viks and Shamir [41]).

In this contribution, we considered only regulatory programs with single regulators and experiments with perturbations of one molecule. Our approach is general and can be extended to investigate combinatorial control by taking into account regulatory programs with multiple regulators and experiments with more than one perturbed variable. The MEED algorithm, which is linear in the number of regulatory programs, will scale to the enlarged problem, with the condition that only a small selection of a vast number of all combinatorial possibilities is considered. For example, for two regulators, and three possible states of the variables, the number of all possible regulation functions is $3^{(3^2)} = 19683$. Already in the case of single regulator programs we choose six biologically relevant regulation functions (out of 27 possible). Applying the same selection criteria, one could consider only a handful of biologically relevant combinatorial functions (e.g., the combinatorial schemas proposed by Buchler et al., [19] or Yeang and Jaakkola [135]).

Chapter 4

Gene deregulation revealed using perturbation experiments and knowledge

The approach presented in this chapter is designed for quantifying deregulation, i.e., changes of regulatory relations between two cell populations. In chapter 2 we showed how to use known TF targets as examples of differentially expressed genes in the TF perturbation experiments. In chapter 3 we utilized a predictive pathway model for elucidating regulatory relations downstream of the pathway. Our deregulation analysis, introduced in section 4.1 benefits from both these kinds of knowledge as well as perturbation data, collected for the two compared cell populations. The approach was applied to deregulation in response to DNA damage. Section 4.2 presents a clustering of the deregulated genes into functional clusters, reflecting the rich spectrum of biological activities in the DNA damage response program. Section 4.3 investigates the connectivity within the clusters by analyzing enrichment in signaling pathways as well as known gene-regulatory and protein-protein interactions. Section 4.4 reviews the genes with most extreme deregulation scores, reporting their involvement in DNA damage response. Finally, in section 4.5 we determine the indirect regulatory impact of the ATM pathway on the deregulated genes, and in section 4.6 we build a hypothetical hierarchy of direct regulation.

4.1 Quantifying deregulation

Overview The approach presented in this chapter is designed for quantifying *deregulation*, i.e., changes of gene regulatory relations occurring between two given populations of cells. It performs joint analysis of perturbation data from the two cell populations, and is referred to as joint deregulation analysis (JODA) throughout the text. The cell populations may correspond to healthy and diseased cells, or diseased cells in two different stages, or, more generally, cells exposed to two different external stimuli, with different cellular signaling and downstream transcriptional targets. We consider only single gene perturbations that artificially down-regulate the gene, and

refer to them as knockdown experiments (although the approach is equally well applicable to knockout data). Introduction to essential biological notions, e.g., regulatory relation and mechanism, signaling relation and perturbation experiment are given in sections 1.1 and 1.2.

We distinguish two sets of genes: the *regulators*, and all *remaining genes* (shortly, *genes*). The regulators are components of a signaling pathway, which is important for the switch between the cell populations and may have a different topology in one cell population than in the other. We require that each regulator is knocked down in each cell population. The remaining genes show effects of the knockdowns in their expression. We are interested in regulatory relations connecting regulators to the remaining genes and how these relations change between the cell populations.

In addition to the knockdown data, for both cell populations separately, JODA takes as input two kinds of qualitative knowledge, described in detail below: (i) a *pathway topology*, which describes the signaling relations between all regulators within the pathway, and (ii) regulator-target relations, between some regulators, which are also transcription factors, and some remaining genes. The output of JODA are *deregulation scores* that quantify deregulation using the difference of knockdown effects between the two cell populations. An up-regulation effect indicates (possibly indirect) inhibition, and down-regulation indicates activation of the genes by the regulator, which was knocked down. The most extreme deregulation scores are assigned to those genes which switch regulatory mechanism and show different knockdown effects between the cell populations.

Extant approaches Extant deregulation studies combine gene expression data available for two compared cell populations with additional information. For example, known pathways are incorporated into the deregulation analysis to explore changing functionality [15, 37, 54, 83, 128]. On a network level, the switch between cell populations was characterized by deregulated sets of gene interconnections [84, 120, 34, 56]. Analyzing deregulation, these extant approaches do not take into account available knowledge about cellular signaling pathways nor their transcriptional targets, which may differ between the cell populations. For example, Mani *et al.* [84] and Taylor *et al.* [120] take as input a static interactome, which is not specific for the two cell populations, to discover loss or gain of expression correlation between its nodes. Workman *et al.* [134] showed extensive re-wiring of gene regulatory networks in yeast cells undergoing DNA damage using genome-wide measurements of gene expression upon transcription factor (TF) knockdowns, as well as TF binding to DNA. This advanced approach could be further improved by incorporating prior information about the signaling pathways that are differentially activated upstream of the re-wired gene regulatory network, and the complementarity between the TF DNA-binding and the TF knockdown data.

Input pathway topologies The first kind of knowledge are two *pathway topologies*, which describe the signaling relations between all regulators within the pathway in the

two cell populations. The set of the regulators is the same for the two cell populations, but their signaling relations may be different. We assume that researcher's expertise, literature findings or external experimental data provide qualitative knowledge about the signaling relations, describing "who signals to whom" in both cell populations. This knowledge is given to the input of JODA in a form of two directed graphs (one per each cell population). The nodes in the graphs correspond to the regulators (pathway components). There is an edge between two nodes in a given pathway topology whenever it is known that the pathway component corresponding to one node activates the component corresponding to the other node.

We denote the set of regulators as $V = \{v_1, ..., v_n\}$. Formally, a pathway topology in a cell population t is a graph $G_t = (V, A_t)$ with the set of nodes V and directed edges A_t. G_t may be cyclic and may have several connected components. There is an edge $(v_i, v_j) \in A_t$ whenever it is known that the pathway component v_i activates v_j in the cell population t. Examples of two given ATM pathway topologies, one in the healthy cells (denoted h) and second in cells undergoing DNA damage (shortly, *damaged* cells, denoted d), are illustrated in Fig.4.1 A.

Model predictions of knockdown effects In each cell population separately, the known pathway topology can be utilized to predict effects of knockdown experiments. Consider an experiment $\Delta^t v$, where a given regulator v is knocked down in a given cell population t. The regulator v together with all regulators, which are reachable from v in the pathway topology t, are called *affected* by the experiment $\Delta^t v$. The set of all experiments knocking down the regulators in V in cell population t is denoted E_t. The predictions of affected regulators for all knockdowns in E_t are given by the transitive reflexive closure $G_t^* = (V, A_t^*)$ of the pathway topology G_t. To compute G_t^*, we add an edge $(v_i, v_j) \in A_t^*$ whenever there exists a directed path from v_i to v_j in the pathway topology G_t (allowing $v_i = v_j$, i.e., the path may be empty). The incidence matrix for G_t^* is called the *model matrix*, or shortly, the *model*, and denoted $\mathcal{M}_t$. There is an entry 1 in row v_i and column v_j of the model matrix when $(v_i, v_j) \in A^*$, otherwise the entries are 0. In this way, an entry 1 tells that its row's knockdown affects its column's regulator. Thus, the set $E_{v,t}$ of all knockdown experiments that affect regulator v in cell population t is given by the rows of $\mathcal{M}_t$ which have an entry 1 in column v:

$$E_{v,t} = \{\Delta^t w \in E_t | \mathcal{M}_t^{w,v} = 1\}. \tag{4.1}$$

In other words, the set of affecting experiments $E_{v,t}$ contains both the knockdown of the regulator itself, and knockdowns of its upstream activators in the pathway. Assuming the model $\mathcal{M}_t$ is correct, the experiments in $E_{v,t}$ are expected to have similar effect on the target genes of v. The effects on the genes, which are targets of other regulators upstream of v, should be different between the experiments. Example model matrices for the ATM pathway in the healthy and in the damaged cells are shown in Fig.4.1 B.

The input TF-targets The second kind of knowledge, the known TF-target relations, is also given separately for each cell population. It originates from reports about established individual TF targets, or from high-throughput TF DNA-binding data. The known TF targets are expected to show an effect, i.e., either be up- or down-regulated by the knockdown experiments. Those targets serve as examples of genes that are differentially expressed upon their TF knockdown. This kind of knowledge is rarely certain and in our approach is given as a belief (chapter 2) about the TF-target relationships rather than a fixed statement.

Processing steps JODA processes given data and knowledge in three steps (Fig.4.1 B):

1. In the first step, we analyze the input data from each knockdown experiment to estimate the effect of the knockdown on the genes. To this end, we apply our belief-based differential expression analysis method (chapter 2), implemented in the R package bgmm [118]. Our method assigns each gene a probability that it was differentially expressed in the experiment. In this step, the knowledge about the known TF targets is used. To improve the analysis for a knockdown of each regulator v in each cell population t, the known targets of v in t are given a high prior of differential expression in this knockdown. We add a sign to each returned probability to indicate whether the effect of the knockdown was up- or down-regulation. The signed probability lies in the $[-1, 1]$ interval. The resulting vector of signed differential expression probabilities of the genes in a knockdown of v in t is denoted $\mathbf{p}_v^t$. Together, we obtain $2|V|$ such vectors, one for each regulator in V and for each of the two cell populations.
2. In the second step, for each regulator v in each cell population t, we obtain a vector $\mathbf{R}_v^t$ of *regulation scores* that quantify the effect of v on the genes in t. In this step, the input knowledge about the pathway topologies is used. Recall that the pathway model $\mathcal{M}_t$ defines the set $E_{v,t}$ (Eq.4.1) of knockdown experiments that affect the regulatory activity of v in t. Each target gene of v is expected to have pronouncedly high or low signed differential expression probabilities in the knockdown of v that are consistent between all experiments in $E_{v,t}$. The regulation scores (each lying in the $[-1, 1]$ interval) reconcile the signed probabilities in the experiments in $E_{v,t}$ by taking an average:

$$\mathbf{R}_v^t = \frac{\sum_{w \in \{v_i | \Delta^t v_i \in E_{v,t}\}} \mathbf{p}_w^t}{|E_{v,t}|}. \tag{4.2}$$

   For example, in Fig.4.1 B the regulation scores $\mathbf{R}_{\text{RelA}}^d$ for RelA in the damaged cells are an average of the signed probabilities for the knockdowns of RelA and of its upstream activator ATM. In the healthy cells, only its own knockdown affects RelA, and its regulation scores $\mathbf{R}_{\text{RelA}}^h$ are the same as its signed probabilities $\mathbf{p}_{\text{RelA}}^h$. As explained above, under the condition that the model is correct, the experiments affecting a given regulator should have a common effect on this regulator's target genes. In other words, each target gene is expected to have either high or low signed

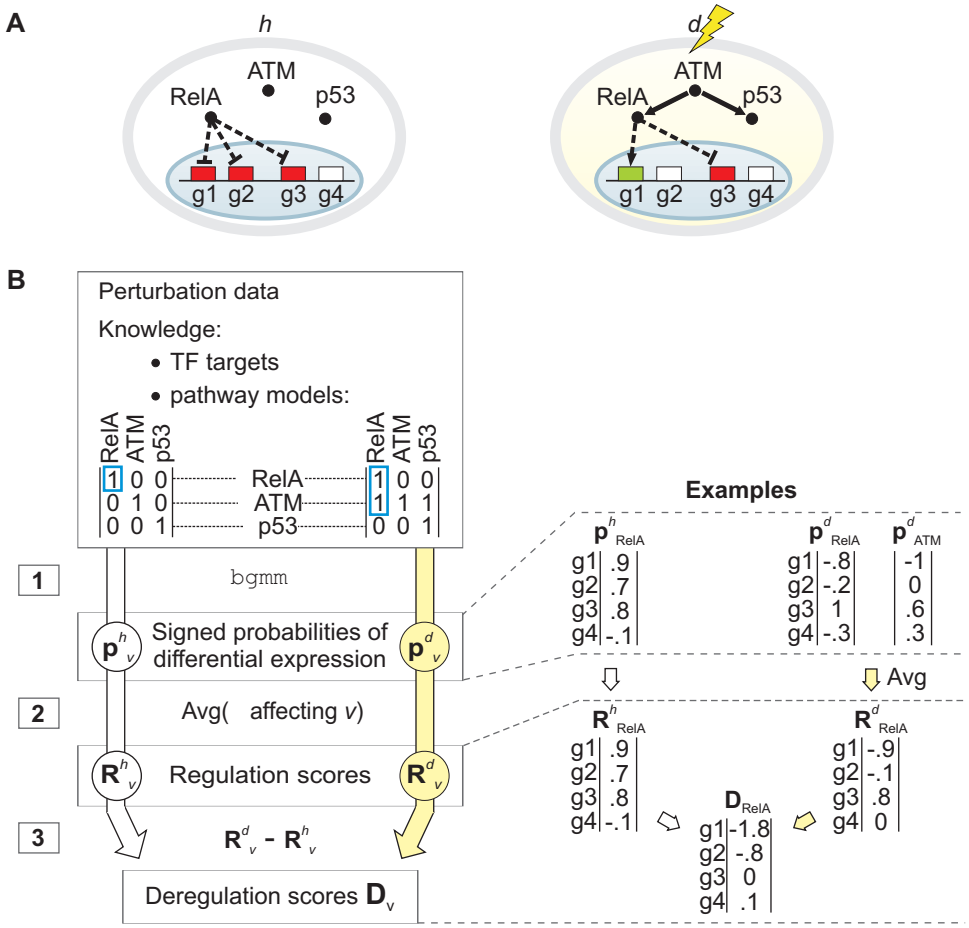


Figure 4.1: Method overview. (**A**) The ovals illustrate two different cells: a healthy cell h in a neutral environment (left) and a damaged cell d treated with neocarzinostatin (right). Inside each oval: a signaling pathway with regulators RelA, ATM and p53, and a set of remaining genes $g1$–$g4$. In d, ATM signals down to RelA and p53. In h the ATM pathway is inactive. Δ RelA – an experiment, where RelA is knocked down. Genes are colored according to the effect of the knockdown: up-regulation in red (indicating inhibition by RelA), down-regulation in green (activation by RelA) and no change in white. Dashed edges connect RelA to its target genes. (**B**) The JODA approach. Input: (i) knockdown data (here, together six knockdown experiments, for three regulators in two cell populations, h and d), (ii) known targets of those regulators, which are TFs and control gene expression directly, and (iii) known pathway models encoded in matrices with an entry 1 when a knockdown experiment (rows) affects the regulator (columns; otherwise the entries are 0). Experiments affecting RelA are marked in blue. The input is processed for the healthy (left) and the damaged cells (right) separately in three steps, until merged in deregulation scores, as indicated by the white and yellow flow arrows, respectively. Examples on the right illustrate the steps leading to deregulation scores for RelA. In the first step, we apply our method bgmm (chapter 2; [118]), which utilizes the known TF targets to better identify probabilities of differential expression of the genes in knockdown of each regulator $v \in \{\text{RelA, ATM, p53}\}$ (denoted $\mathbf{p}^h_v$ and $\mathbf{p}^d_v$ for the two cell populations h and d). The probabilities are signed to indicate whether the effect of knockdown was down- (negative sign) or up-regulation (positive). In the second step we obtain regulation scores $\mathbf{R}^h_v$ and $\mathbf{R}^d_v$, which quantify the effect of each regulator v on the genes in a given cell population. In the last step, we subtract regulation scores in the healthy cells from regulation scores in the damaged cells to obtain deregulation scores $\mathbf{D}_v$, quantifying how strongly each regulator v deregulates the genes.

probabilities of differential expression that are consistent between all affecting experiments. Thus, taking an average yields either high or low regulation scores for the true targets, and rules out those genes which respond to the perturbation experiments in a model-independent manner.
Note that a negative regulation score indicates (possibly indirect) activation of a target gene, and a positive score indicates inhibition. This rule, counter-intuitive at first sight, is motivated by the fact that genes with positive regulation scores have mostly positive probabilities of differential expression, i.e., tend to be up-regulated in those perturbation experiments that affect their regulator. The genes with negative scores have mostly negative probabilities, i.e., are down-regulated. Accordingly, we define genes *more activated* in a given cell population (e.g., damaged d), as having lower regulation score in this cell population than in the other (e.g., healthy h). For example, genes $g1$ and $g2$ in Fig.4.1 are more activated in d. $g1$ is (indirectly) inhibited by RelA in h, and (indirectly) activated by RelA in d. $g2$ is (indirectly) inhibited by RelA in h, and does not depend on RelA in d.

3. In the third step, to quantify deregulation of genes by a given regulator v, we define a vector $\mathbf{D}_v$ of *deregulation scores* as the difference of the regulation scores for v between the two cell populations. In this way, each deregulation score lies in the $[-2; 2]$ interval. Fig.4.1 illustrates these notions by a toy example. Two genes, $g1$ and $g2$, are deregulated between the healthy and damaged cells, while gene $g3$ stays regulated the same way, and $g4$ is unrelated to the pathway. Accordingly, $g1$ and $g2$ have dominant deregulation scores, which are well discriminated from the scores of $g3$ and $g4$ (Fig.4.1 B). Note that in the case when regulation scores for cell population h are subtracted from scores for cell population d, genes more activated in d (e.g., genes $g1$ and $g2$ in Fig.4.1) obtain negative deregulation scores, whereas genes more activated in h obtain positive scores.

4.2 Deregulated genes group into biologically relevant functional clusters

JODA was applied to identify genes deregulated in response to DNA damage induced by neocarzinostatin (NCS). NCS is a drug known to cause double strand breaks in the DNA. Our analysis aimed at a biological verification of the deregulation scores produced by JODA.

Input knockdown data We analyzed transcriptional effects of silencing the regulators ATM, RelA and p53, performed by Elkon *et al.* [32] on the healthy and the damaged cells. The raw dataset consists of 30 expression measurements, in normal and in NCS-treated human HEK293 cells, composed of three replicates for each siRNA knockdown of ATM, RelA, and p53, and six for control, in both cell populations (GEO series GSE1676, with 8794 genes measured). The raw data was normalized using quantile normalization and transformed into robust multi-array average

expression values [60]. Quality of the expression measurements was assessed with arrayQualityMetrics [67]. Four low-quality measurements were removed. We filtered out all genes without an ENTREZ identifier. In the case of multiple genes with the same identifier, we selected the one with the highest interquartile range (leaving 8498 genes). Consequent removal of outliers left 8463 genes. To provide input to JODA, we calculated vectors of log mean expression ratios for each knockdown versus control in both cell populations (averaging over repeats; together six).

Input knowledge Additionally, we provided two kinds of knowledge. First, the ATM pathway topologies in the damaged and in the healthy cells. As presented in Fig.4.1 A, in the damaged cells NCS triggers a cellular pathway, where the central kinase ATM signals down to TFs RelA and p53. This pathway is inactive in the healthy cells [76]. Second, known target genes were collected for p53 in both healthy and damaged cells, and for RelA only in the healthy cells. For p53 in the damaged cells, we composed a set of 47 known targets by selecting genes that have a *DNA repair* or *chromatin modification* function from experimentally verified p53 targets collected by Horvath *et al.* [53], the direct p53 targets detected with ChIP-PET and confirmed by expression analysis by Wei *et al.* [129], and finally by adding genes targeted by p53 upon ionizing radiation [62]. For p53 knockdown in the healthy cells, we took those verified targets of Horvath *et al.*, and those direct p53 targets of Wei *et al.*, which were not selected as targets in the damaged cells. Finally, for the analysis of RelA knockdown in the healthy cells we utilized a set of genes, identified using the ChIP-PET technology by Lim *et al.* [80], whose promoters are bound by RelA and contain an NF-κB consensus motif.

Analysis of the deregulation lists Application of the approach resulted in three lists of deregulation scores (shortly, *deregulation lists*), one for each of the regulators RelA, ATM and p53. We sorted the lists decreasingly, so that the one extreme of each list contains genes more activated in the healthy cells, and the other contains genes more activated in the damaged cells. We performed Gene Set Enrichment Analysis (GSEA [116]) with default parameters to identify gene sets significantly overrepresented on the extremes of the sorted deregulation lists. Sets with fewer than 15 and more than 500 genes were excluded from the analysis. Only results with FDR ≤ 0.01 and FWER ≤ 0.5 were considered significant. We focused the analysis first on Gene Ontology (GO [5]) term, and second on pathway enrichment (both taken from the MSigDB database [116] that is utilized by GSEA). We discuss the overrepresented MSigDB pathways in section 4.3.

Functional clusters Fig.4.3 presents the identified overrepresented GO terms (together, 51) and their enrichment in the deregulation lists. The terms were grouped by similarity into *functional clusters*. Similarity between the GO terms was assessed using GOsim [36] with the 'relevance' measure [105]. Next, the terms were hierarchically clustered by this similarity. We checked the possible clusterings with the number of clusters from five to twenty (Fig.4.3 A,B). For an assumed number of

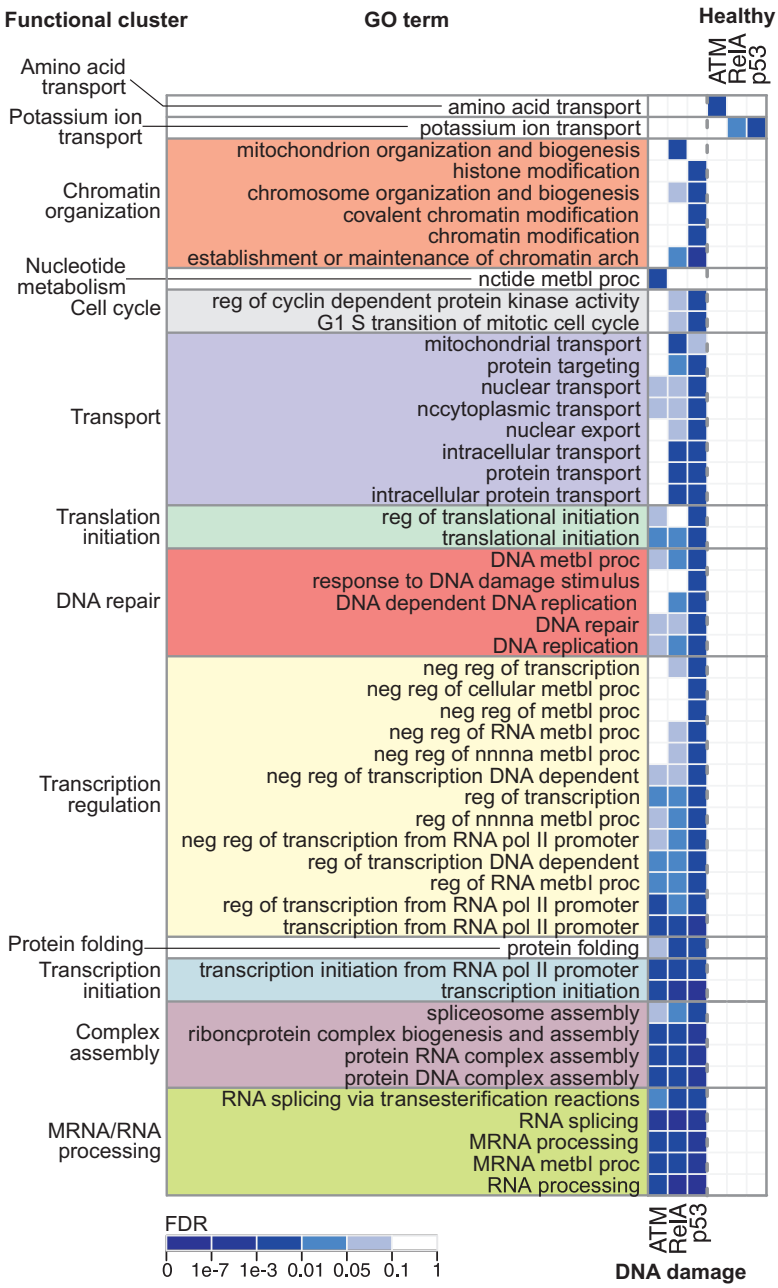


Figure 4.2: Functional enrichment. The matrix shows GO terms enriched with high confidence (FDR ≤ 0.01, indicated in blue, and FWER ≤ 0.5; identified using GSEA; [116]) in the genes more activated in the damaged cells by ATM, RelA and p53 (left three columns) and more activated in the healthy cells (right three columns). Each GO term shown is enriched in at least one column. The terms were grouped into functional clusters with names indicated on the left, and sorted by the average enrichment in the first three columns. The GO term enrichment is mutually exclusive for the genes more activated in the healthy and in the damaged cells. Eleven functional clusters of terms are enriched exclusively in genes more activated in the damaged, and two exclusively in the healthy cells. Abbreviations: mtbl, metabolic; nc, nucleo; pol, polymerase; reg, regulation; neg, negative; pos, positive; proc, process; arch, architecture; nnnna, nucleobase, nucleoside, nucleotide, and nucleic acid. The identified clusters confirm that the dominant deregulation scores are correlated with a functionality which is highly relevant to the switch between the compared cell populations.

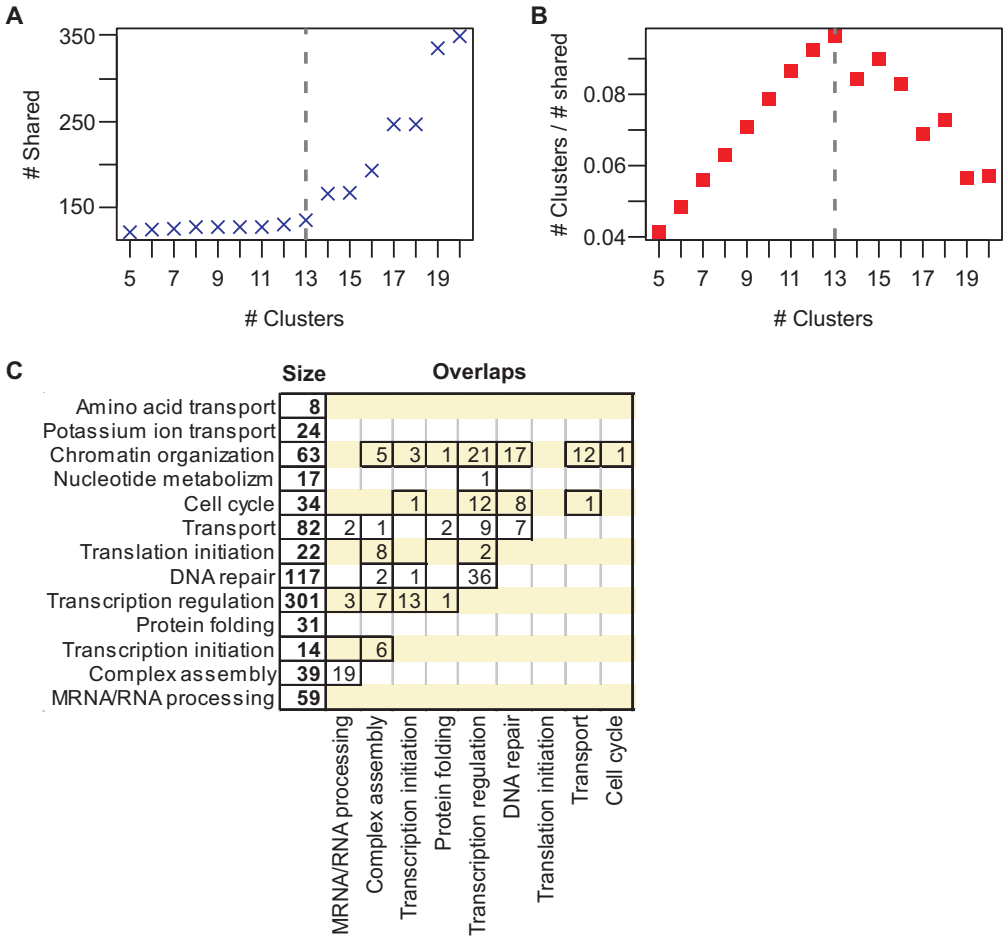


Figure 4.3: Choice of the number of clusters. (**A**) The total number of unique genes shared between the clusters, traced over all clusterings with the numbers of clusters from five to twenty. The set of genes shared between the clusters is obtained as the union of all pairwise cluster intersections. The number of shared genes increases with the number of clusters. (**B**) The number of clusters in our functional clustering is selected from the $[5; 20]$ interval so that the ratio of the number of clusters over the number of shared genes is maximized. The more clusters, the more functions is represented in the clustering (one cluster groups a set of GO terms). The less shared genes, the smaller overall overlap between the clusters. Gray dashed line in **A**, **B** marks the selected cluster number (thirteen). (**C**) Functional gene cluster sizes and overlaps. The matrix represents the clusters (rows), their sizes (Size column), and their pairwise overlaps (Overlaps matrix). Columns of the overlaps matrix correspond only to those clusters that do overlap with any other cluster. Entries show the non-zero number of genes in the overlap and otherwise are empty.

clusters, the GO term clusters were formed by cutting the hierarchical clustering tree on a corresponding level. Next, from the functional clustering of GO terms, we obtained a functional clustering of genes, where each gene cluster corresponds to one GO term cluster. To this end, we collected the deregulated genes that are annotated with the terms from the GO term clusters, using the following procedure: First, for each GO term, we collected the corresponding deregulated genes in three steps: (i)

Identify the deregulation lists in which this term is significantly overrepresented. (ii) From each identified deregulation list, collect the leading edge genes for this term, i.e., genes that contributed to the enrichment of the term in this list [116]. (iii) Take the intersection of the sets of genes collected from all lists identified for this term. Next, for each cluster, we took a union of the sets of genes collected for the terms in this cluster. The number of clusters in both GO term and gene clusterings was set to thirteen, minimizing the overlap between the gene clusters. The resulting gene clustering separates 611 genes. Each functional cluster was assigned a general name, summarizing the GO terms grouped in this cluster (Fig.4.3 C).

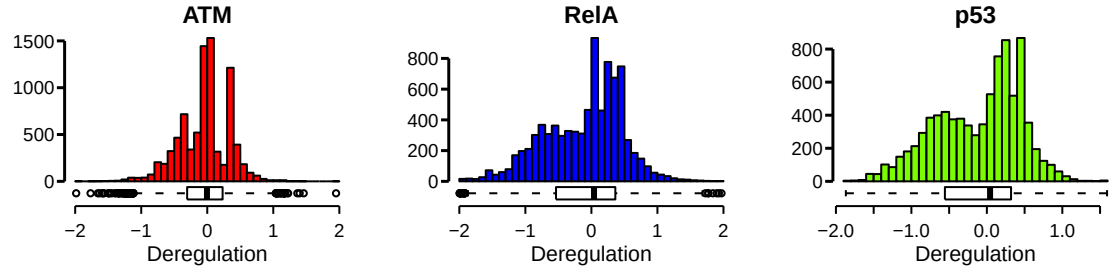


Figure 4.4: Distribution of the deregulation scores. Histograms of the deregulation scores of all measured genes for the regulators ATM, RelA and p53. Below each histogram, a boxplot of the scores is shown.

Several functional clusters, e.g. *DNA repair*, *Chromatin organization*, *Transcriptional regulation* and *Cell cycle*, indicate that our method assigns dominant deregulation scores to genes playing crucial roles in response to DNA damage. Additionally, we find enrichment of deregulated genes in *RNA/MRNA and nucleotide processing*, *Complex assembly*, *Protein folding*, *Transport* as well as transcription- and translation-related processes. This rich involvement of genes up-regulated in response to DNA damage in various processes is in agreement with previous findings [112, 78].

Eleven functional clusters of GO terms are found for the genes more activated in the damaged cells and only two in the healthy cells, even though the distributions of the deregulation scores have the median of zero and are not biased in number towards the negative values (Fig.4.4). The eleven clusters more activated in the damaged cells are shortly referred to as *damage-activated*, and the two more activated in the healthy cells are called *healthy-activated* throughout the text. Strikingly, the regulators agree on the functional processes they activate: no GO term overrepresented in the genes more activated in the damaged cells is also overrepresented in the genes more activated in the healthy cells. This shows the tightly coordinated way in which the ATM pathway governs the downstream response to the damaging agent.

The main general function of each cluster is captured by its label. We used the Ingenuity Pathway Analysis software (Ingenuity Systems) to annotate the largest clusters of genes with additional, secondary functions. Importantly, enrichment analysis of the *DNA repair*, *Transcription regulation* and *Chromatin organization* clusters revealed

that they also contribute to *cell death*, *cell cycle*, as well as *cellular growth and proliferation* and *DNA replication, recombination, and repair*. These three clusters are also significantly enriched in cancer-related genes. All three have strong enrichment for tumorigenesis processes, leukemia-related genes, as well as other cancer types, which agrees with the well known connection between DNA damage and cancer [51].

4.2.1 The deregulated functional clusters and pathways cannot be found without prior knowledge or in separate analysis

We next verify whether the functional clusters of genes with extreme deregulation scores given by JODA are significant. Fig.4.5 shows that the functional clusters of genes have deregulation scores that stand out significantly from the background of deregulation scores for all analyzed genes. The power of JODA becomes apparent when comparing it to a separate analysis of the two cell populations, or an analysis without incorporation of prior knowledge.

Advantage over separate analysis We compare the deregulation scores to regulation scores. Regulation scores are obtained in the second step of JODA, separately for each cell population (Fig.4.1). We thus refer to an analysis of the regulation scores as a *separate analysis*. Several clusters, although performing functions important for the switch between the healthy and damaging environment, are likely to be missed when analyzing the cell populations separately. Consider an example of the *DNA repair* cluster. Interestingly, positive regulation scores suggest strong inhibition of genes in this cluster in the healthy cells. The regulation scores in the damaged cells are distributed around zero and therefore do not indicate activation nor inhibition (Fig.4.5 A). Based on the regulation scores in the damaged cells only, the genes in this cluster cannot be significantly differentiated from all genes (Fig.4.5 B). The separate analysis misses gene sets that are only slightly down-regulated in one cell population and slightly up-regulated in the other. Deregulation scores, being a difference of the small but opposing effects, amplify them, making detection of such gene sets possible.

Advantage over analysis without incorporation of knowledge To compare our knowledge-based JODA approach to analysis without incorporation of knowledge, we assessed deregulation using *decorrelation* scores. To this end, for each analyzed gene and each regulator (ATM, RelA, and p53) we computed Pearson correlation first between the expression profiles of the regulator and the gene measured in the healthy cells, and second between the profiles of the regulator and the gene in the damaged cells. Strong positive correlation in a given cell population can be interpreted as an activation of the gene by the regulator in this cell population, whereas strong negative correlation can be interpreted as inhibition. To compute the decorrelation scores for each regulator, we subtracted the correlation values for all genes in the damaged cells from the correlations in the healthy cells. In this way, the decorrelation scores,

belonging to the interval $[-2, 2]$, can be read similarly as the deregulation scores: strongly negative decorrelation scores indicate more activation in the damaged cells, and strongly positive indicate more activation in the healthy cells. The decorrelation scores are a simple implementation of the ideas applied by Taylor *et al.* [120] and Mani *et al.* [84]. Taylor *et al.* used Pearson correlation of interactome hubs to their interaction partners to verify whether these interactions are context-specific. Mani *et al.* investigated gain and loss of correlation between cell populations using a mutual information-based approach.

Although interpreted in the same way, the deregulation scores differ from the decorrelation scores by the ability to incorporate given prior knowledge about the known cell population-specific pathway topology and target genes downstream of the pathway. JODA outperforms also deregulation analysis assessed with the decorrelation scores. Using the decorrelation scores, the same two clusters can be identified as healthy-activated and eleven as damage-activated, but they are less significant than when deregulation scores are used (Fig.4.5 B).

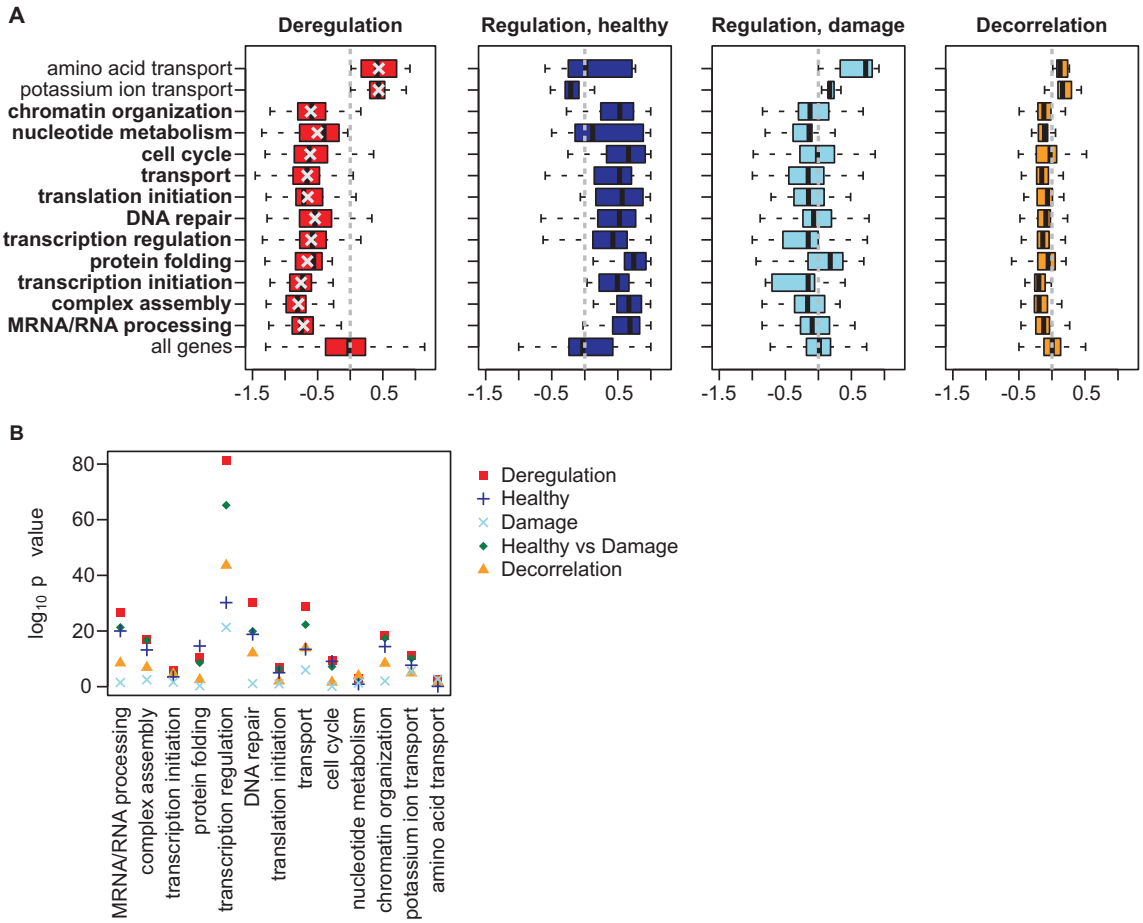


Figure 4.5: Significance of the deregulation scores. (**A**) Distributions of deregulation scores of genes in the functional clusters (averaged over the three regulators, ATM, RelA and p53) strongly deviate from the distribution of averaged deregulation scores for all genes (left plot). The distributions of average regulation scores in the healthy (middle left) and in the damaged cells (middle right), as well as of average decorrelation scores (right) are more similar to the distributions of the same scores for all genes. Gray dashed vertical lines mark score 0 in each plot. Decorrelation scores do not incorporate prior knowledge and assess the difference of expression correlations between the regulators and the genes. To compare significance between the clusters, the mean of averaged cluster deregulation scores (left, light gray crosses) can be used. The eleven damage-activated clusters are assigned negative deregulation and decorrelation scores and have names printed in bold. Healthy-activated clusters have positive deregulation and decorrelation scores. Separately for each cell population, negative regulation scores indicate activation of the clusters, whereas positive regulation scores indicate inhibition. (**B**) A t-test comparing the cluster deregulation scores with the deregulation scores for all genes (Deregulation; red squares) gives for majority of the clusters the most significant p-values, when contrasted with: the p-values obtained in a t-test comparing cluster regulation scores to regulation scores of all genes in the healthy cells (Regulation, Healthy; blue pluses), and the same t-test but in the damaged cells (Regulation, Damage; light blue crosses), the p-values in a t-test comparing cluster regulation scores in the healthy directly to regulation scores in the damaged cells (Healthy vs Damage; green diamonds), and in a t-test comparing the cluster decorrelation scores with the decorrelation scores for all genes (Decorrelation, yellow triangles). All tests are two-sided. The p-values evaluate the differentiating power of the scores for each cluster. They depend on cluster sizes and therefore should not be compared between the clusters. Taken together, our joint and knowledge-based approach assigns more significant scores to the functional clusters than a separate analysis, or analysis without incorporation of knowledge.

4.3 Deregulated pathways and complexes elucidate cooperation within functional clusters

Deregulated pathways and complexes Since the genes in each cluster share the same functionality, they may interact in a common cellular pathway or complex. To determine these interactions we first identified pathways and complexes that are over-represented on the extremes of the deregulation lists. Next, we checked their overlap with the functional clusters. The enrichment in pathways was assessed using GSEA (see section 4.2). The identified pathways are stored in the MSigDB database as sets of genes, but their signaling relations are well described in the literature. Eleven identified MSigDB pathways significantly overlap with our functional clusters (Fig.4.6 A). For example, the *apoptosis* pathway contains genes from the *DNA repair* and *Transcription regulation* clusters. To test enrichment in complexes, we downloaded sets of genes forming each complex from the Reactome database ([123]; together, 2816 complexes). For a set of genes in a given complex, and for a given deregulation list, we performed higher-tail hypergeometric enrichment tests iteratively for a number of 10 up to 500 most extreme (top or bottom) deregulated genes. Finally, the minimum resulting p-value was selected to signify the enrichment of this complex in this deregulation list. To focus on big complexes containing a considerable number of deregulated genes, we excluded complexes with fewer than 15 genes, overlapping by less than 10 with the current set of deregulated genes in each iteration. The size of the universe was set to all genes analyzed on the array (8498). Only results with the enrichment p-value $\leq 10^{-5}$ were considered significant. We found the *Exon junction* complex and several spliceosome complexes (Fig.4.6 B) significantly overrepresented in the genes more activated in the damaged cells. Interestingly, these complexes overlap (hyper-geometric higher tail p-value $1.1 \cdot 10^{-29}$) with the *MRNA processing* cluster. Similarly as pathway interactions, membership in complexes explains the way the genes in the clusters are connected and collaborate to exhibit the common function.

Connectivity within the *DNA repair* cluster Finally, we focused on the *DNA repair* cluster, which is of pivotal interest in the context of the switch between the healthy and the damaged cells. We investigated physical relations connecting genes within this cluster. The cluster is strongly enriched in eight canonical pathways involved in the DNA damage response (p-values from $1.33 \cdot 10^{-41}$ to $8.17 \cdot 10^{-11}$; identified using SPIKE [32]). SPIKE is a database and an analysis tool, storing manually curated pathways playing key roles in response to damage. The table in Fig.4.7 A lists 51 genes from the *DNA repair* cluster, which belong to those canonical pathways stored in SPIKE as well as three additional pathways, described in a comprehensive review on DNA damage response by Wood *et al.* [133]. The position of the listed genes in the well known damage response pathways describes their role in the response, as well as their interaction partners in the cluster.

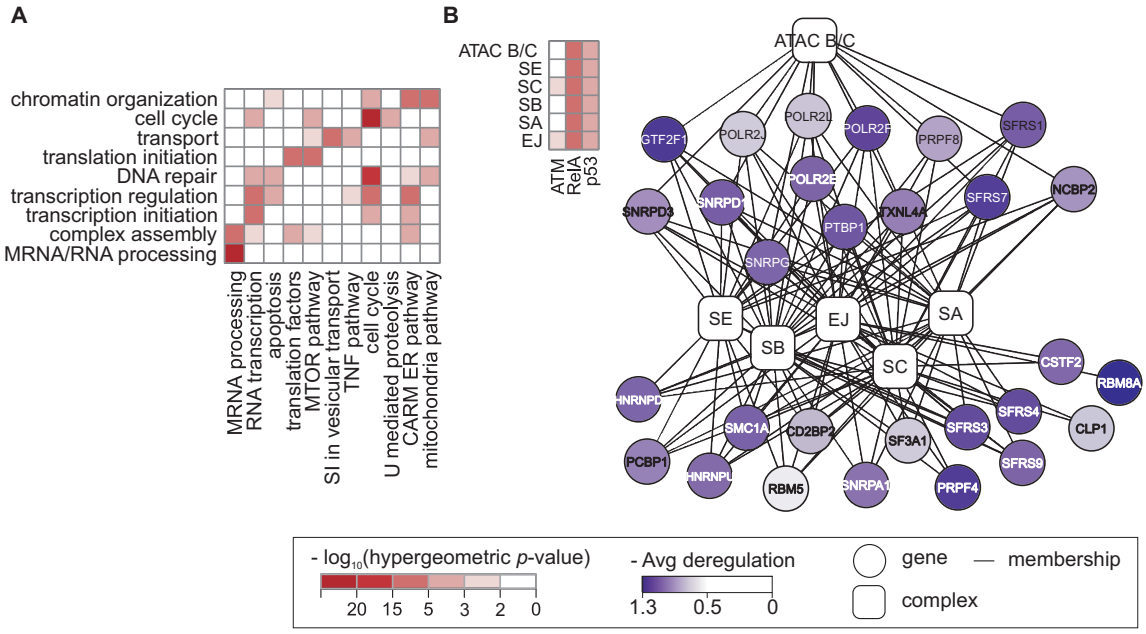


Figure 4.6: (**A**) Connectivity between genes in functional clusters explained with deregulated pathways stored in the MSigDB database. Matrix shows pathways enriched in the genes more activated in the damaged cells (columns) that overlap significantly with functional clusters (rows). Only pathways and clusters overlapping with higher-tail hypergeometric p-value at most 0.001 (indicated in red) are reported. Abbreviations of pathway names: SI, snare interactions; U, ubiquitin. (**B**). Deregulated complexes. Top left: The exon junction (EJ) complex and five spliceosome complexes (rows) are overrepresented in the genes more activated in DNA-damage by the regulators ATM, RelA and p53 (columns). Only complexes with a p-value at most 10^{-5} (indicated in red) are reported. Genes in those complexes overlap significantly with the functional cluster *MRNA/RNA processing*. Right: Graph representation of the genes (round nodes shaded in violet by their average deregulation scores) in the complexes (rounded square nodes). Edges represent gene - complex membership. Abbreviations of complex names: EJ, exon junction; SA, spliceosomal A; SB, spliceosomal B; SC, spliceosomal Active C/ spliceosomal intermediate C/ spliceosomal active C with lariat containing 5-end cleaved pre-mRNP:CBC; ATAC B/C, ATAC B/ ATAC C/ ATAC C with lariat containing 5-end cleaved mRNA ("/" lists complexes sharing common genes that have identical enrichment p-values and thus are abbreviated with the same name). Pathways and complexes carry information about relations between their member genes. Therefore, these enrichment results broaden our view on the connectivity within the sets of genes in the functional clusters.

To further infer the cooperation between the 66 remaining genes in the *DNA repair* cluster, we collected their interactions using SPIKE and Ingenuity (Fig.4.7 B). Together, we identified 126 relations connecting 52 of those genes either with each other or with other intermediate genes, complexes and protein families. SPIKE was applied to find all direct connections that are stored in the SPIKE database that join the set of 66 genes, allowing connection via a single intermediate node not included

in the set. The connections may represent membership in a complex or regulation of different biochemical types, e.g. phosphorylation, protein-DNA (transcriptional) regulation, activation and protein-protein interaction. Ingenuity was applied to find interconnections between the 66 genes in two different ways, always restricting that each relationship was reported for Human molecules, and that it is of one of the following types: expression, transcription, protein-DNA (all summarized as transcriptional regulation), activation, inhibition, membership, modification, phosphorylation, or protein-protein interaction. We collected all such direct relationships that are stored in the Ingenuity database. In addition, we applied Ingenuity to score known networks based on their enrichment in the input set of genes and collected all direct interactions present in the top three scoring networks (with scores 57, 45 and 18, respectively). The top scoring networks are related mostly to DNA replication, recombination, and repair, as well as tumor morphology, cell cycle and cell death. The networks have additional nodes that are not included in the input set but are highly connected to the genes in the set.

The analysis revealed a number of complexes, like the Origin of Replication complex (ORC) containing five *DNA repair* genes, which join subsets of genes together. Grouping the complexes by common functionality, we selected functional sub-parts of the network. For example, we identified a sub-network of genes belonging to the RFC, DNA polymerase epsilon, and the ORC complexes, which are involved in the DNA replication process (marked with a light grey background in Fig.4.7 B).

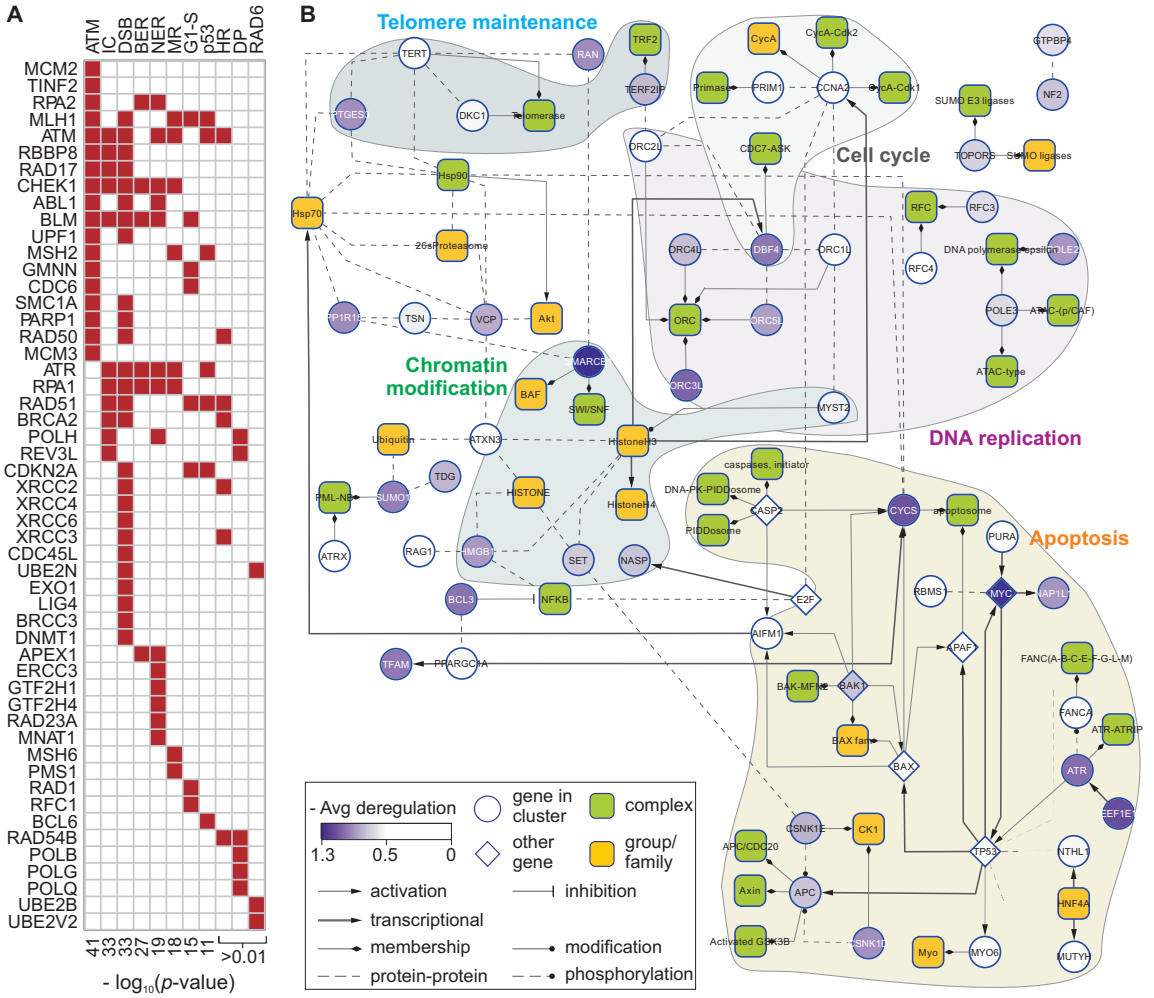


Figure 4.7: Cooperation of genes in the *DNA repair* cluster. (**A**) The matrix shows 51 out of 117 genes in the *DNA repair* cluster belonging (marked in red) to eleven known pathways involved in DNA repair (columns; listed on the top). First eight of those pathways are strongly enriched in the cluster (identified using SPIKE; [33]) and three other [133] overlap with the cluster, but not significantly (p-values listed on the bottom). Many genes are shared between the pathways. Abbreviations: ATM, ATM pathway; IC, repair of interstrand crosslinks; DSB, repair of double strand breaks; BER, base excision repair; NER, nucleotide excision repair; MR, mismatch repair; G1-S, G1-S pathway; p53, p53 pathway; HR, homologous recombination; PD, polymerase; RAD6, RAD6 pathway. Such strong enrichment in canonical DNA damage response pathways confirms the biological relevance of the deregulated genes in the *DNA repair* cluster. (**B**) To identify interconnections between the remaining genes in the *DNA repair* cluster, we searched for pathways of length at most one connecting each pair of those genes in a protein-protein and protein-DNA interaction network (using SPIKE and Ingenuity). The resulting graph connects 33 genes (remaining 33 are isolated and not displayed) and represents many complexes to which the genes belong. Some of the complexes are involved in the same processes: DNA replication, apoptosis, cell cycle, or telomere maintenance. The network explains connectivity within the cluster that goes beyond the canonical pathways.

4.4 Genes most activated in damaged cells work in the damage response

Functionality of the top hundred most activated genes Functional clusters contain deregulated genes that accumulate within the extremes of the deregulation lists, but not on the strict top or bottom. We investigated the composition of three sets of the strict top one hundred genes that are most activated in the damaged cells by each regulator RelA, ATM and p53. All three sets are significantly enriched in genes involved in *transcription*, with five common genes active in this process: CHD4, RBM14, RCAN1, SMAD4, and UBN1. Interestingly, some of the genes most activated by RelA are interaction partners (for example, SMARCB1) of the genes most activated by p53 (SMARCB4). Apart from *transcription*, the set most activated by ATM is also enriched in *cell death*, *cell cycle* and *growth and proliferation*-related genes. Additionally, both the sets of genes most activated by ATM and p53 are enriched in cancer-related genes.

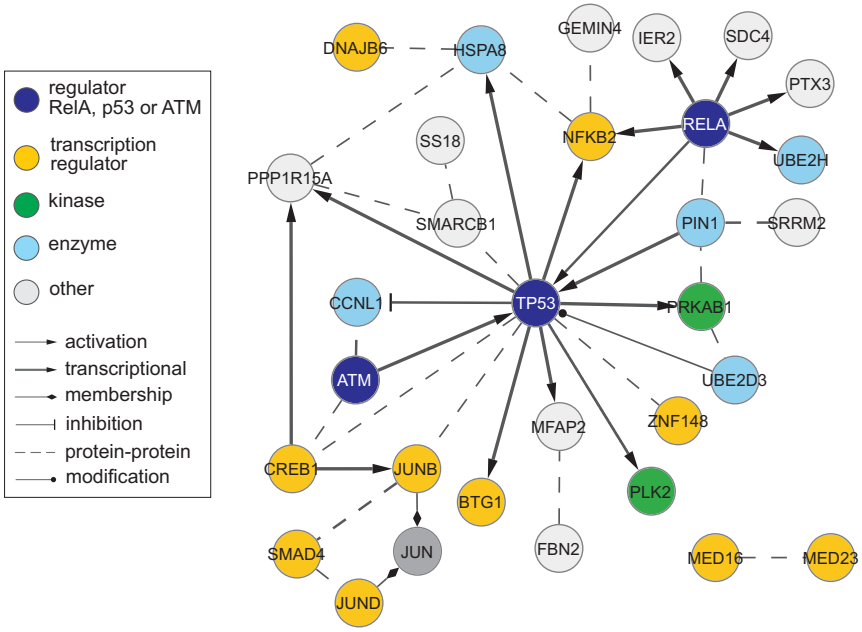


Figure 4.8: Network of genes most activated in DNA damage. A network of known regulatory and interaction relations (edges) connecting genes (nodes) from the lists of top twenty most activated by RelA, ATM or p53 in DNA damage. The relations are collected from the Ingenuity Pathway Analysis (Ingenuity Systems) database. The nodes are labeled with gene names and colored according to gene functions, whereas relations are given edge styles according to their type.

Regulatory relations between the top twenty most activated genes Next, we reviewed the individual examples out of three shorter lists of twenty genes, that are

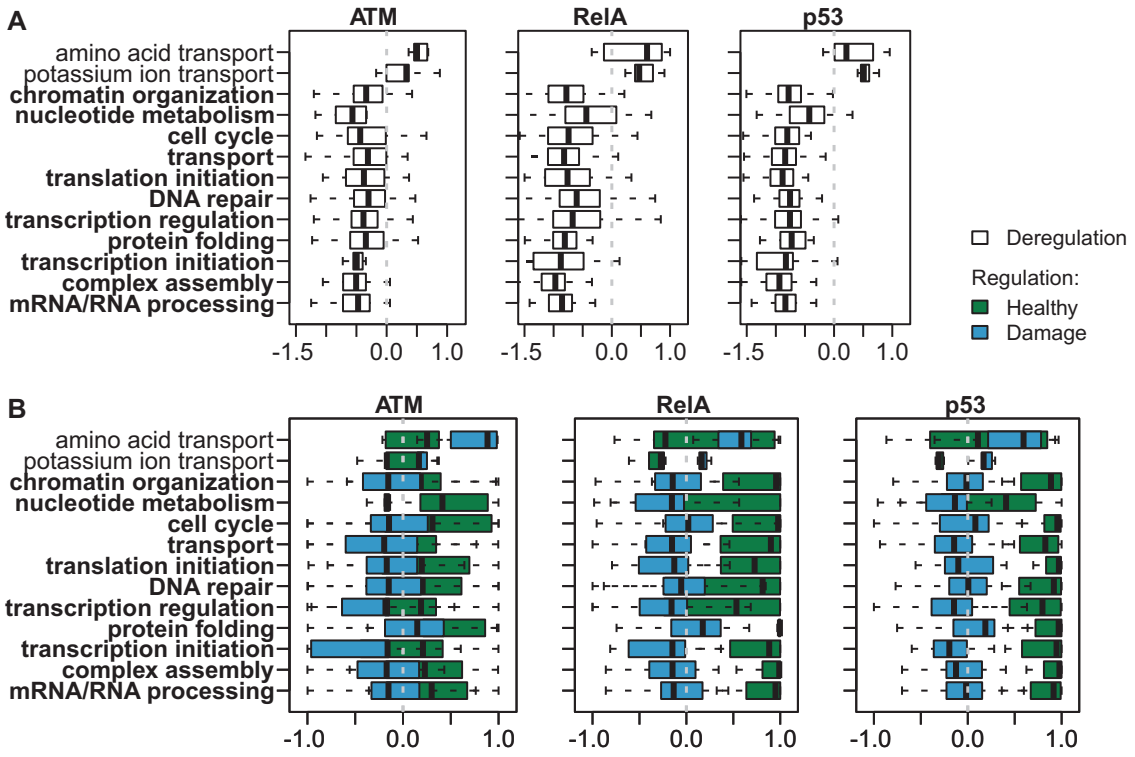


Figure 4.9: Summary of indirect deregulation and regulation of each functional cluster. (**A**) Distributions of the deregulation scores (x-axis) of the genes in the functional clusters (y-axis). (**B**) Distributions of the regulation scores (x-axis) in the healthy cells (drawn in green) and in the damaged cells (drawn in blue) of the genes in the functional clusters (y-axis). In **A** and **B** the score distributions are plotted separately for the three regulators, from left to right: ATM, RelA and p53. The eleven damage-activated clusters are assigned negative deregulation scores and have names printed in bold. Healthy-activated clusters have positive deregulation scores. For each cell population separately, negative regulation scores indicate activation of the clusters, whereas positive regulation scores indicate inhibition.

most activated in the damaged cells by RelA, ATM and p53. These shorter lists contain together 51 unique genes. Fig.4.8 presents a network interconnecting 28 of the 51 genes, for which regulatory interconnections are known. Both p53 and RelA, with seven and five regulatory targets, respectively, are major regulators for the genes in this network. Moreover, 10 out of the 28 genes in the network are transcription regulators themselves. From all 51 most activated genes there are 12 transcription regulators, 19 genes involved in apoptosis, 18 in proliferation and 6 in cell cycle progression.

4.5 RelA and p53 are the key deregulators of genes in functional clusters

Deregulation is inferred from knockdown effects, and as such can be due to an indirect impact of the regulators on the genes. Here, we summarize these possibly indirect effects on functional clusters of the deregulated genes identified by JODA.

Fig.4.9 reports deregulation and regulation scores of the genes in functional clusters, for each regulator ATM, RelA and p53. The per cluster distributions of deregulation scores for RelA and p53 are shifted further away from zero than for ATM, suggesting a stronger deregulatory impact on the clusters (Fig.4.9 A). Indeed, Fig.4.9 B shows that the distributions of regulation scores for ATM in the healthy and in the damaged cells are generally less separated than for RelA and for p53. Interestingly, for all three regulators, the regulation scores indicate that the damage-activated clusters are only slightly (possibly indirectly) activated in the damaged cells. Instead, these clusters are strongly (possibly indirectly) inhibited in the healthy cells both by RelA and by p53, as indicated by the respective distributions shifted towards value 1. The inhibitory impact of ATM on these clusters in the healthy cells is less prominent. In the case of the two healthy-activated clusters, a strong, possibly indirect inhibition in the damaged cells is observed for all three regulators. Distinctively, the *Potassium ion transport* cluster is also (possibly indirectly) activated in the healthy cells by RelA and p53.

4.6 Deregulation can be explained by a hierarchy of direct TF-DNA binding events

Finally, we investigate the hierarchy of direct regulatory relations, which could explain the effect of the ATM pathway on the deregulated target genes. The first expected scenario would involve regulation by direct binding of the regulators in the pathway to the gene promoters. Alternatively, the most parsimonious hierarchy would connect the regulators to the genes via a single TF. To investigate these hypotheses, we follow a two step procedure. In the first step we computationally predict the TFs directly binding to the promoters of the genes. In the second step we verify whether the TFs are the regulators themselves, or whether they are controlled by the regulators.

TF-DNA binding predictions To implement the first step, we applied TransFind [68] to search the promoters of the genes in each functional cluster for over-representation of high-affinity binding of human TFs, which is conserved in their mouse orthologs (Fig.4.10 A). Since the genes in each cluster share the same functionality, they can be expected to be directly regulated by a common TF. Given a set of genes, TransFind predicts TFs with affinities to the gene promoters significantly higher compared to a background set of genes (by default, all genes in the Ensembl57

database). Affinities are computed from a physical model, based on positional TF weight matrices. We tested a reduced set of human TRANSFAC [130] matrices, containing only a single, the most informative matrix for each TRANSFAC TF, setting all parameters to default. TransFind assesses the significance of binding to the promoters of a set of input genes with a multiple-testing-corrected (FDR) version of the Fisher's exact test. Only results with FDR ≤ 0.05 were considered significant. Among the identified TFs, CREB has binding sites significantly enriched in the promoters of genes in the *DNA repair* cluster. Neither RelA nor p53 were predicted to bind directly to the promoters of the genes in the functional clusters.

CREB as the intermediate factor in the hierarchy In the second step, we consider the hypothesis of the parsimonious hierarchy. Here, we focus on CREB, leaving other predicted TFs as candidates for future investigation. The hypothesis consists of a deregulatory connection from the ATM pathway to CREB, implemented by RelA or p53 directly binding to CREB promoter in the damaged cells and not binding in the healthy cells. To complete the picture, we collected the most likely direct target genes of CREB. Based on high-throughput CREB binding data in HEK293T cells by Zhang *et al.* [140], seven out of top twenty predicted CREB targets in the *DNA repair* cluster bear evidence of CREB binding (Fig.4.10 B). Additionally, we report two genes outside of the cluster, PWP1 and NOLC1. Both are deregulated in our system, as well as have yeast homologs, which according to the study by Workman *et al.* [134] are deregulated by SKO1, an yeast homolog of CREB. PWP1 is also reported as bound by CREB in HEK293T cells [140]. Fig.4.10 C brings together these pieces of evidence in a hypothetical regulatory network. The network shows a two-step hierarchy, going from the ATM pathway, via CREB, to the nine most likely CREB target genes, which are deregulated between healthy and the damaged cells.

A

TF	FDR	
E2F	2E-06	DNA repair
NRF-1	0.027	
CREB	0.027	
ATF4	0.008	Transcription regulation
AhR Arnt	0.024	
ATF6	0.024	
WT1	0.024	
HLF	0.03	
MAZ	0.03	
NRF-1	0.03	
CREB	0.03	
HOXA7	0.03	

B

Name	AvgD	YH	DB	YPD	MMS	CREB
EXO1	-1.07	EXO1	0.7	0.3	1	0
PPP1R15A	-0.82	NA	NA	NA	NA	0
NAP1L1	-0.78	NAP1	0.3	1	0.8	0
ORC5L	-0.76	ORC5	0.3	1	1	0
BRCA2	-0.57	NA	NA	NA	NA	0
MCM2	-0.54	MCM2	0.2	1	0.6	0
UBE2B	-0.54	RAD6	0.2	0.4	0.1	0
PWP1	-1.05	PWP1	0	1	0	0
NOLC1	-0.83	SRP40	0	0.3	0	0.43

C

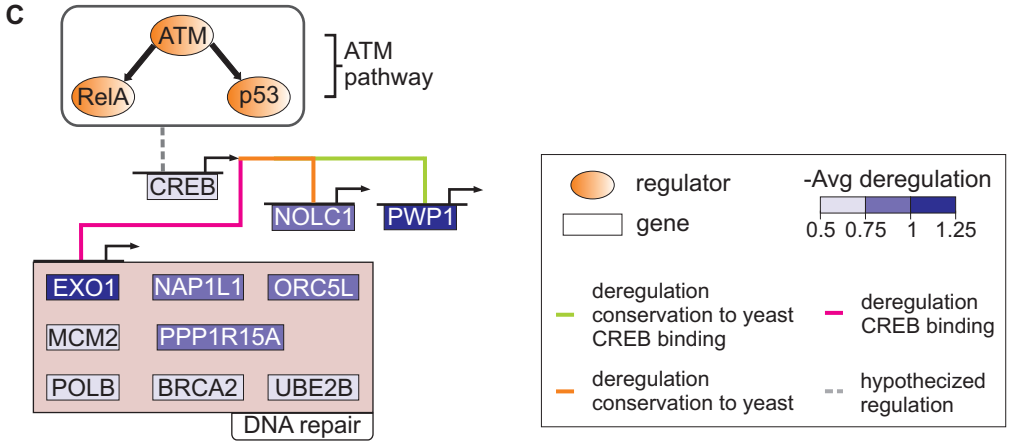


Figure 4.10: A hypothetical deregulatory hierarchy. (**A**) TFs with high affinity binding overrepresented in the promoters of genes in the functional clusters and in promoters of their mouse orthologs. Only binding predictions with FDR ≤ 0.05 are shown. (**B**) Nine deregulated genes with strong evidence of regulation by CREB (gene names are given in the Name column; deregulation scores averaged over RelA, ATM and p53 in the AvgD column). Seven genes belong to the *DNA repair* cluster and have a high affinity binding of CREB to their promoters conserved in mouse (colored background). PWP1 and NOCL1 have yeast homologs (given in the YH column; NA – homolog not known), which are *deletion-buffered* (DB – deletion-buffering p-value; [134]) by SKO1, a CREB homolog in yeast. That means, both genes change expression in wild-type cells in response to methyl-methanesulfonate (shortly, MMS, a double-strand breaks-inducing drug) and do not change when SKO1 is knocked down. Moreover, the promoters of PWP1 and NOLC1 homologs are not bound by SKO1 in the healthy yeast cells (binding p-values for the SKO1 binding given in the YPD column), and are bound by SKO1 in the cells damaged by MMS (p-values in the MMS column). Promoters of all nine genes but NOLC1 are bound by CREB in HEK293T cells (p-values, averaged over three time points of forskolin stimulation given in the CREB column; [140]). (**C**) Putative gene deregulatory network. Top: ATM pathway. Middle: The pathway deregulating CREB. Below: CREB regulating its most likely gene targets (genes with additional evidence shown in **B**). Genes are colored in shades of violet according to their deregulation, averaged over the regulators RelA, ATM and p53. The hierarchy is a hypothetical mechanistic explanation of deregulation of those genes, observed between the healthy and the damaged cells. The ATM pathway indirectly deregulates the genes by deregulating CREB.

4.7 Discussion

Reprogramming of cellular character manifests itself in different cellular signaling, and, in a consequence, changes of the downstream regulatory relations. In this chapter we presented an approach for quantifying these changes. JODA combines cell population-specific data and prior information from the interconnected levels of the pathway and of gene regulation.

The information about the pathway topology in a given cell populations is formalized in a simple model. Note that neither the pathway topologies nor the pathway models are intended to capture the dynamics and full spectrum of molecular interactions in signaling pathways. Instead, they are static and limited only to activatory signaling relations. The chosen modeling formalism is adjusted to the analyzed ATM pathway and available qualitative knowledge. The model in a simplified way represents how the perturbations interrupt the flow of activations in the modeled pathway. The perturbations are required to turn the targeted regulator down (i.e., we do not model over-expressions). To relax these constraints the approach could be adapted to incorporate logical models introduced in chapter 3, formalizing a broad range of signaling relations and allowing all possible perturbation experiments. Such extension would require distinguishing the experiments affecting a given regulator into two classes: one of experiments which down-regulate, and one of experiments which up-regulate the target genes of this regulator. Logical models represent deterministic knowledge. To incorporate a probabilistic formalism in our approach, a Bayesian network model could be applied instead.

Note that, unlike numerous approaches inferring gene regulation from expression data [86, 85], we do not measure the activity of the regulators from their expression levels. Concluding activity from expression has several drawbacks. First, transcription factors are often expressed on low mRNA levels and thus detection of their activity profile based on expression measurements may fail due to noise in the data. Second, regulator activity is modulated in many ways on post-translational level of signaling, by phosphorylation, ligand binding, degradation, etc. Thus, we follow Gat-Viks and Shamir [41] and Szczurek *et al.* [117] and derive the regulator activity from a given model of signaling pathway. The regulators are treated as proteins, and their activity in a given perturbation experiment depends on the perturbation and on signaling relations which exist on post-translational level. Thus, assuming the input pathway topologies are correct, the pathway models should encompass all means of influencing the regulators present in the two cell populations, such as phosphorylation or ligand binding.

This dependence on the pathway models implies that the correctness of the models is critical for the correctness of our results. JODA may fail when the input pathway topologies are insufficient. To assure high quality of the pathway models, they should first be confronted with available data and corrected using refinement procedures (see, for example, refinement strategy introduced by Gat-Viks and Shamir [41]). Moreover, the remaining genes (not the regulators) are measured from their expression levels,

and their regulation is judged based on their transcriptional response to the perturbations of the regulators. The current view of regulation of gene expression in molecular biology [111] is more complex and includes, for example, post-transcriptional degradation by microRNAs. Ideally, our approach should integrate evidence of all means of gene regulation. We hope such integrative methods will be developed in the future.

Importantly, our analysis can still be performed without any input knowledge. This option is valuable particularly in non-model organisms or under unusual experimental circumstances, where not much more is available other than newly generated expression data. In case when signaling relations between the regulators are not known, the input topologies given to JODA should be fully disconnected graphs. This corresponds to inferring regulator-target gene relations for each regulator independently, only based on the perturbation data for this regulator. In case when no regulator-target gene relations are given, JODA evaluates probabilities of differential expression (see the first step of the algorithm above) using unsupervised, instead of partially supervised mixture modeling. However, as we show below, incorporation of knowledge greatly improves the quality of deregulation analysis. Therefore, even if only partial information is available either about the signaling pathways, or about the target genes, it is still beneficial to provide it as input to JODA.

Chapter 5

Conclusions and discussion

In this final chapter we summarize the improvements brought by the solutions proposed in this thesis. We contrast the simplifying assumptions made in the thesis with the biochemistry of cellular processes. We conclude by placing our approach in the general context of systems biology.

Advantages of our knowledge-based approach Methods proposed in this thesis bring several important improvements in different aspects of the solved problems.

In chapter 2 we develop a novel belief-based approach for partially supervised mixture modeling. Partially supervised modeling utilizes examples of observations that are labeled with their known mixture model component. The examples can be imprecise and are treated as probable rather than certain. The newly introduced belief-based modeling is compared to previously proposed soft-label modeling. The results favor application of the belief-based method in case when there are only a few examples and a large amount of unlabeled data. We show that the belief-based mixture estimation is less susceptible to bias in numbers of examples per cluster than the soft-label modeling. We demonstrate the advantage of both partially supervised methods over other mixture modeling variants which differ with respect to incorporating knowledge: from unsupervised and semi-supervised modeling to fully supervised modeling. In contrast to the semi- and fully supervised methods, partially supervised modeling handles even erroneous examples. Such examples may not fit with the rest of the data in their believed cluster and can get "re-clustered" in the output. In this way, partially supervised modeling tells which of the examples are incorrect according to the data.

We propose the application of both belief-based and soft-label methods to partially supervised differential expression analysis. The analysis utilizes given examples of genes that are believed to be differentially expressed. The examples guide model-based clustering to more precisely delineate genes that are down- or up-regulated from genes that are unchanged in one experimental condition compared to the other. Thus, the partially supervised approach itself decides on cut-off thresholds between expression measurements of these clusters of genes, which otherwise have to be set *ad-hoc*. Our results on synthetic and real data argue for the use of both belief-based

and soft-label methods in determining the differentially expressed genes. We show three applications of our partially supervised approach to gene expression data: First, we identify targets of Ste12 using knockout data in yeast, given knowledge from a Ste12 binding experiment. Second, we distinguish miR-1 from miR-124 targets based on expression data from transfection experiments, with the use of computationally-predicted targets of either microRNAs. Third, we improve the clustering of cell cycle gene time-course expression profiles by using our approach in a pre-processing step: data from each time point in a particular cycle phase is analyzed to find probability of up-regulation, given literature examples of genes active in this cycle phase.

In chapter 3 we present a new systems biology framework guiding the choice of perturbation experiments in research on a particular signaling pathway, investigating the regulation of the pathway's target genes. The framework iterates design of experiments and identification of regulatory relationships. To avoid ambiguity in the identification process, the experimental design algorithm MEED chooses the experiments that maximize diversity between expression profiles of genes regulated through different mechanisms. MEED takes advantage of qualitative knowledge about signaling relations in the pathway, formalized in a simple logical model. Basing on the model predictions, the algorithm has the ability to choose experiments without access to high-throughput experimental data. The novelty of MEED lies in considering potential dependencies between the suggested experiments. With this innovative feature, unlike extant approaches, MEED can design at once a set of informative, non-redundant experiments that can be efficiently performed together in a lab. MEED instructs the researcher about both environmental conditions and eventual perturbations to be performed on the pathway. The expansion procedure reconciles the model predictions with the data from the designed experiments to provide rich regulatory hypothesis: a set of identified target gene modules, their regulators in the pathway and their regulatory mechanisms.

These features of our framework make up a practical research procedure, which was extensively analyzed (both computationally and biologically) and applied to identify gene regulatory modules downstream of interconnected yeast MAPK pathways. Our results show that MEED can significantly reduce the amount of experimental work required to elucidate regulatory mechanisms downstream of a given pathway. Moreover, we demonstrated that even having a predefined set of perturbed molecules, an experimenter can significantly benefit from consulting MEED with regard to possible environmental stimulations and the type of genetic perturbations. Taken together, our approach opens the way to practical experimental design based on well-established qualitative biological knowledge.

Deregulation analysis of knockdown data with JODA (chapter 4) quantifies changes of gene regulation between two cell populations. The practical benefit from our approach is that ut keeps the deregulation analysis in the strict biological context of pathway-induced gene regulation in the cell populations under study. For each cell population separately, the approach incorporates given information about: (i) signaling topology active upstream of the studied regulatory relations and (ii) known

relations between TFs in the pathway and their target genes. To our knowledge, we are the first to consider that such prior information can be different for the compared cell populations. JODA proceeds in three steps. In the first step, we employ the partially supervised approach introduced in chapter 2 for differential expression analysis of the knockdown data. Here, the known target genes are utilized as examples of differential genes. In the second step, we use signaling pathway topologies, formalized in two simple models, one per each cell population. The models tell which of the knockdown experiments affect regulatory activity of a given regulator in the pathway. Given data from those experiments, we compute regulation scores summarizing the regulatory impact of each regulator on the target genes downstream of the pathway. In the third step, the regulatory signal from the two different cell populations is joined into one deregulation score by taking a difference of the regulation scores. Our results show advantage of JODA over investigating each cell population separately or without incorporation of prior knowledge.

In our analysis we focused on deregulation between healthy cells and cells with NCS-induced DNA double strand breaks. The obtained deregulation scores were further analyzed, first validating their congruence with the existing biological knowledge and next bringing new results. By finding functional clusters of the deregulated genes, we showed that the method assigns dominant deregulation scores to the genes playing important roles in the program of general response to DNA damage. Additionally, we investigated cooperativity between these deregulated genes, identifying known pathways and complexes in which the genes participate. We reviewed the DNA-damage related functionality of the genes with most extreme deregulation scores. Finally, we analyzed the indirect regulatory impact of the regulators in the ATM pathway on the genes in the functional clusters. An important advantage of our methodology is that it leads to testable mechanistic hypotheses. Here, we proposed a hierarchy of direct regulatory interactions by connecting the pathway to the deregulated DNA repair genes via the transcription factor CREB. Our analysis shows that JODA is a step forward to a systems level, mechanistic understanding of the of changes in gene regulation between different cellular environments.

Our simplifying assumptions versus cellular biochemistry In this work, we make several simplifying assumptions.

- *Known signaling pathway* Both the MEED framework (chapter 3) and the deregulation analysis (chapter 4) can fail when the input signaling pathway models are insufficient. The models are always treating signaling as a part isolated from the rest of the cell entity. In reality, there are many molecular interactions between both parts. To assure high quality of the input models, they should first be confronted with available data and corrected using refinement procedures (e.g., introduced by Gat-Viks and Shamir [41]).
- *Measurement errors* Errors in gene expression data may distort the partially supervised differential expression analysis proposed in chapter 2. The quality of the data influences reconstruction of regulatory relations both by the MEED frame-

work (chapter 3) and the deregulation analysis (chapter 4), where gene expression measurements are processed using POE ([40]; section 2.4) and the partially supervised approach, respectively.

- *Separation of time scales* Reasoning about gene regulation based on perturbation data is limited to the immediate gene expression response that is secondary to signaling. We rely on the assumption that the system state can be explored before transcriptional feedback mechanisms are activated and affect the pathway. Indeed, in our case studies, the signaling models do not include slower temporal processes such as feedback loops, and are integrated with expression profiles measured shortly after stimulation, during the immediate gene expression response.
- *Gene regulation* The current view of regulation of gene expression in molecular biology [77, 111] is more complex than the pure dependence of mRNA levels on transcription factor activity. Other means of regulation include, for example, post-transcriptional degradation by microRNAs. Ideally, our approaches should integrate measurements from high-throughput microRNA activity screens. Such data is not available now, but hopefully will be a subject of future experimental studies.

Setting in systems biology Systems biology paradigm states that the biological function of a given system can only emerge from interconnection of basic system components on multiple levels [2]. Bruggeman and Westerhoff [18] divide systems biology into two classes of either bottom-up or top-down approaches. The approach administrated in this thesis belongs rather to the top-down class, extended by the input of additional knowledge. In plain words, our approach can be described as "grasping a basic system component by placing it in a bigger picture". The basic component in our focus is gene regulation. The bigger picture is the cellular context of upstream signaling pathway and regulatory relations already established by previous studies (see section 1.1). Information about this context is collected from various sources, from literature reports about individual genes to high-throughput experimental measurements of gene expression or transcription factor-DNA binding.

Empowered by the additional sources of knowledge, our approach naturally yields increased biological specificity and robustness of our findings. Importantly, our thinking leads beyond standard applications of modeling in systems biology. The dominant holistic trend promotes building and improving elaborate models of the biological system *per se* (e.g. series of ever advancing models of the EGF MAPK pathway [106, 14, 73, 96, 98]). Such models are applied to provide experimentally testable predictions of system behavior and are refined if the predictions do not agree with the data [58, 70]. In contrast, our approach opens an opportunity to utilize such optimized models to infer *other* components of the biological system. For example, both in the MEED framework (chapter 3) and in the deregulation analysis (chapter 4) we expand a given signaling pathway model with downstream regulatory relations. Similarly, outcomes of numerous approaches for inferring regulatory networks [8, 86] may provide input knowledge for our partially supervised differential expression analysis (chapter 2).

Bibliography

[1] Akaike, H. (1973). Information Theory and an Extension of the Maximum Likelihood Principle. In *International Symposium on Information Theory, 2 nd, Tsahkadsor, Armenian SSR*, pages 267–281.

[2] Alberghina, L. and Westerhoff, H. (2007). *Systems Biology : Definitions and Perspectives (Topics in Current Genetics)*. Springer.

[3] Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. (2007). *Molecular Biology of the Cell*. Garland Science, 5 edition.

[4] Alexandridis, R., Lin, S., and Irwin, M. (2004). Class discovery and classification of tumor samples using mixture modeling of gene expression data - a unified approach. *Bioinformatics*, **20**(16), 2545–2552.

[5] Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. (2000). Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet*, **25**(1), 25–9.

[6] Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. (2007). Network motifs: theory and experimental approaches. *Nat Rev Genet*, **8**, 450–461.

[7] Baldi, P. and Long, A. D. (2001). A bayesian framework for the analysis of microarray expression data: regularized t -test and statistical inferences of gene changes. *Bioinformatics*, **17**(6), 509–519.

[8] Bansal, M., Belcastro, V., Ambesi-Impiombato, A., and di Bernardo, D. (2007). How to infer gene networks from expression profiles. *Mol Syst Biol*, **3**(NIL), 78.

[9] Bantounas, I., Phylactou, L. A., and Uney, J. B. (2004). RNA interference and the use of small interfering RNA to study gene function in mammalian systems. *J Mol Endocrinol*, **33**(3), 545–57.

[10] Barret, C. L. and Palsson, B. O. (2006). Iterative reconstruction of transcriptional regulatory networks: an algorythmic approach. *Plos Comput Biol*, **2**(5), e53.

[11] Bauer, S., Grossmann, S., Vingron, M., and Robinson, P. N. (2008). Ontologizer 2.0–a multifunctional tool for GO term enrichment analysis and data exploration. *Bioinformatics*, **24**(14), 1650–1.

[12] Berman, P., DasGupta, B., and Kao, M.-Y. (2005). Tight approximability results for test set problems in bioinformatics. *J. Comput. Syst. Sci.*, **71**(2), 145–162.

[13] Betel, D., Wilson, M., Gabow, A., Marks, D. S., and Sander, C. (2008). The microRNA.org resource: targets and expression. *Nucl. Acids Res.*, **36**, D149–D153.

[14] Bhalla, U. S. (2004). Models of cell signaling pathways. *Curr Opin Genet Dev*, **14**(4), 375–81.

[15] Bild, A. H., Yao, G., Chang, J. T., Wang, Q., Potti, A., Chasse, D., Joshi, M.-B., Harpole, D., Lancaster, J. M., Berchuck, A., Olson, J. A., Marks, J. R., Dressman, H. K., West, M., and Nevins, J. R. (2006). Oncogenic pathway signatures in human cancers as a guide to targeted therapies. *Nature*, **439**(7074), 353–357.

[16] Bishop, C. M. (2007). *Pattern Recognition and Machine Learning*. Springer.

[17] Boyle, E. I., Weng, S., Gollub, J., Jin, H., Botstein, D., Cherry, J. M., and Sherlock, G. (2004). GO::TermFinder–open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes. *Bioinformatics*, **20**(18), 3710–3715.

[18] Bruggeman, F. J. and Westerhoff, H. V. (2007). The nature of systems biology. *Trends Microbiol.*, **15**, 45–50.

[19] Buchler, N. E., Gerland, U., and Hwa, T. (2003). On schemes of combinatorial transcription logic. *Proc. Natl. Acad. Sci. U.S.A.*, **100**, 5136–5141.

[20] Cho, R., Campbell, M., Winzeler, E., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T., Gabrielian, A., Landsman, D., Lockhart, D., and Davis, R. (1998). A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, **2**(1), 65–73.

[21] Chua, G., Morris, Q. D., Sopko, R., Robinson, M. D., Ryan, O., Chan, E. T., Frey, B. J., Andrews, B. J., Boone, C., and Hughes, T. R. (2006). Identifying transcription factor functions and targets by phenotypic activation. *Proc Natl Acad Sci U S A*, **103**(32), 12045–50.

[22] Côme, E., Oukhellou, L., Denux, T., and Aknin, P. (2009). Learning from partially supervised data using mixture models and belief functions. *Pattern Recognition*, **42**(3), 334–348.

[23] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms, Second Edition*. The MIT Press.

[24] Costa, I. G., Krause, R., Opitz, L., and Schliep, A. (2007). Semi-supervised learning for the identification of syn-expressed genes from fused microarray and in situ image data. *BMC Bioinformatics*, **8 Suppl 10**, S3.

[25] Costa, I. G., Schönhuth, A., Hafemeister, C., and Schliep, A. (2009). Constrained mixture estimation for analysis and robust classification of clinical time series. *Bioinformatics*, **25**(12), i6–i14.

[26] Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. Wiley Series in Telecommunication.

[27] Cui, X. and Churchill, G. A. (2003). Statistical tests for differential expression in cDNA microarray experiments. *Genome Biol*, **4**(4), 210–210.

[28] de Jong, H. (2002). Modeling and simulation of genetic regulatory systems: a literature review. *J. Comput. Biol.*, **9**, 67–103.

[29] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, **39**(1), 1–38.

[30] Do, K.-A., Mủller, P., and Tang, F. (2005). A bayesian mixture model for differential gene expression. *Journal Of The Royal Statistical Society Series C*, **54**(3), 627–644.

[31] Dortet-Bernadet, J.-L. and Wicker, N. (2008). Model-based clustering on the unit sphere with an illustration using gene expression profiles. *Biostat*, **9**(1), 66–80.

[32] Elkon, R., Rashi-Elkeles, S., Lerenthal, Y., Linhart, C., Tenne, T., Amariglio, N., Rechavi, G., Shamir, R., and Shiloh, Y. (2005). Dissection of a DNA-damage-induced transcriptional network using a combination of microarrays, RNA interference and computational promoter analysis. *Genome Biol*, **6**(5), R43.

[33] Elkon, R., Vesterman, R., Amit, N., Ulitsky, I., Zohar, I., Weisz, M., Mass, G., Orlev, N., Sternberg, G., Blekhman, R., Assa, J., Shiloh, Y., and Shamir, R. (2008). SPIKE–a database, visualization and analysis tool of cellular signaling pathways. *BMC Bioinformatics*, **9**, 110.

[34] Ergun, A., Lawrence, C. A., Kohanski, M. A., Brennan, T. A., and Collins, J. J. (2007). A network biology approach to prostate cancer. *Mol Syst Biol*, **3**, 82.

[35] Even, S. (1979). *Graph algorithms*. Computer Science Press, Potomac, Maryland.

[36] Frohlich, H., Speer, N., Poustka, A., and Beissbarth, T. (2007). GOSim–an R-package for computation of information theoretic GO similarities between terms and gene products. *BMC Bioinformatics*, **8**, 166.

[37] Furge, K. A., Tan, M. H., Dykema, K., Kort, E., Stadler, W., Yao, X., Zhou, M., and Teh, B. T. (2007). Identification of deregulated oncogenic pathways in renal cell carcinoma: an integrated oncogenomic approach based on gene expression profiling. *Oncogene*, **26**(9), 1346–1350.

[38] Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman.

[39] Gari, E., Piedrafita, L., Aldea, M., and Herrero, E. (1997). A set of vectors with a tetracycline-regulatable promoter system for modulated gene expression in Saccharomyces cerevisiae. *Yeast*, **13**(9), 837–48.

[40] Garrett, E. S. and Parmigiani, G. (2003). POE: statistical methods for qualitative analysis of gene expression. In R. A. I. Giovanni Parmigiani, Elizabeth S. Garrett and S. L. Zeger, editors, *The Analysis of Gene Expression Data*, chapter 16, pages 362–387. Springer London.

[41] Gat-Viks, I. and Shamir, R. (2007). Refinement and expansion of signaling pathways: the osmotic response network in yeast. *Genome Res*, **17**(3), 358–367.

[42] Gat-Viks, I., Tanay, A., and Shamir, R. (2004). Modeling and analysis of heterogeneous regulation in biological networks. *Journal of Computational Biology*, **11**(6), 1034–1049.

[43] Ghosh, D. and Chinnaiyan, A. M. (2002). Mixture modelling of gene expression data from microarray experiments. *Bioinformatics*, **18**(2), 275–286.

[44] Glass, L. and Kauffman, S. A. (1973). The logical analysis of continuous, nonlinear biochemical control networks. *J. Theor. Biol.*, **39**, 103–129.

[45] Goeddel, D. V. and Chen, G. (2007). Tumor necrosis factor pathway. . *Science's STKE (Connections Map, as seen November 2007).*

[46] Grubbs, F. E. (1969). Procedures for detecting outlying observations in samples. *Technometrics*, **11**(1), 1–21.

[47] Hahn, J.-S., Hu, Z., Thiele, D. J., and Iyer, V. R. (2004). Genome-wide analysis of the biology of stress responses through heat shock transcription factor. *Mol Cell Biol*, **24**(12), 5249–56.

[48] Hannon, G. J. (2002). RNA interference. *Nature*, **418**(6894), 244–51.

[49] Harbison, C. T., Gordon, D. B., Lee, T. I., Rinaldi, N. J., Macisaac, K. D., Danford, T. W., Hannett, N. M., Tagne, J.-B., Reynolds, D. B., Yoo, J., Jennings, E. G., Zeitlinger, J., Pokholok, D. K., Kellis, M., Rolfe, P. A., Takusagawa, K. T., Lander, E. S., Gifford, D. K., Fraenkel, E., and Young, R. A. (2004). Transcriptional regulatory code of a eukaryotic genome. *Nature*, **431**(7004), 99–104.

[50] Herskowitz, I. (1995). MAP kinase pathways in yeast: for mating and more. *Cell*, **80**(2), 187–197.

[51] Hoeijmakers, J. H. J. (2009). DNA damage, aging, and cancer. *N Engl J Med*, **361**(15), 1475–85.

[52] Hohmann, S. (2002). Osmotic stress signaling and osmoadaptation in yeasts. *Microbiol Mol Biol Rev*, **66**(2), 300–72.

[53] Horvath, M. M., Wang, X., Resnick, M. A., and Bell, D. A. (2007). Divergent evolution of human p53 binding sites: cell cycle versus apoptosis. *PLoS Genet*, **3**(7), e127.

[54] Huang, E., Ishida, S., Pittman, J., Dressman, H., Bild, A., Kloos, M., D'Amico, M., Pestell, R. G., West, M., and Nevins, J. R. (2003). Gene expression phenotypic models that predict the activity of oncogenic pathways. *Nat Genet*, **34**(2), 226–230.

[55] Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of classification*, **2**(1), 193–218.

[56] Hudson, N. J., Reverter, A., and Dalrymple, B. P. (2009). A differential wiring analysis of expression data correctly identifies the gene containing the causal mutation. *PLoS Comput Biol*, **5**(5), e1000382.

[57] Hughes, T. R., Marton, M. J., Jones, A. R., Roberts, C. J., Stoughton, R., Armour, C. D., Bennett, H. A., Coffey, E., Dai, H., He, Y. D., Kidd, M. J., King, A. M., Meyer, M. R., Slade, D., Lum, P. Y., Stepaniants, S. B., Shoemaker, D. D., Gachotte, D., Chakraburtty, K., Simon, J., Bard, M., and Friend, S. H. (2000). Functional discovery via a compendium of expression profiles. *Cell*, **102**(1), 109–26.

[58] Ideker, T., Galitski, T., and Hood, L. (2001). A new approach to decoding life: systems biology. *Annu Rev Genomics Hum Genet*, **2**, 343–72.

[59] Ideker, T. E., Thorsson, V., and Karp, R. M. (2000). Discovery of regulatory interactions through perturbation: Inference and experimental design. In *In Pacific Symposium on Biocomputing 5*, pages 302–313.

[60] Irizarry, R. A., Bolstad, B. M., Collin, F., Cope, L. M., Hobbs, B., and Speed, T. P. (2003). Summaries of Affymetrix GeneChip probe level data. *Nucleic Acids Res*, **31**(4), e15.

[61] Jarque, C. M. and Bera, A. K. (1980). Efficient tests for normality, homoscedasticity and serial independence of regression residuals. *Economics Letters*, **6**(3), 255–259.

[62] Jen, K.-Y. and Cheung, V. G. (2005). Identification of novel p53 target genes in ionizing radiation response. *Cancer Res*, **65**(17), 7666–73.

[63] Karginov, F. V., Conaco, C., Xuan, Z., Schmidt, B. H., Parker, J. S., Mandel, G., and Hannon, G. J. (2007). A biochemical approach to identifying microRNA targets. *Proceedings of the National Academy of Sciences*, **104**(49), 19291–19296.

[64] Karp, R. M. (1972). Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press.

[65] Karp, R. M., Stoughton, R., and Yeung, K. Y. (1999). Algorithms for choosing differential gene expression experiments. In *RECOMB '99: Proceedings of the third annual international conference on Computational molecular biology*, pages 208–217, New York, NY, USA. ACM.

[66] Kauffman, S. (1969). Homeostasis and differentiation in random genetic control networks. *Nature*, **224**, 177–178.

[67] Kauffmann, A., Gentleman, R., and Huber, W. (2009). arrayQualityMetrics–a bioconductor package for quality assessment of microarray data. *Bioinformatics*, **25**(3), 415–6.

[68] Kielbasa, S., Klein, H., Roider, H., Vingron, M., and Blüthgen, N. (2010). Transfind–predicting transcriptional regulators for gene sets. *Nucleic Acids Research*, **38**(Web Server issue), W275–W280.

[69] Kirkman-Correia, C., Stroke, I. L., and Fields, S. (1993). Functional domains of the yeast STE12 protein, a pheromone-responsive transcriptional activator. *Mol Cell Biol*, **13**(6), 3765–72.

[70] Kitano, H. (2002). Systems biology: a brief overview. *Science*, **295**(5560), 1662–4.

[71] Klau, G. W., Rahmann, S., Schliep, A., Vingron, M., and Reinert, K. (2004). Optimal robust non-unique probe selection using Integer Linear Programming. *Bioinformatics*, **20 Suppl 1**(NIL), i186–93.

[72] Klau, G. W., Rahmann, S., Schliep, A., Vingron, M., and Reinert, K. (2007). Integer linear programming approaches for non-unique probe selection. *Discrete Appl. Math.*, **155**(6-7), 840–856.

[73] Kolch, W. (2005). Coordinating ERK/MAPK signalling through scaffolds and inhibitors. *Nat Rev Mol Cell Biol*, **6**(11), 827–37.

[74] Konwar, K. M., Mandoiu, I. I., Russell, A., and Shvartsman, A. A. (2005). Improved algorithms for multiplex pcr primer set selection with amplification length constraints. In *APBC*, pages 41–50.

[75] Krek, A., Grün, D., Poy, M. N., Wolf, R., Rosenberg, L., Epstein, E. J., MacMenamin, P., da Piedade, I., Gunsalus, K. C., Stoffel, M., and Rajewsky, N. (2005). Combinatorial microRNA target predictions. *Nat Genet*, **37**(5), 495–500.

[76] Kurz, E. U. and Lees-Miller, S. P. (2004). DNA damage-induced activation of ATM and ATM-dependent signaling pathways. *DNA Repair (Amst.)*, **3**, 889–900.

[77] Latchman, D. (2005). *Gene Regulation - A Eukaryotic Perspective (Advanced Texts)*. Taylor & Francis, 5 edition.

[78] Lavin, M. F. and Kozlov, S. (2007). ATM activation and DNA damage response. *Cell Cycle*, **6**(8), 931–42.

[79] Lilliefors, H. (1967). On the kolmogorovsmirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, **62**, 399–402.

[80] Lim, C.-A., Yao, F., Wong, J. J.-Y., George, J., Xu, H., Chiu, K. P., Sung, W.-K., Lipovich, L., Vega, V. B., Chen, J., Shahab, A., Zhao, X. D., Hibberd, M., Wei, C.-L., Lim, B., Ng, H.-H., Ruan, Y., and Chin, K.-C. (2007). Genome-wide mapping of RELA(p65) binding identifies E2F1 as a transcriptional activator recruited by NF-kappaB upon TLR4 activation. *Mol Cell*, **27**(4), 622–35.

[81] Lim, L. P., Lau, N. C., Garrett-Engele, P., Grimson, A., Schelter, J. M., Castle, J., Bartel, D. P., Linsley, P. S., and Johnson, J. M. (2005). Microarray analysis shows that some microRNAs downregulate large numbers of target mRNAs. *Nature*, **433**(7027), 769–773.

[82] Linhart, C., Halperin, Y., and Shamir, R. (2008). Transcription factor and microRNA motif discovery: the Amadeus platform and a compendium of metazoan target sets. *Genome Res*, **18**(7), 1180–9.

[83] Liu, Y. and Ringnér, M. (2007). Revealing signaling pathway deregulation by using gene expression signatures and regulatory motif analysis. *Genome Biol*, **8**(5).

[84] Mani, K. M., Lefebvre, C., Wang, K., Lim, W. K., Basso, K., Dalla-Favera, R., and Califano, A. (2008). A systems biology approach to prediction of oncogenes and molecular perturbation targets in B-cell lymphomas. *Mol Syst Biol*, **4**, 169.

[85] Markowetz, F. (2010). How to understand the cell by breaking it: network analysis of gene perturbation screens. *PLoS Comput Biol*, **6**(2), e1000655.

[86] Markowetz, F. and Spang, R. (2007). Inferring cellular networks–a review. *BMC Bioinformatics*, **8 Suppl 6**, S5.

[87] Martinez, R., Pasquier, C., and Pasquier, N. (2007). GenMiner: Mining Informative Association Rules from Genomic Data. In *BIBM '07: Proceedings of the 2007 IEEE International Conference on Bioinformatics and Biomedicine*, pages 15–22, Los Alamitos, CA, USA. IEEE Computer Society.

[88] Maslov, S. and Sneppen, K. (2002). Specificity and stability in topology of protein networks. *Science*, **296**(5569), 910–913.

[89] McLachlan, G. J. and Peel, D. (2000). *Finite Mixture Models*. Wiley Series in Probability and Statistics. Wiley-Interscience.

[90] McLachlan, G. J., Bean, R. W., and Peel, D. (2002). A mixture model-based approach to the clustering of microarray expression data . *Bioinformatics*, **18**(3), 413–422.

[91] Miyagishi, M. and Taira, K. (2002). U6 promoter-driven siRNAs with four uridine 3' overhangs efficiently suppress targeted gene expression in mammalian cells. *Nat Biotechnol*, **20**(5), 497–500.

[92] Mnaimneh, S., Davierwala, A. P., Haynes, J., Moffat, J., Peng, W.-T., Zhang, W., Yang, X., Pootoolal, J., Chua, G., Lopez, A., Trochesset, M., Morse, D., Krogan, N. J., Hiley, S. L., Li, Z., Morris, Q., Grigull, J., Mitsakakis, N., Roberts, C. J., Greenblatt, J. F., Boone, C., Kaiser, C. A., Andrews, B. J., and Hughes, T. R. (2004). Exploration of essential gene functions via titratable promoter alleles. *Cell*, **118**(1), 31–44.

[93] Nemhauser, G. L. and Wolsey, L. A. (1988). *Integer and combinatorial optimization*. Wiley-Interscience, New York, NY, USA.

[94] Nern, A. and Arkowitz, R. A. (1999). A Cdc24p-Far1p-Gbetagamma protein complex required for yeast orientation during mating. *J Cell Biol*, **144**(6), 1187–202.

[95] Newton, M. A., Noueiry, A., Sarkar, D., and Ahlquist, P. (2004). Detecting differential gene expression with a semiparametric hierarchical mixture method. *Biostat*, **5**(2), 155–176.

[96] Oda, K., Matsuoka, Y., Funahashi, A., and Kitano, H. (2005). A comprehensive pathway map of epidermal growth factor receptor signaling. *Mol Syst Biol*, **1**(NIL), 2005.0010.

[97] O'Rourke, S. M. and Herskowitz, I. (2004). Unique and redundant roles for HOG MAPK pathway components as revealed by whole-genome expression analysis. *Mol Biol Cell*, **15**(2), 532–42.

[98] Orton, R. J., Sturm, O. E., Vyshemirsky, V., Calder, M., Gilbert, D. R., and Kolch, W. (2005). Computational modelling of the receptor-tyrosine-kinase-activated MAPK pathway. *Biochem J*, **392**(Pt 2), 249–61.

[99] Pan, W. (2006). Incorporating gene functions as priors in model-based clustering of microarray gene expression data. *Bioinformatics*, **22**(7), 795–801.

[100] Pan, W., Lin, J., and Le, C. (2002). Model-based cluster analysis of microarray gene-expression data. *Genome Biology*, **3**(2), research0009.1–research0009.8.

[101] Pan, W., Shen, X., Jiang, A., and Hebbel, R. P. (2006). Semi-supervised learning via penalized mixture model with application to microarray sample classification. *Bioinformatics*, **22**(19), 2388–2395.

[102] Papadimitriou, C. H. and Steiglitz, K. (1982). *Combinatorial optimization: algorithms and complexity*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[103] Pokholok, D. K., Zeitlinger, J., Hannett, N. M., Reynolds, D. B., and Young, R. A. (2006). Activated signal transduction kinases frequently occupy target genes. *Science*, **313**(5786), 533–6.

[104] Roberts, C. J., Nelson, B., Marton, M. J., Stoughton, R., Meyer, M. R., Bennett, H. A., He, Y. D., Dai, H., Walker, W. L., Hughes, T. R., Tyers, M., Boone, C., and Friend, S. H. (2000). Signaling and Circuitry of Multiple MAPK Pathways Revealed by a Matrix of Global Gene Expression Profiles. *Science*, **287**(5454), 873–880.

[105] Schlicker, A., Domingues, F. S., Rahnenfuhrer, J., and Lengauer, T. (2006). A new measure for functional similarity of gene products based on Gene Ontology. *BMC Bioinformatics*, **7**, 302.

[106] Schoeberl, B., Eichler-Jonsson, C., Gilles, E. D., and Mller, G. (2002). Computational modeling of the dynamics of the MAP kinase cascade activated by surface and internalized EGF receptors. *Nature Biotechnology*, **20**, 370 – 375.

[107] Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics*, **6**(2), 461–464.

[108] Selbach, M., Schwanhausser, B., Thierfelder, N., Fang, Z., Khanin, R., and Rajewsky, N. (2008). Widespread changes in protein synthesis induced by microRNAs. *Nature*, **455**(7209), 58–63.

[109] Seymour, P. D. (1995). Packing directed circuits fractionally. *Combinatorica*, **15**, 281–288.

[110] Shafer, G. (1976). *A mathematical theory of evidence*. Princeton university press.

[111] Shafer, G., editor (2008). *Post-transcriptional gene regulation*. Humana press.

[112] Shiloh, Y. (2006). The ATM-mediated DNA-damage response: taking shape. *Trends Biochem Sci*, **31**(7), 402–10.

[113] Slonim, D. K. and Yanai, I. (2002). From patterns to pathways: gene expression data analysis comes of age. *Nat Genet*, **32**, Suppl502–508.

[114] Slonim, D. K. and Yanai, I. (2009). Getting started in gene expression microarray analysis. *PLoS Computational Biology*, **5**(10), e1000543+.

[115] Smyth, G. K. (2005). Limma: linear models for microarray data. In R. Gentleman, V. Carey, S. Dudoit, and W. H. R. Irizarry, editors, *Bioinformatics and Computational Biology Solutions using R and Bioconductor*, pages 397–420. Springer, New York.

[116] Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., and Mesirov, J. P. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci U S A*, **102**(43), 15545–15550.

[117] Szczurek, E., Gat-Viks, I., Tiuryn, J., and Vingron, M. (2009). Elucidating regulatory mechanisms downstream of a signaling pathway using informative experiments. *Molecular Systems Biology*, **5**.

[118] Szczurek, E., Biecek, P., Tiuryn, J., and Vingron, M. (2010). Introducing knowledge into differential expression analysis. *Journal of Computational Biology*, **17**(8), 953–967.

[119] Taneja, I. (2001). *Generalized Information Measures and Their Applications*. http://www.mtm.ufsc.br/∼taneja/book/book.html.

[120] Taylor, I. W., Linding, R., Warde-Farley, D., Liu, Y., Pesquita, C., Faria, D., Bull, S., Pawson, T., Morris, Q., and Wrana, J. L. (2009). Dynamic modularity in protein interaction networks predicts breast cancer outcome. *Nat Biotechnol*, **27**(2), 199–204.

[121] Tegnér, J. and Björkegren, J. (2007). Perturbations to uncover gene networks. *Trends Genet*, **23**(1), 34–41.

[122] Tusher, V. G., Tibshirani, R., and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proc Natl Acad Sci U S A*, **98**(9), 5116–21.

[123] Vastrik, I., D'Eustachio, P., Schmidt, E., Joshi-Tope, G., Gopinath, G., Croft, D., de Bono, B., Gillespie, M., Jassal, B., Lewis, S., Matthews, L., Wu, G., Birney, E., and Stein, L. (2007). Reactome: a knowledge base of biologic pathways and processes. *Genome Biol*, **8**(3), R39.

[124] Vatcheva, I., de Jong, H., Bernard, O., and Mars, N. J. I. (2006). Experiment selection for the discrimination of semi-quantitative models of dynamical systems. *Artif. Intell.*, **170**(4-5), 472–506.

[125] Wang, X. (2008). miRDB: a microRNA target prediction and functional annotation database with a wiki interface. *RNA (New York, N.Y.)*, **14**(6), 1012–1017.

[126] Wang, X. and El (2007). Prediction of both conserved and nonconserved microRNA targets in animals. *Bioinformatics*, **24**(3), 325–332.

[127] Wang, X. and Wang, X. (2006). Systematic identification of microRNA functions by combining target prediction and expression profiling. *Nucleic Acids Res*, **34**(5), 1646–52.

[128] Watters, J. W. and Roberts, C. J. (2006). Developing gene expression signatures of pathway deregulation in tumors. *Mol Cancer Ther*, **5**(10), 2444–9.

[129] Wei, C.-L., Wu, Q., Vega, V. B., Chiu, K. P., Ng, P., Zhang, T., Shahab, A., Yong, H. C., Fu, Y., Weng, Z., Liu, J., Zhao, X. D., Chew, J.-L., Lee, Y. L., Kuznetsov, V. A., Sung, W.-K., Miller, L. D., Lim, B., Liu, E. T., Yu, Q., Ng, H.-H., and Ruan, Y. (2006). A global map of p53 transcription-factor binding sites in the human genome. *Cell*, **124**(1), 207–19.

[130] Wingender, E., Dietze, P., Karas, H., and Knppel, R. (1996). Transfac: a database on transcription factors and their dna binding sites. *Nucleic Acids Res*, **24**(1), 238–241.

[131] Winzeler, E. A., Shoemaker, D. D., Astromoff, A., Liang, H., Anderson, K., Andre, B., Bangham, R., Benito, R., Boeke, J. D., Bussey, H., Chu, A. M., Connelly, C., Davis, K., Dietrich, F., Dow, S. W., Bakkoury, M. E., Foury, F., Friend, S. H., Gentalen, E., Giaever, G., Hegemann, J. H., Jones, T., Laub, M., Liao, H., Liebundguth, N., Lockhart, D. J., Lucau-Danila, A., Lussier, M., M'Rabet, N., Menard, P., Mittmann, M., Pai, C., Rebischung, C., Revuelta, J. L., Riles, L., Roberts, C. J., Ross-MacDonald, P., Scherens, B., Snyder, M., Sookhai-Mahadeo, S., Storms, R. K., Veronneau, S., Voet, M., Volckaert, G., Ward, T. R., Wysocki, R., Yen, G. S., Yu, K., Zimmermann, K., Philippsen, P., Johnston, M., and Davis, R. W. (1999). Functional characterization of the S. cerevisiae genome by gene deletion and parallel analysis. *Science*, **285**(5429), 901–6.

[132] Wolsey, L. A. (1998). *Integer Programming*. Wiley-Interscience, 1 edition.

[133] Wood, R. D., Mitchell, M., and Lindahl, T. (2005). Human dna repair genes, 2005. *Mutat Res*, **577**(1-2), 275–283.

[134] Workman, C. T., Mak, H. C., McCuine, S., Tagne, J.-B., Agarwal, M., Ozier, O., Begley, T. J., Samson, L. D., and Ideker, T. (2006). A systems approach to mapping dna damage response pathways. *Science*, **312**(5776), 1054–1059.

[135] Yeang, C.-H. and Jaakkola, T. (2006). Modeling the combinatorial functions of multiple transcription factors. *J Comput Biol*, **13**(2), 463–80.

[136] Yeang, C.-H., Mak, H. C., McCuine, S., Workman, C., Jaakkola, T., and Ideker, T. (2005). Validation and refinement of gene-regulatory pathways on a network of physical interactions. *Genome Biol*, **6**(7), R62.

[137] Yeung, K. Y., Fraley, C., Murua, A., Murua, R., Raftery, A. E., and Ruzzo, W. L. (2001). Model-based clustering and data transformations for gene expression data. *Bioinformatics*, **17**(10), 977–987.

[138] Yoshimoto, H., Saltsman, K., Gasch, A. P., Li, H. X., Ogawa, N., Botstein, D., Brown, P. O., and Cyert, M. S. (2002). Genome-wide analysis of gene expression regulated by the calcineurin/Crz1p signaling pathway in Saccharomyces cerevisiae. *J Biol Chem*, **277**(34), 31079–88.

[139] Zeitlinger, J., Simon, I., Harbison, C. T., Hannett, N. M., Volkert, T. L., Fink, G. R., and Young, R. A. (2003). Program-specific distribution of a transcription factor dependent on partner transcription factor and MAPK signaling. *Cell*, **113**(3), 395–404.

[140] Zhang, X., Odom, D. T., Koo, S.-H., Conkright, M. D., Canettieri, G., Best, J., Chen, H., Jenner, R., Herbolsheimer, E., Jacobsen, E., Kadam, S., Ecker, J. R., Emerson, B., Hogenesch, J. B., Unterman, T., Young, R. A., and Montminy, M. (2005). Genome-wide analysis of cAMP-response element binding protein occupancy, phosphorylation, and target gene activation in human tissues. *Proc Natl Acad Sci U S A*, **102**(12), 4459–4464.

[141] Zhao, Y., Samal, E., and Srivastava, D. (2005). Serum response factor regulates a muscle-specific microRNA that targets Hand2 during cardiogenesis. *Nature*, **436**(7048), 214–20.

[142] Zhu, H., Klemic, J. F., Chang, S., Bertone, P., Casamayor, A., Klemic, K. G., Smith, D., Gerstein, M., Reed, M. A., and Snyder, M. (2000). Analysis of yeast protein kinases using protein chips. *Nat Genet*, **26**(3), 283–9.

[143] Zhu, H., Bilgin, M., Bangham, R., Hall, D., Casamayor, A., Bertone, P., Lan, N., Jansen, R., Bidlingmaier, S., Houfek, T., Mitchell, T., Miller, P., Dean, R. A., Gerstein, M., and Snyder, M. (2001). Global analysis of protein activities using proteome chips. *Science*, **293**(5537), 2101–5.

[144] Zhu, X. and Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, **3**(1), 1–130.

Appendix Figures

f_{PKA}

EOC	PKA
-1	1
0	1
1	-1

f_{Sho1}

EOC	Sho1
-1	-1
0	-1
1	1

$f_{Sln1/Ypd1}$

EOC	SLN1 /Ypd1
-1	1
0	0
1	-1

f_{Ste11}

Ste20	Ste11
-1	-1
0	0
1	1

f_{Pbs2}

min (HogScaffold , Ste11)	Ssk1 /2/22	PKA	Pbs2
-1	-1	-1	1
0	-1	-1	1
1	-1	-1	1
-1	0	-1	1
0	0	-1	1
1	0	-1	1
-1	1	-1	1
0	1	-1	1
1	1	-1	1
-1	-1	0	-1
0	-1	0	0
1	-1	0	1
-1	0	0	0
0	0	0	0
1	0	0	1
-1	1	0	0
0	1	0	0
1	1	0	1
-1	-1	1	-1
0	-1	1	0
1	-1	1	1
-1	0	1	-1
0	0	1	0
1	0	1	1
-1	1	1	0
0	1	1	0
1	1	1	1

f_{Sko1}

PKA	Sko1
-1	-1
0	0
1	1

f_{Hog1}

Pbs2	Hog1
-1	-1
0	0
1	1

$f_{Fus3/Kss1}$

Ste7	Fus3 /Kss1
1	-1
0	0
1	1

f_{Ste12}

Fus3/ Kss1	Ste12
-1	-1
0	0
1	1

f_{Msn1}

Hog1	Msn1
-1	-1
0	0
1	1

f_{Hot1}

Hog1	Hot1
-1	-1
0	0
1	1

$f_{Ssk1/2/22}$

Ypd1	Ssk1 /2/22
-1	1
0	0
1	-1

$f_{HogScaffold}$

Sho1	Hog Scaffold
-1	-1
0	0
1	1

$f_{Ssk1/2/22}$

Phero mone	Sho1	Ste20
-1	-1	-1
0	-1	0
1	-1	1
-1	0	0
0	0	0
1	0	1
-1	1	1
0	1	1
1	1	1

$f_{Msn2/4}$

PKA	Hog 0	Msn 2/4
-1	-1	1
0	-1	0
1	-1	-1
1	0	1
0	0	0
1	0	0
-1	1	1
0	1	1
1	1	1

f_{Ste7}

Ste11	Hog1	Ste7
-1	-1	-1
0	-1	0
1	-1	1
-1	0	-1
0	0	0
1	0	0
-1	1	-1
0	1	-1
1	1	-1

Figure .1: Regulatory functions in the yeast model. Regulation functions determine the state of each variable (output, in red) given the states of its regulators (input, in black). EOC – Environmental Osmotic Concentration.

#	Performed Experiment		Modeled experiment		Experimental Study	ED***			
	Stimulation****	Perturbation	Stimulation	Perturbation		MEED	M-TOPOL	Barret and Pallson	Ideker *et al.*
1	Control	STE11 KO*	EOC:0 Pheromone:0	Ste11:0	(Roberts *et al.*, 2000)				
2	Control	SLN1 KO	EOC:0 Pheromone:0	Sln1Ypd1:0	(Mnaimneh *et al.*, 2004)	5	4		6
3	Control	HOG1 KO	EOC:0 Pheromone:0	Hog1:0	(Hahn *et al.*, 2004)				
4	Control	STE7 KO	EOC:0 Pheromone:0	Ste7:0	(Roberts *et al.*, 2000)		8		
5	Control	KSS1 KO	EOC:0 Pheromone:0	Kss1/Fus3:0	(Roberts *et al.*, 2000)				
6	Control	STE12 KO	EOC:0 Pheromone:0	Ste12:0	(Roberts *et al.*, 2000)				
7	Control	STE12 OE	EOC:0 Pheromone:0	Ste12:2	(Chua *et al.*, 2006)	9	10	6	
8	Control	MSN1 KO	EOC:0 Pheromone:0	Msn1:0	(Hahn *et al.*, 2004)		11		
9	Control	SKO1 KO	EOC:0 Pheromone:0	Sko1:0	(Hahn *et al.*, 2004)	2	12		
10	Control	MSN2 KO	EOC:0 Pheromone:0	Msn2Msn4:0	(Hahn *et al.*, 2004)				
11	Control	MSN2 OE**	EOC:0 Pheromone:0	Msn2Msn4:2	(Chua *et al.*, 2006)	11	13	11	7
12	50aF in 30min	-	EOC:0 **Pheromone:2**	-	(Roberts *et al.*, 2000)				
13	50aF in 30min	STE20 KO	EOC:0 **Pheromone:2**	Ste20:0	(Roberts *et al.*, 2000)		2	9	
14	50aF in 30min	FUS3D KO	EOC:0 **Pheromone:2**	Kss1/Fus3:0	(Roberts *et al.*, 2000)	4	9		
15	50aF in 30min	STE12 KO	EOC:0 **Pheromone:2**	Ste12:0	(Roberts *et al.*, 2000)			5	
16	0.125M KCl in 20min	-	**EOC:1** Pheromone:0	-	(O'Rourke and Herskowitz, 2004)				
17	0.125M KCl in 20min	STE11 KO	**EOC:1** Pheromone:0	Ste11:0	(O'Rourke and Herskowitz, 2004)				
18	0.125M KCl in 20min	SSK1 KO	**EOC:1** Pheromone:0	Ssk1/2/22:0	(O'Rourke and Herskowitz, 2004)			10	
19	0.125M KCl in 20min	HOG1 KO	**EOC:1** Pheromone:0	Hog1:0	(O'Rourke and Herskowitz, 2004)				
20	0.5M KCl in 40min	-	**EOC:2** Pheromone:0	-	(O'Rourke and Herskowitz, 2004)	1			
21	0.5M KCl in 40min	SHO1 KO	**EOC:2** Pheromone:0	Sho1:0	(O'Rourke and Herskowitz, 2004)	8	1		
22	0.5M KCl in 40min	STE11 KO	**EOC:2** Pheromone:0	Ste11:0	(O'Rourke and Herskowitz, 2004)	10	3	8	
23	0.5M KCl in 40min	SSK1 KO	**EOC:2** Pheromone:0	Ssk1/2/22:0	(O'Rourke and Herskowitz, 2004)	7	5		
24	0.5M KCl in 40min	PBS2 KO	**EOC:2** Pheromone:0	Pbs2:0	(O'Rourke and Herskowitz, 2004)	3	6		
25	0.5M KCl in 40min	HOG1 KO	**EOC:2** Pheromone:0	Hog1:0	(O'Rourke and Herskowitz, 2004)	6	7	7	5

Figure .2: 25 candidate experiments on the yeast model. *KO – knock-out; **OE – over-expression; ***The numbers indicate the order in which the experiments were chosen by each method; ****The time point between 20 and 40min is the peak immediate gene expression response to the pathway.

Notation and Definitions

All chapters

BGMM	belief-based Gaussian mixture modeling
MEED	model expansion experimental design
JODA	joint deregulation analysis
FDR	false discovery rate
POE	Probability Of Expression
GO	Gene Onthology

Chapter 1

DNA	deoxyribonucleic acid
RNA	ribonucleic acid
mRNA	messenger RNA
TF	transcription factor
PCR	polymerase chain reaction
RNAi	RNA interference
dsRNA	double-stranded RNA
siRNA	short interfering RNA
shRNA	small hairpin RNA
RISC	RNA-induced silencing complex

Chapter 2

$X = \{x_1, ..., x_N\}$	dataset of N observations
$\mathcal{Y} = \{1, ..., K\}$	set of K clusters
k	cluster or model component, $k \in \mathcal{Y}$
M	number of examples
P	probability of an event
$Y_1, ..., Y_N$	random variables with values in $\mathcal{Y}$
π_k	mixing proportion or prior of the mixture component k
$M(1, \pi_1, ..., \pi_K)$	multinomial distribution with probabilities $\pi_1, ..., \pi_K$
$\pi = \{\pi_1, ..., \pi_K\}$	set of mixing proportions for all components
$\mathcal{R}$	set of real numbers
$X_1, ..., X_N$	random variables with values in $\mathcal{R}$
$f(x_i \mid Y_i = k)$	density function of the variable X_i given the value of variable Y_i
θ_k	parameters of the density function f (in chapter 2 Gaussian with mean μ_k and variance σ_k^2)
$\theta = \{\theta_1, ..., \theta_K\}$	set of Gaussian parameters for all components
$\Psi = \{\pi, \theta\}$	all parameters of a mixture model
Ψ_K	all parameters of a mixture model with K components
ϵ	minimal accepted change in log likelihood in subsequent iterations (stop criterion)
Q	maximal number of iterations (stop criterion)
q	iteration number
X^s	input dataset in the fully supervised modeling
z_i	indicator function, for observation x_i, its cluster label y_i and cluster k returns value 1 if $y_i = k$, and 0 otherwise.
z_{ik}	value of function z_i for cluster k
$l(\cdot)$	log likelihood
t_{ik}	posterior probability for observation x_i to belong to cluster k
X^s	input dataset in fully supervised modeling
X^{ss}	input dataset in semi-supervised modeling
X^p	input dataset in soft-label modeling
X^b	input dataset in belief-based modeling
$b_{ik} = P(Y_i = k)$	belief for observation x_i and cluster k
$b = \{b_1, ..., b_K\}$	set of all beliefs
$p = \{p_1, ..., p_K\}$	set of all plausibilities
$\mathcal{U}(\cdot, \cdot)$	Uniform distribution
$\mathcal{G}(\cdot, \cdot)$	Gaussian distribution
miRNA	microRNA
differential	differentially expressed
EM	Expectation Maximization
MAP	maximum a posteriori
NorDi	Normal Discretization
AIC	Akaike Information Criterion
BIC	Bayesian Information Criterion

Chapter 3

$\mathcal{M}$	logical model
V	model variables
$U = \{1, ..., k\}$...	discrete states that the model variables may attain
$Pa(v)$	parents of a model variable $v \in V$
f_v	regulation function, defines state of $v \in V$ given its parents' states
$\mathcal{F}$	set of regulation functions
$G = (V, A)$	model graph with nodes in V and edges in A
I	model stimulators
P	model variables that are subject to perturbation
e	experiment
E	set of experiments
s	model state
s_e	model state predicted for experiment e
$s_e(v)$	e-predicted state of variable v
F	feedback set
e_S	observed partial state for experiment e
$D(s_e, e)$	discrepancy between experiment e and model state s_e
R	set of regulators
r	regulatory program
$T(e)$.....	partition induced by experiment e
$S(E)$	partition induced by set of experiments E
M	matching
$H(E)$	entropy score of experiment set E
$H(e\|E)$	entropy gain with adding experiment e to experiment set E
$\mathcal{P}$	set of partitions
$\mathcal{P}^+$.....	closure of a set of partitions $\mathcal{P}$ under finite intersections
$\bigcap \mathcal{P}$.....	intersection of all partitions from a set $\mathcal{P}$
T, T', T''.....	partition in $\mathcal{P}$
S, S', S''.....	partition in $\mathcal{P}^+$
$\{R\}$	full (one block) partition of a set R
id_R	identity partition of a set R into singletons
$T'' \leq T'$	partition T'' is included in T'
Φ	class of strictly increasing functions satisfying (A0) and (A1)
Δ_T.....	gain function determined by a function in Φ and a partition T
ρ	a function in Φ
ED.....	experimental design
FUP	fraction of undistinguished pairs
NP.....	nondeterministic polynomial time class of problem complexity
3DM.....	3-DIMENSIONAL MATCHING
ILP	integer linear programming
TNF	Tumor Necrosis Factor
EOC.....	environmental osmotic concentration

Chapter 4

t	cell population
$V = \{v_1, ..., v_n\}$	set of regulators
$G_t = (V, A_t)$	pathway topology in a cell population t with set of nodes V and directed edges A_t
$\Delta^t v$	perturbation of v in a given cell population t
E_t	set of all experiments perturbing regulators in V in cell population t
G_t^*	transitive reflexive closure of given pathway topology G_t
$\mathcal{M}_t$	model matrix
$E_{v,t}$	set of all perturbation experiments that affect regulator v in cell population t
h	population of healthy cells
d	population of damaged cells
$\mathbf{p}_v^t$	vector of signed differential expression probabilities of the genes for perturbation of v in cell population t
$\mathbf{R}_v^t$	vector of regulation scores of the genes for regulator v in cell population t
$\mathbf{D}_v$	vector of deregulation scores of the genes for regulator v
NCS	neocarzinostatin
deregulation list	list of deregulation scores
GSEA.....	Gene Set Enrichment Analysis
ORC	origin of replication complex

Zusammenfassung

Die vorliegende Doktorarbeit befasst sich mit der Aufklärung der Regulierung von Genexpression im Kontext von bekannten zellulären Signalwegen und regulierten Genen. Wir analysieren Daten von experimentellen Interventionen, die auf Signalkomponenten zielen. Solche Experimente verursachen Änderungen in der Genexpression der durch den Signalweg regulierten Genen. Die in dieser Doktorarbeit entwickelten Ansätze lösen verschiedene Probleme im Bereich der Kontext-spezifischen Genregulierung.

In Kapitel 2 entwickeln wir eine Methode zur differentiellen Expressionsanalyse der Interventionsdaten, die vorgegebene Beispiele differentieller Gene nutzt. Hochdurchsatz-Genexpressionsexperimente ermöglichen einen Vergleich zweier experimenteller Bedingungen. Die Messungen werden einer Analyse unterzogen, um die Gruppen von Genen zu bestimmen, die unter einer der Bedingungen hoch- oder herunterreguliert werden, oder deren Expression gleich bleibt. Mittels Expertenwissen können bestimmte Gene diesen verschiedenen Gruppen zuordnen werden. Zum Beispiel erwartet man, dass Gene, die von einem transkriptionellen Aktivator reguliert werden, nach dem Ausschalten dieses Aktivators herunterreguliert werden. Etablierte Methoden zur differentiellen Expressionsanalyse ignorieren solch unpräzise Beispiele, unsere schließt sie systematisch mit ein. Wir benutzen sogennante *partially supervised* Mischmodellierung, die eindimensionale Expressionsdaten in Gruppen von differentiell regulierten und unveränderten Genen aufteilt und dabei von unpräzisen Beispielen profitiert. Dieser Ansatz wird von zwei Methoden realisiert: einer neuen *belief-based* Mischmodellierung, die wir hier vorstellen, und der früher entwickelte *soft-label* Mischmodellierung. Tests zeigen, dass sowohl die *belief-based* als auch die *soft-label* Methode falsche Beispiele besser korrigieren als die *semi-supervised* Mischmodellierung. Wir vergleichen unsere *partially supervised* Methodik auch mit alternativen Ansätzen zur differentiellen Expressionsanalyse und zeigen, dass die Aufnahme von unpräzisem Wissen bessere Ergebnisse erzeugt. Wir präsentieren verschiedene Anwendungen der Methodik.

In Kapitel 3 befassen wir uns mit der Planung von Interventionsexperimenten für einen gegebenen Signalweg. Für die systematische Rekonstruktion der Genregulation durch einen Signalweg werden informative experimentelle Daten benötigt. Wir stellen einen allgemeinen Ansatz für diese Rekonstruktion vor. MEED, eine experimentelle Design-Komponente unseres Ansatzes, schlägt eine möglichst kleine Anzahl von gezielten Interventionsexperimenten in dem Signalweg vor. Um Mehrdeutigkeit in der Identifizierung der Regulierungsverhältnisse zu vermeiden, maximiert die Auswahl

der Experimente den Unterschied zwischen Expressionsprofilen von Genen, die durch verschiedene Mechanismen reguliert werden. Mittels eines prädiktiven logischen Modells bezieht dieser Ansatz auch Expertenwissen über die Signalwege mit ein. MEED berücksichtigt prognostizierte Abhängigkeiten zwischen Experimenten und kann so einen ganzen Satz Experimente vorschlagen, die gleichzeitig durchgeführt werden können. Wir wenden unseren Ansatz auf verbundene Signalwege in der Hefe *Saccharomyces cerevisiae* an. Im Vergleich zu anderen Methoden schlägt MEED die informativsten Experimente für unzweideutige Identifizierung von transkriptioneller Regulation in diesem System vor.

In Kapitel 4 stellen wir eine Anwendung zur Deregulationsanalyse vor, d.h., zum Vergleich von Änderungen in der Genregulierung zwischen zwei Zellpopulationen. Vorhandene Deregulationsstudien lassen verfügbares Wissen über den zellulären Kontext dieser Änderungen außer acht. Wir untersuchen Deregulation mittels zellpopulationsspezifische Interventionsdaten, und mittels zusätzlichen Wissens, das für beide Zellpopulationen über der Signalweg-Topologien und Gene, die von diesem Signalweg reguliert werden, gegeben ist. Unser Ansatz verbindet Ideen aus den vorherigen Kapiteln. Die bekannten regulierten Gene werden als Beispiele von differentiellen Genen in der *partially supervised* differentiellen Expressionsanalyse der Interventionsdaten (Kapitel 2) benutzt. Die Signalweg-Topologien werden als einfache Modelle formalisiert und in der Rekonstruktion der Genregulierung wie in Kapitel 3 genutzt. Wir quantifizieren Deregulation durch die Zusammenfassung von Regulierungssignalen der zwei Zellpopulationen in einen Wert. Unser Ansatz, JODA, stellt sich als vorteilhaft gegenüber separater Analyse der Zellpopulationen, sowie Analyse ohne Aufnahme von verfügbarem Wissen heraus. Mittels JODA charakterisieren wir weit verbreitete Veränderungen der regulatorischen Netzwerke, die durch DNA Schäden in menschlichen Zellen verursacht sind.

Curriculum Vitae

For reasons of data protection,
the curriculum vitae is not included in the online version.

For reasons of data protection,
the curriculum vitae is not included in the online version.

For reasons of data protection,
the curriculum vitae is not included in the online version

Ehrenwörtliche Erklärung

Hiermit erkläre ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel und Quellen verwendet habe.

Berlin, Oktober 2010 Ewa Szczurek