Chapter 3

Inferring transcriptional regulatory networks

The last chapter described statistical models to infer the topology of cellular networks by elucidating the correlation structure of pathway components. This chapter extends these models to include direct observations of intervention effects at other pathway components (section 3.1). The main contribution is a general concept of probabilistic interventions in Bayesian networks. My approach generalizes deterministic interventions, which fix nodes to certain states (section 3.2). I propose "pushing" variables in the direction of target states without fixing them (section 3.3) and formalize this idea in a Bayesian framework based on conditional Gaussian networks (section 3.4).

3.1 Graphical models for interventional data

In modern biology, the key to inferring gene function and regulatory pathways are experiments with interventions into the normal course of action in a cell. A common technique is to perturb a gene of interest experimentally and to study which other genes' activities are affected. A number of deterministic and probabilistic techniques have been proposed to infer regulatory dependencies from primary effects. In this section, we will give an overview over recent approaches, which are extensions of the methods discussed in the last chapter.

Linking causes with effects Rung *et al.* [112] build a directed graph by drawing an edge (i, j) if perturbing gene i results in a significant expression change at gene j. The authors focus on features of the network that are robust over a range of significance cutoffs. The inferred networks do not distinguish between direct and indirect effects. In this sense they are similar to co-expression networks. Fig. 3.1 shows the difference between a causal network and a network of affected components. In graph-theoretic terminology, the second network is the transitive closure of the first one.

Distinguishing direct from indirect effects A transitively closed network can be used as a starting point for further analysis. Wagner [142, 141, 140] uses graph-theoretic methods of *transitive reduction* [1, 75] to find the most parsimonious subgraph explaining all observed effects. These methods are deterministic and do not

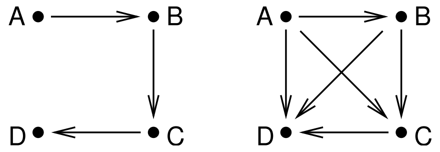

Figure 3.1: *From the causal network (left) it is easy to deduce how effects spread through the pathway (right). The harder problem is to deduce the causal pathway from observing effects of interventions (going from right to left).*

account for measurement noise. Wang and Cooper [143] describe a Bayesian generalization of the Wagner algorithm [140] yielding a distribution over possible causal relationships between genes.

Boolean networks A simple deterministic model of regulatory networks are Boolean networks: they are defined by a directed (and possibly cyclic) graph. Nodes correspond to genes and can take values 0 and 1. For each node exists a boolean function relating parent states to the child state. Perturbations allow to infer the structure and the logic of Boolean networks [63, 2, 3].

Correlation Rice *et al.* [107] build correlation graphs on knockout data. They assume that the data contain measurements of the unperturbed cell and several replicates of measurements for every gene knockout. For each gene i, they combine the wild-type data with the intervention data of this gene and compute on the joint data the correlation of gene i to all other genes. In the final graph, there is an arrow (i, j) whenever gene j was highly correlated to gene i. Since the correlation was computed on knockout data, the graph encodes causation and not only correlation. The big disadvantage of the method is the need for many (≥ 10) replicates of knockout experiments for every gene in the model. Data are used more efficiently by several regression methods.

Regression Rogers and Girolami [109] use sparse Bayesian regression based on a Gaussian linear model. They regress each gene onto all other genes by combining all the data corresponding to knockouts of genes other than the particular gene of interest. The measurements of the knockout gene are ignored when predicting this gene's expression from the other genes. In the next section we will see that this strategy is the same as Pearl's *ideal interventions* used in Bayesian networks [97]. A prior on model parameters constrains most regression coefficients to zero and enforces a sparse solution. Non-zero regression coefficients are indicated by arrows in the regulation network. The resulting graph is a special case of a Gaussian graphical model where directed edges are justified because the dataset contained knockouts of predictor variables.

Other regression methods for network reconstruction are derived from a branch of engineering called *system identification* [77]. Functional relations between network components are inferred from measurements of system dynamics. Several papers [151, 47, 28, 29] use multiple regression to model the response of genes and proteins to external perturbations.

Bayesian networks Bayesian networks represent the finest resolution of correlation structure. As shown in section 2.2, they present a prominent approach to derive a theoretical model for regulatory networks and pathways. Genes are represented by vertices of a network and the task is to find a topology, which explains dependencies between the genes. When learning from observational data only, groups of Bayesian networks may be statistically indistinguishable [139] as discussed in section 2.2. Information about effects of interventions helps to resolve such equivalence classes by including causal knowledge into the model [136, 137]. The final goal is to learn a graph structure, which not only represents statistical dependencies, but also causal relations between genes.

The following sections develop a theory for learning Bayesian network structure when data from different gene perturbation experiments is available. Section 3.2 reviews classical theory on modelling interventions in Bayesian networks. It shows that these concepts do not fit to realistic biological situations. A more appropriate model is introduced in section 3.3. It develops a theory of *soft interventions*, which push an LPD towards a target state without fixing it. A soft intervention can be realized by introducing a "pushing parameter" into the local prior distribution, which captures the pushing strength. We propose a concrete parametrization of the pushing parameter in the classical cases of discrete and Gaussian networks. Ideal interventions, which have been formally described by choosing a Dirac prior [137], can then be interpreted as infinite pushing.

Section 3.4 summarizes the results in the general setting of conditional Gaussian networks. This extends the existing theory on learning with hard interventions in discrete networks to learning with soft interventions in networks containing discrete and Gaussian variables. The concluding Section 3.4.3 deals with *probabilistic* soft interventions. In this set-up the pushing parameter becomes a random variable and we assign a prior to it. Hence, we account for the experimentalist's lack of knowledge on the actual strength of intervention by weighted averaging over all possible values.

3.2 Ideal interventions and mechanism changes

It is crucial that models reflect the way data was generated in the perturbation experiments. In Bayesian structure learning, Tian and Pearl [137] show that interventions can be modeled by imposing different parameter priors when the gene is actively perturbed or passively observed. They only distinguish between two kinds of interventions: most generally, interventions that change the local probability distribution of the node within a given family of distributions, and as a special case, interventions that fix the state of the variable deterministically. The first is called a *mechanism change*. It does not assume any prior information on *how* the local probability distribution changes. The second type of intervention, which fixes the state of the variable, is called a *do-operator* [97]. We will shortly describe both approaches to motivate our own model, which can be seen as lying intermediate these two extremes.

Ideal interventions Pearl [97] proposes an idealized *do-operator* model, in which the manipulation completely controls the node distribution. The influence of parent nodes is removed and the LPD $p(x_v|\mathbf{x}_{pa(v)}, \theta_v)$ degenerates to a point mass at the target state x'_v, that is,

$$p(x_v|\mathbf{x}_{pa(v)}, \theta_v) \quad \xrightarrow{\mathrm{do}(X_v=x'_v)} \quad p(x_v) = \begin{cases} 1 & \text{if } x_v = x'_v \\ 0 & \text{else.} \end{cases} \tag{3.1}$$

Fixing a variable to a state tells us nothing about its "natural" behaviour. When considering a single variable, data in which it was experimentally fixed has to be omitted. Cooper and Yoo [24] show: the marginal likelihood for data including interventional cases is of the same form as for observational cases only, but the counts go only over observations where a node was not fixed by external manipulation. We will discuss this result more deeply in section 3.4.

We will call Pearl's model a *hard (pushing) intervention*: it is directed to a target state and fixes the LPD deterministically. Hard interventions are used in almost all applications of interventional learning in Bayesian networks [152, 153, 130, 138, 99, 88, 22, 24].

A simulation study To test the effect of ideal interventions on structure learning, we conducted a simulation study on a small network of five nodes. Here, exhaustive enumeration is still possible and we can assess the complete score landscape. The simulation evaluated reconstruction accuracy with varying levels of noise and three different dataset sizes. The LPDs are convex combinations of signal and noise regulated by a parameter κ. The technical set-up is summarized in Fig. 3.2.

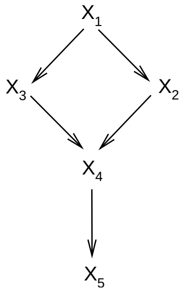

Figure 3.2: *The network topology used in the simulation. All random variables can take three values. For each parent state, the LPDs are a convex combination* $\kappa \cdot \text{signal} + (1-\kappa) \cdot \text{noise}$*, where "noise" is a uniform distribution over the three states and "signal" propagates the parent state. If* X_2 *and* X_3 *disagree,* X_4 *chooses uniformly between the two signals. More technical details are found in [82].*

Varying κ in steps of 0.1 in the intervall $[0, 0.9]$ we sampled two datasets of the same size: one only containing passive observations, and one sampled after ideal interventions at each node with equal number of replicates for each intervention experiment. On both datasets we scored all possible DAGs on 5 nodes and counted differences between the true and the top scoring topology. As errors we counted missing and spurious edges and also false edge directions. All these features are important when interpreting network topologies biologically.

The results of 5 repetitions can be seen in Fig. 3.3. The more data and the clearer the signal, the more pronounced is the advantage of active interventional learning over purely observational learning. While observational learning results in three equivalent topologies with the same high score, interventional learning resolves these ambiguities and yields a single best model. In summary, interventions are critical for effective

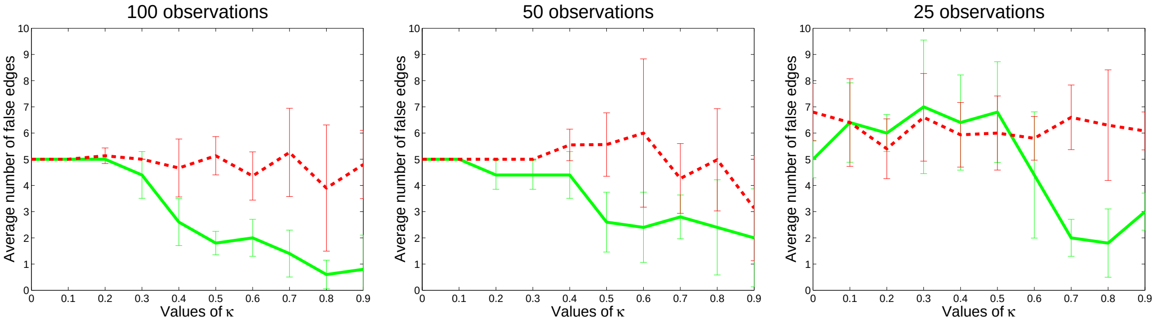

Figure 3.3: *Results of simulation experiments. The red dashed line corresponds to learning from observational data, the green solid line from learning with interventions. The bigger the sample-size and the clearer the signal, the larger is the gap between both lines.*

inference, particularly to establish directionality of the connections. Recently, this finding has been confirmed in other simulations [156] and on real data [114].

Mechanism changes Tian and Pearl [137] propose a model for local spontaneous changes that alter LPDs. They assume that no knowledge is available on the nature of the change, its location, or even whether it took place. Tian and Pearl derive a Bayesian score for structure learning by splitting the marginal likelihood for a node, at which a local change occurred, into two parts: one for the cases obtained before the change and one for the cases obtained after the change. A hard intervention as in Eq. (3.1) can be incorporated in this framework by assigning an informative prior distribution to the second part of the marginal likelihood. Tian and Pearl [137] show that the assumption (or knowledge) that only a *single* causal mechanism has changed, increases power in structure learning. Previously indistinguishable equivalent topologies may now be distinguished.

Problems Both hard interventions and mechanism changes face problems when being applied to real biological data from gene silencing experiments. Pearl's model of ideal interventions contains a number of idealizations: manipulations only affect single genes and results can be controlled deterministically. The first assumption may not be true for drug treatment and even in the case of single-gene knockouts there may be compensatory effects involving other genes. The second assumption is also very limiting in realistic biological scenarios. Often the experimentalist lacks knowledge about the exact size of perturbation effects. Due to measurement error or noise inherent in the observed system it may often happen that a variable, at which an intervention took place, is observed in a state different from the target state. In Pearl's framework, a single observation of this kind results in a marginal likelihood of zero. Mechanism changes, on the other hand, are also not suited to model real biological experiments, even though they capture uncertainty on intervention strength and accuracy. In real applications to reverse screens, at least the target of intervention is known and there is an expected response of the target to the intervention. Gene perturbations are *directed* in the sense that the experimental technique used tells us whether we should expect more or less functional target mRNA in the cell.

In summary, we need interventional data for successfull small-sample network reconstruction. Hard interventions (do-operations) are deterministic, mechanism changes are undirected. Both frameworks do not fit realistic biological situations. If we treat gene perturbation experiments as unfocussed mechanism changes we lose valuable information about what kind of intervention was performed. If we model them by a do-operator, we underestimate the stochastic nature of biological experiments. Thus, we need a concept of interventions, which is more directed than general mechanism changes, but still softer than deterministic fixing of variables. In the following, we focus on interventions, which specifically concentrate the local distribution at a certain node around some target state. We will call them *pushing interventions*, they are examples of mechanism changes with prior knowledge. We generalize hard pushing interventions (do-operator) to *soft pushing interventions*: the local probability distribution only centers more around the target value without being fixed. We follow Tian and Pearl [137] in splitting the marginal likelihood locally in two parts and assigning informative prior distributions. All interventions we will discuss are external manipulations of *single* nodes. None of them models global changes in the environment, which would change the dependency structure over the whole network and not just in a single family of nodes. Thus, we can start explaining soft interventions in the next section by concentrating on a single node in a Bayesian network.

3.3 Pushing interventions at single nodes

A Bayesian network is a graphical representation of the dependency structure between the components of a random vector $\mathbf{X}$. The individual random variables are associated with the vertices of a directed acyclic graph (DAG) D, which describes the dependency structure. Once the states of its parents are given, the probability distribution of a given node is fixed. Thus, the Bayesian network is completely specified by the DAG and the local probability distributions (LPDs). Although this definition is quite general, there are basically three types of Bayesian networks which are used in practice: discrete, Gaussian and conditional Gaussian (CG) networks. CG networks are a combination of the former two and will be treated in more detail in Section 3.4, for the rest of this section we focus on discrete and Gaussian networks. In discrete and Gaussian networks, LPDs are taken from the family of the multinomial and normal distribution, respectively. In the theory of Bayesian structure learning, the parameters of these distributions are not fixed, but instead a prior distribution is assumed [23, 48, 11]. The priors usually chosen because of conjugacy are the Dirichlet distribution in the discrete case and the Normal-inverse-χ^2 distribution in the Gaussian case. Averaging the likelihood over these priors yields the marginal likelihood – the key quantity in structure learning (see section 2.3.2).

An intervention at a certain node in the network can in this setting easily be modeled by a change in the LPDs' prior. When focusing on (soft) pushing interventions, this change should result in an increased concentration of the node's LPD around the target value. We model this concentration by introducing a pushing parameter w, which measures the strength of the pushing. A higher value of w results in a stronger concentration of the LPD. We now explain in more detail how this is done for discrete and Gaussian networks. Since the intervention only affects single variables and the joint distribution $p(\mathbf{x})$ in a Bayesian network factors according to the DAG structure in terms only involving a single node and its parents, it will suffice to treat families of discrete and Gaussian nodes separately.

3.3.1 Pushing by Dirichlet priors

We denote the set of discrete nodes by Δ and a discrete random variable at node $\delta \in \Delta$ by I_δ. The set of possible states of I_δ is $\mathcal{I}_\delta$. The parametrization of the discrete LPD at node δ is called θ_δ. For every configuration $\mathbf{i}_{pa(\delta)}$ of parents, θ_δ contains a vector of probabilities for each state $i_\delta \in \mathcal{I}_\delta$. Realizations of discrete random variables are multinomially distributed with parameters depending on the state of discrete parents. The conjugate prior is Dirichlet with parameters also depending on the state of discrete parents:

$$\begin{aligned} I_\delta \mid \mathbf{i}_{pa(\delta)}, \theta_\delta &\sim \text{Multin}(1, \theta_{\delta|\mathbf{i}_{pa(\delta)}}), \\ \theta_{\delta|\mathbf{i}_{pa(\delta)}} &\sim \text{Dirichlet}(\alpha_{\delta|\mathbf{i}_{pa(\delta)}}). \end{aligned} \tag{3.2}$$

We assume that the $\alpha_{\delta|\mathbf{i}_{pa(\delta)}}$ are chosen to respect likelihood equivalence [58]. A pushing intervention at node δ amounts to changing the prior parameters $\alpha_{\delta|\mathbf{i}_{pa(\delta)}}$ such that the multinomial density concentrates at some target value j. We formalize this by introducing a pushing operator $\mathcal{P}$ defined by

$$\mathcal{P}(\alpha_{\delta|\mathbf{i}_{pa(\delta)}}, w_\delta, j) \;=\; \alpha_{\delta|\mathbf{i}_{pa(\delta)}} + w_\delta \cdot \mathbf{1}_j, \tag{3.3}$$

where $\mathbf{1}_j$ is a vector of length $|\mathcal{I}_\delta|$ with all entries zero except for a single 1 at state j. The pushing parameter $w_\delta \in [0, \infty]$ determines the strength of intervention at node δ: if $w_\delta = 0$ the prior remains unchanged, if $w_\delta = \infty$ the Dirichlet prior degenerates to a Dirac distribution and fixes the LPD to the target state j. Figure 3.4 shows a three-dimensional example of increasing pushing strength w_δ.

3.3.2 Pushing by Normal-inverse-χ^2 priors

The set of Gaussian nodes will be called Γ and we denote a Gaussian random variable at node $\gamma \in \Gamma$ by Y_γ. In the purely Gaussian case it depends on the values of parents $\mathbf{Y}_{pa(\gamma)}$ via a vector of regression coefficients β_γ. If we assume that β_γ contains a first entry $\beta_\gamma^{(0)}$, the parent-independent contribution of Y_γ, and attach to $\mathbf{Y}_{pa(\gamma)}$ a leading 1, we can write for Y_γ the following regression model

$$\begin{aligned} Y_\gamma \mid \beta_\gamma, \sigma_\gamma^2 &\sim \text{N}(\mathbf{Y}_{pa(\gamma)}^\top \beta_\gamma,\ \sigma_\gamma^2), \\ \beta_\gamma \mid \sigma_\gamma^2 &\sim \text{N}(\mathbf{m}_\gamma,\ \sigma_\gamma^2 \mathbf{M}_\gamma^{-1}), \\ \sigma_\gamma^2 &\sim \text{Inv-}\chi^2(\nu_\gamma,\ s_\gamma^2). \end{aligned} \tag{3.4}$$

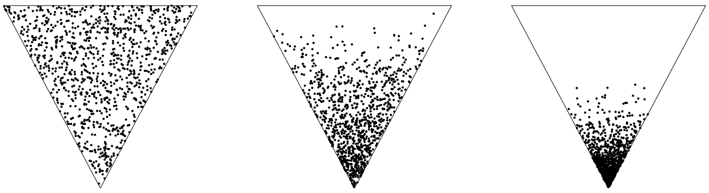

Figure 3.4: *Examples of pushing a discrete variable with three states. Each triangle represents the sample space of the three-dimensional Dirichlet distribution (which is the parameter space of the multinomial likelihood of the node). The left plot shows a uniform distribution with Dirichlet parameter* $\alpha = (1, 1, 1)$*. The other two plots show effects of pushing with increasing weight:* $w = 3$ *in the middle and* $w = 10$ *at the right. In each plot 1000 points were sampled.*

The regression coefficients follow a multivariate normal distribution with mean $\mathbf{m}_\gamma$ and covariance matrix $\sigma_\gamma^2 \mathbf{M}_\gamma^{-1}$, where σ_γ^2 is the variance of node Y_γ. The variance follows an inverse-χ^2 distribution. We assume that the prior parameters $\mathbf{m}_\gamma, \mathbf{M}_\gamma, \nu_\gamma, s_\gamma^2$ are chosen as in ref. [11].

As for discrete nodes, we implement a pushing intervention by adapting the prior distributions of model parameters. Pushing the distribution of Y_γ to target value k involves moving the mean by adapting the distribution of regression coefficients and concentrating the distribution by decreasing the variance σ_γ^2. To this end, we propose to exchange $\mathbf{m}_\gamma$ and s_γ^2 by $(\bar{\mathbf{m}}_\gamma, \bar{s}_\gamma^2) = \mathcal{P}((\mathbf{m}_\gamma, s_\gamma^2), w_\gamma, k)$ defined by

$$\begin{aligned} \bar{\mathbf{m}}_\gamma &= e^{-w_\gamma} \cdot \mathbf{m}_\gamma + (1 - e^{-w_\gamma}) \cdot k\mathbf{1}_1, \\ \bar{s}_\gamma^2 &= s_\gamma^2 / (w_\gamma + 1), \end{aligned} \tag{3.5}$$

where $k\mathbf{1}_1$ is a vector of length $|\mathbf{i}_{pa(\gamma)}| + 1$ with all entries zero except the first, which is k. We use $\mathcal{P}$ for the pushing operator as in the case of discrete nodes; which one to use will be clear from the context. Again $w_\gamma \in [0, \infty]$ represents intervention strength. The exponential function maps the real valued w into the interval $[0, 1]$. The interventional prior mean $\bar{\mathbf{m}}$ is a convex combination of the original mean $\mathbf{m}$ with a "pushing" represented by $k\mathbf{1}_1$. If $w_\gamma = 0$ the mean of the normal prior and the scale of the inverse-χ^2 prior remain unchanged. As $w_\gamma \to \infty$ the scale $\bar{s}^2$ goes to 0, so the prior for σ^2 tightens at 0. At the same time, the regression coefficients of the parents converge to 0 and β_0 approaches target value k. All in all, with increasing w_γ the distribution of Y_γ peaks more and more sharply at $Y_\gamma = k$. Note that the discrete pushing parameter w_δ and the Gaussian pushing parameter w_γ live on different scales and will need to be calibrated individually.

3.3.3 Hard pushing

Hard pushing means to make sure that a certain node's LPD produces almost surely a certain target value. It has been proposed by Tian and Pearl [137] to model this by imposing a Dirac prior on the LPD of the node. Although the Dirac prior is no direct member of neither the Dirichlet nor the Normal-inverse-χ^2 family of distributions it arises for both of them when taking the limit $w \to \infty$ for the pushing strength. Tian and Pearl [137] give an example for discrete networks by

$$p(\theta_{\delta|\mathbf{i}_{pa(\delta)}} \mid \mathrm{do}(X_\delta = x'_\delta)) = d(\theta_{i'_\delta|\mathbf{i}_{pa(\delta)}} - 1) \prod_{i_\delta \neq i'_\delta} d(\theta_{i_\delta|\mathbf{i}_{pa(\delta)}}), \tag{3.6}$$

where $d(\cdot)$ is the Dirac function: $d(x) = 1$, if $x = 0$, and $d(x) = 0$ else. This choice of the local prior distribution ensures that

$$\theta_{i_\delta|\mathbf{i}_{pa(\delta)}} = \begin{cases} 1 & \text{for } I_\delta = i'_\delta, \\ 0 & \text{else,} \end{cases}$$

in agreement with the definition of hard interventions in Eq. (3.1). We can easily extend this approach to Gaussian networks by defining a prior density as

$$p(\beta_\gamma, \sigma^2_\gamma \mid \mathrm{do}(Y_\gamma = k)) = d(\beta^{(0)}_\gamma - k) \prod_{i \in pa(\gamma)} d(\beta^{(i)}_\gamma) \cdot d(\sigma^2_\gamma). \tag{3.7}$$

Averaging over this prior sets the variance and the regression coefficients to zero, while $\beta^{(0)}_\gamma$ is set to k. Thus, the marginal distribution of Y_γ is fixed to state k with probability one.

3.3.4 Modeling interventions by policy variables

Hard interventions can be modeled by introducing a policy variable as an additional parent node of the variable at which the intervention is occuring [97, 127, 73]. In the same way we can use policy variables to incorporate soft interventions. For each node v, we introduce an additional parent node F_v ("F" for "force"), which is keeping track of whether an intervention was performed at X_v or not, and if yes, what the target state was. For a discrete variable I_δ, the policy variable F_δ has state space $\mathcal{I}_\delta \cup \emptyset$ and we can write

$$p(\theta_{\delta|\mathbf{i}_{pa(\delta)}, f_\delta}) = \begin{cases} \mathrm{Dirichlet}(\alpha_{\delta|\mathbf{i}_{pa(\delta)}}) & \text{if } F_\delta = \emptyset, \\ \mathrm{Dirichlet}(\bar{\alpha}_{\delta|\mathbf{i}_{pa(\delta)}}) & \text{if } F_\delta = j, \end{cases} \tag{3.8}$$

where $\bar{\alpha}_{\delta|\mathbf{i}_{pa(\delta)}} = \mathcal{P}(\alpha_{\delta|\mathbf{i}_{pa(\delta)}}, w_\delta, j)$ is derived from $\alpha_{\delta|\mathbf{i}_{pa(\delta)}}$ as defined in Eq. (3.3). For a continuous variable Y_γ, the policy variable F_γ has state space $\mathrm{I\!R} \cup \emptyset$ and we can write

$$p(\beta_{\gamma|f_\gamma}, \sigma^2_{\gamma|f_\gamma}) = \begin{cases} \mathrm{N}(\mathbf{m}_\gamma, \sigma^2_\gamma \mathbf{M}^{-1}_\gamma) \cdot \text{Inv-}\chi^2(\nu_\gamma, s^2_\gamma) & \text{if } F_\gamma = \emptyset, \\ \mathrm{N}(\bar{\mathbf{m}}_\gamma, \sigma^2_\gamma \mathbf{M}^{-1}_\gamma) \cdot \text{Inv-}\chi^2(\nu_\gamma, \bar{s}^2_\gamma) & \text{if } F_\gamma = k, \end{cases} \tag{3.9}$$

where $(\bar{\mathbf{m}}_\gamma, \bar{s}^2_\gamma) = \mathcal{P}((\mathbf{m}_\gamma, s^2_\gamma), w_\gamma, k)$ as defined in Eq. (3.5). Equations (3.8) and (3.9) will be used in section 3.4.2 to compute the marginal likelihood of conditional Gaussian networks from a mix of interventional and non-interventional data.

3.4 Pushing in conditional Gaussian networks

We summarize the results of the last section in the general framework of conditional Gaussian networks and compute the marginal likelihood for learning from soft interventions.

3.4.1 Conditional Gaussian networks

Conditional Gaussian (CG) networks are Bayesian networks encoding a joint distribution over discrete and continuous variables. We consider a random vector $\mathbf{X}$ splitting into two subsets: $\mathbf{I}$ containing discrete variables and $\mathbf{Y}$ containing continuous ones. The dependencies between individual variables in $\mathbf{X}$ can be represented by a directed acyclic graph (DAG) D with node set V and edge set E. The node set V is partitioned as $V = \Delta \cup \Gamma$ into nodes of discrete (Δ) and continuous (Γ) type. Each discrete variable corresponds to a node in Δ and each continuous variable to a node in Γ. The distribution of a variable X_v at node v only depends on variables $\mathbf{X}_{pa(v)}$ at parent nodes $pa(v)$. Thus, the joint density $p(\mathbf{x})$ decomposes as

$$\begin{aligned} p(\mathbf{x}) &= p(\mathbf{i}, \mathbf{y}) = p(\mathbf{i})p(\mathbf{y}|\mathbf{i}) \\ &= \prod_{\delta \in \Delta} p(i_\delta | \mathbf{i}_{pa(\delta)}) \cdot \prod_{\gamma \in \Gamma} p(y_\gamma | \mathbf{y}_{pa(\gamma)}, \mathbf{i}_{pa(\gamma)}). \end{aligned} \tag{3.10}$$

The discrete part, $p(\mathbf{i})$, is given by an unrestricted discrete distribution. The distribution of continuous random variables given discrete variables, $p(\mathbf{y}|\mathbf{i})$, is multivariate normal with mean and covariance matrix depending on the configuration of discrete variables. Since discrete variables do not depend on continuous variables, the DAG D contains no edges from nodes in Γ to nodes in Δ.

For discrete nodes, the situation in CG networks is exactly the same as in the pure case discussed in Section 3.3: The distribution of $I_\delta | \mathbf{i}_{pa(\delta)}$ is multinomial and parametrized by θ_δ. Compared to the purely Gaussian case treated in Section 3.3, we have for Gaussian nodes in CG networks an additional dependency on discrete parents. This dependency shows in the regression coefficients and the variance, which now not only depend on the node, but also on the state of the discrete parents:

$$Y_\gamma \mid \beta_{\gamma|\mathbf{i}_{pa(\gamma)}}, \sigma^2_{\gamma|\mathbf{i}_{pa(\gamma)}} \sim \mathrm{N}(\mathbf{Y}^\top_{pa(\gamma)} \beta_{\gamma|\mathbf{i}_{pa(\gamma)}},\ \sigma^2_{\gamma|\mathbf{i}_{pa(\gamma)}}). \tag{3.11}$$

As a prior distribution we again take the conjugate normal-inverse-χ^2 distribution as in Eq. (3.4). For further details on CG networks we refer to references [72, 11].

3.4.2 Learning from interventional and non-interventional data

Assuming an uniform prior over network structures D, the central quantity to be calculated is the *marginal likelihood* $p(M|D)$. In the case of only one type of data it can be written as

$$p(M|D) \;=\; \int_{\Theta} p(M|D,\theta)\; p(\theta|D)\; \mathrm{d}\theta. \tag{3.12}$$

Here $p(\theta|D)$ is the prior on the parameters θ of the LPDs. If the dataset contains both interventional and non-interventional cases, the basic idea is to choose parameter priors locally for each node as in Eq. (3.8) and Eq. (3.9) according to whether a variable was perturbed in a certain case or not. We will see that this strategy effectively leads to a local split of the marginal likelihood into an interventional and a non-interventional part.

A family-wise view of marginal likelihood To compute the marginal likelihood of CG networks on interventional and non-interventional data, we rewrite Eq. (3.12) in terms of single nodes such that the theory of (soft) pushing from Section 3.3 can be used. In the computation we will use the following technical utilities:

1. The dataset M consists of N cases $\mathbf{x}^1, \dots, \mathbf{x}^N$, which are sampled independently. Thus we can write $p(M|D,\theta)$ as a product over all single case likelihoods $p(\mathbf{x}^c|D,\theta)$ for $c = 1, \dots, N$.
2. The joint density $p(\mathbf{x})$ factors according to the DAG D as in Eq. (3.10). Thus, for each case $\mathbf{x}^c$ we can write $p(\mathbf{x}^c|D,\theta)$ as a product over node contributions $p(x_v^c|\mathbf{x}_{pa(v)}^c, \theta_v)$ for all $v \in V$.
3. We assume *parameter independence*: the parameters associated with one variable are independent of the parameters associated with other variables, and the parameters are independent for each configuration of the discrete parents [58]. Thus, all dependencies between variables are encoded in the network structure. Parameter independence allows us to decompose the prior $p(\theta|D)$ in Eq. (3.12) into node-wise priors $p(\theta_{v|\mathbf{i}_{pa(v)}}|D)$ for a given parent configuration $\mathbf{i}_{pa(v)}$.
4. All interventions are soft pushing. For a given node, intervention strength and target state stay the same in all cases in the data, but of course different nodes may have different pushing strengths and target values. This constraint just helps us to keep the following formulas simple and can easily be dropped.

These four assumptions allow a family-wise view of the marginal likelihood. Before we present it in a formula, it will be helpful to introduce a *batch notation.* In CG networks, the parameters of the LPD at a certain node depend only on the configuration of discrete parents. This holds for both discrete and Gaussian nodes. Thus, when evaluating the likelihood of data at a certain node, it is reasonable to collect all cases in a batch, which correspond to the same parent configuration:

$$\begin{aligned} p(M|D,\theta) &= \prod_{c\in M} p(\mathbf{x}^c|D,\theta) \\ &= \prod_{c\in M}\prod_{v\in V} p(x_v^c|\mathbf{x}^c_{pa(v)},\theta_v) \\ &= \prod_{v\in V}\ \prod_{\mathbf{i}_{pa(v)}\in\mathcal{I}_{pa(v)}}\ \prod_{c:\mathbf{i}^c_{pa(v)}=\mathbf{i}_{pa(v)}} p(x_v^c|\mathbf{i}^c_{pa(v)},\mathbf{y}_{pa(v)},\theta_v) \end{aligned} \tag{3.13}$$

The last formula is somewhat technical: If the node v is discrete, then $\mathbf{y}_{pa(v)}$ will be empty, and usually not all parent configuration $\mathbf{i}_{pa(v)}$ are found in the data, so some terms of the product will be missing. For each node we gather the cases with the same joint parent state in a batch $B_{\mathbf{i}_{pa(v)}} = \{c \in 1,\dots,N \ :\ \mathbf{i}^c_{pa(v)} = \mathbf{i}_{pa(v)}\}$. When learning with interventional data, we have to distinguish further between observations of a variable which were obtained passively and those that are result of intervention. Thus, for each node v we split the batch $B_{\mathbf{i}_{pa(v)}}$ into one containing all observational cases and one containing the interventional cases:

$$\begin{aligned} B^{obs}_{\mathbf{i}_{pa(v)}} &= \{c \in 1,\dots,N\ : \mathbf{i}^c_{pa(v)} = \mathbf{i}_{pa(v)} \text{ and no intervention at } v\}, \\ B^{int}_{\mathbf{i}_{pa(v)}} &= \{c \in 1,\dots,N\ : \mathbf{i}^c_{pa(v)} = \mathbf{i}_{pa(v)} \text{ and intervention at } v\}. \end{aligned}$$

If there is more than one type of intervention applied to node v, the batch containing interventional cases has to be split accordingly. Using this notation we can now write down the marginal likelihood for CG networks in terms of single nodes and parents:

$$\begin{aligned} p(M|D) = &\prod_{v\in V}\prod_{\mathbf{i}_{pa(v)}}\int_\Theta \prod_{o\in B^{obs}_{\mathbf{i}_{pa(v)}}} p(x_v^o|\mathbf{i}_{pa(v)},\mathbf{y}^o_{pa(v)},\theta_v)\ p'(\theta_v|D)\ \mathrm{d}\theta_v\ \times \\ &\prod_{v\in V}\prod_{\mathbf{i}_{pa(v)}}\int_\Theta \prod_{e\in B^{int}_{\mathbf{i}_{pa(v)}}} p(x_v^e|\mathbf{i}_{pa(v)},\mathbf{y}^e_{pa(v)},\theta_v)\ p''(\theta_v|D,w_v)\ \mathrm{d}\theta_v. \end{aligned} \tag{3.14}$$

At each node, we use distributions and priors as defined in Eq. (3.8) for discrete nodes and Eq. (3.9) for Gaussian nodes. The non-interventional prior p' corresponds to $F_v = \emptyset$ and the interventional prior p'' corresponds to F_v equalling some target value. We denoted the intervention strength explicitly in the formula, since we will focus on it further when discussing *probabilistic* soft interventions in Section 3.4.3. Equation (3.14) consists of an observational and an interventional part. Both can further be split into a discrete and a Gaussian part, so we end up with four terms to consider.

Discrete observational part To write down the marginal likelihood of discrete observational data, we denote by $n_{i_\delta|\mathbf{i}_{pa(\delta)}}$ the number of times we passively observe $I_\delta = i_\delta$ in batch $B^{obs}_{\mathbf{i}_{pa(\delta)}}$, and by $\alpha_{i_\delta|\mathbf{i}_{pa(\delta)}}$ the corresponding pseudo-counts of the Dirichlet prior. Summation of $\alpha_{i_\delta|\mathbf{i}_{pa(\delta)}}$ and $n_{i_\delta|\mathbf{i}_{pa(\delta)}}$ over all $i_\delta \in \mathcal{I}_\delta$ is abbreviated by $\alpha_{\mathbf{i}_{pa(\delta)}}$

and $n_{\mathbf{i}_{pa(\delta)}}$, respectively. Then, the integral in the observational part of Eq. (3.14) can be computed as follows:

$$\begin{aligned}
&\int\limits_{\Theta} \prod_{o \in B^{obs}_{\mathbf{i}_{pa(v)}}} p(x^o_v|\mathbf{i}_{pa(v)}, \mathbf{y}^o_{pa(v)}, \theta_v)\; p'(\theta_v|D)\; \mathrm{d}\theta_v \;= \\
&= \int\limits_{\Theta} \left(\prod_{i_\delta \in \mathcal{I}_\delta} \theta_{i_\delta|\mathbf{i}_{pa(\delta)}}^{n_{i_\delta|\mathbf{i}_{pa(\delta)}}} \right) \left(\frac{\Gamma(\alpha_{\mathbf{i}_{pa(\delta)}})}{\prod_{i_\delta} \Gamma(\alpha_{i_\delta|\mathbf{i}_{pa(\delta)}})} \prod_{i_\delta \in \mathcal{I}_\delta} \theta_{i_\delta|\mathbf{i}_{pa(\delta)}}^{\alpha_{i_\delta|\mathbf{i}_{pa(\delta)}}-1} \right) \mathrm{d}\theta_v \\
&= \frac{\Gamma(\alpha_{\mathbf{i}_{pa(\delta)}})}{\prod_{i_\delta} \Gamma(\alpha_{i_\delta|\mathbf{i}_{pa(\delta)}})} \int\limits_{\Theta} \prod_{i_\delta \in \mathcal{I}_\delta} \theta_{i_\delta|\mathbf{i}_{pa(\delta)}}^{\alpha_{i_\delta|\mathbf{i}_{pa(\delta)}}+n_{i_\delta|\mathbf{i}_{pa(\delta)}}-1} \,\mathrm{d}\theta_v && (3.15) \\
&= \frac{\Gamma(\alpha_{\mathbf{i}_{pa(\delta)}})}{\prod_{i_\delta} \Gamma(\alpha_{i_\delta|\mathbf{i}_{pa(\delta)}})} \cdot \frac{\prod_{i_\delta} \Gamma(\alpha_{i_\delta|\mathbf{i}_{pa(\delta)}} + n_{i_\delta|\mathbf{i}_{pa(\delta)}})}{\Gamma(\alpha_{\mathbf{i}_{pa(\delta)}} + n_{\mathbf{i}_{pa(\delta)}})} && (3.16)
\end{aligned}$$

The first equations follow from substituting the densities of likelihood and prior into the integral. The last equation results from the fact that the Dirichlet distribution integrates to one and thus the Dirichlet integral in line (3.15) is equal to the inverse normalizing constant of Dirichlet$(\alpha_{i_\delta|\mathbf{i}_{pa(\delta)}} + n_{i_\delta|\mathbf{i}_{pa(\delta)}})$.

The formula in Eq. 3.16 describes the score constribution of a single node with fixed parent configuration. The marginal likelihood of the discrete data M_Δ can be written as the local contributions of Eq. (3.16) multiplied over all possible nodes and parent configurations, that is,

$$p_{obs}(M_\Delta \mid D) = \prod_{\delta \in \Delta} \prod_{\mathbf{i}_{pa(\delta)}} \left(\frac{\Gamma(\alpha_{\mathbf{i}_{pa(\delta)}})}{\Gamma(\alpha_{\mathbf{i}_{pa(\delta)}} + n_{\mathbf{i}_{pa(\delta)}})} \prod_{i_\delta \in \mathcal{I}_\delta} \frac{\Gamma(\alpha_{i_\delta|\mathbf{i}_{pa(\delta)}} + n_{i_\delta|\mathbf{i}_{pa(\delta)}})}{\Gamma(\alpha_{i_\delta|\mathbf{i}_{pa(\delta)}})} \right). \quad (3.17)$$

This result was first obtained by Cooper and Herskovits [23] and is further discussed by Heckerman *et al.* [58].

Discrete interventional part Since interventions are just changes in the prior, the marginal likelihood of the interventional part of discrete data is of the same form as Eq. (3.17). The prior parameters $\alpha_{i_\delta|\mathbf{i}_{pa(\delta)}}$ are exchanged by $\alpha'_{i_\delta|\mathbf{i}_{pa(\delta)}} = \mathcal{P}(\alpha_{i_\delta|\mathbf{i}_{pa(\delta)}}, w_\delta, j)$ as given by Eq. (3.3), and the counts $n_{i_\delta|\mathbf{i}_{pa(\delta)}}$ are exchanged by $n'_{i_\delta|\mathbf{i}_{pa(\delta)}}$ taken from batch $B^{int}_{\mathbf{i}_{pa(\delta)}}$.

In the limit $w_\delta \to \infty$ this part converges to one and vanishs from the overall marginal likelihood $p(M|D)$. Thus, in the limit we achieve the result of Cooper and Yoo [24] and Tian and Pearl [137].

Gaussian observational part All cases in batch $B^{obs}_{\mathbf{i}_{pa(\gamma)}}$ are sampled independently from a normal distribution with fixed parameters. If we gather them in a vector $\mathbf{y}_\gamma$ (of length $b = |B^{obs}_{\mathbf{i}_{pa(\gamma)}}|$) and the corresponding states of continuous parents as rows in a matrix $\mathbf{P}_\gamma$ (of dimension $b \times (|pa(\gamma)|+1)$), we yield the standard regression scenario

$$\mathbf{y}_\gamma \mid \beta_\gamma, \sigma^2_\gamma \sim \mathrm{N}(\mathbf{P}_\gamma \beta_\gamma, \sigma^2_\gamma \mathbf{I}), \quad (3.18)$$

where $\mathbf{I}$ is the $b \times b$ identity matrix. As a prior distribution over regression coefficients β_γ and variance σ_γ^2 we choose normal-inverse-χ^2 as shown in Eq. (3.4). Marginalizing with respect to β_γ and σ_γ^2 yields a multivariate t-distribution of dimension b, with location vector $\mathbf{P}_\gamma \mathbf{m}_\gamma$, scale matrix $s(\mathbf{I} + \mathbf{P}_\gamma \mathbf{M}_\gamma^{-1} \mathbf{P}_\gamma^\top)$, and ν_γ degrees of freedom. This can be seen by the following argument. To increase readability, we drop the index "γ" in the following equations. Then, Eq. (3.18) can be rewritten as

$$\mathbf{y} \;=\; \mathbf{P}\beta + \varepsilon \quad \text{with } \varepsilon \sim \mathrm{N}(0, \sigma^2 \mathbf{I}). \tag{3.19}$$

The prior distribution of $\beta|\sigma^2$ is Gaussian with mean $\mathbf{m}$ and variance $\sigma^2\mathbf{M}^{-1}$. Thus we can write

$$\mathbf{P}\beta \mid \sigma^2 \;\sim\; \mathrm{N}(\mathbf{Pm}, \sigma^2 \mathbf{P}\mathbf{M}^{-1}\mathbf{P}^\top) \tag{3.20}$$

Since ε is independent of β when conditioning on σ^2 we conclude that

$$\mathbf{y} \mid \sigma^2 \;\sim\; \mathrm{N}(\mathbf{Pm}, \sigma^2(\mathbf{I} + \mathbf{P}\mathbf{M}^{-1}\mathbf{P}^\top)). \tag{3.21}$$

The prior for σ^2 is inverse-χ^2 with scale s and ν degrees of freedom. Marginalizing with respect to σ^2 yields

$$\mathbf{y} \;\sim\; t_b(\mathbf{Pm}, s(\mathbf{I} + \mathbf{P}\mathbf{M}^{-1}\mathbf{P}^\top), \nu). \tag{3.22}$$

Note that all the distribution parameters above are specific for node γ. When using data from different batches, every parameter additionally carries an index "$\mathbf{i}_{pa(\gamma)}$" indicating that it depends on the state of the discrete parents of the Gaussian node γ. Multiplying t-densities for all nodes and configurations of discrete parents—the outer double-product in Eq. (3.14)—yields the marginal likelihood of the Gaussian part.

Gaussian interventional part Here we consider cases in batch $B^{int}_{\mathbf{i}_{pa(\gamma)}}$. We collect them in a vector and can again write a regression model like in Eq. (3.18). The difference to the observational Gaussian case lies in the prior parameters. They are now given by Eq. (3.5). The result of marginalization is again a t-density with parameters as above. The only difference is that the pair $(\mathbf{m}, s)$ is exchanged by $(\mathbf{m}', s') = \mathcal{P}((\mathbf{m}, s), w_\gamma, k)$. The Gaussian interventional part is then given by a product of such t-densities over nodes and discrete parent configurations.

If we use the hard intervention prior in Eq. (3.7) instead, the Gaussian interventional part integrates to one and vanishs from the marginal likelihood in Eq. (3.14). Thus, we extended the results by Cooper and Yoo [24] to Gaussian networks.

3.4.3 Probabilistic soft interventions

In Section 3.3 we introduced the pushing operator $\mathcal{P}(\cdot, w_v, t_v)$ to model a soft intervention at a discrete or Gaussian node v. The intervention strength w_v is a parameter, which has to be chosen before network learning. There are several possibilities, how to do it. If there is solid experimental experience on how powerful interventions are, this can be reflected in an appropriate choice of w_v. An obvious problem is that w_v needs to be determined on a scale that is compatible with the Bayesian network model. If there is prior knowledge on parts of the network topology, the parameter w_v can be tuned until the result of network learning fits the prior knowledge. Note again that by the parametrization of pushing given in Section 3.3, the pushing strengths for discrete and Gaussian nodes live on different scales and have to be calibrated separately.

However, a closer inspection of the biological background in chapter 1, which motivated the theory of soft pushing interventions, suggests to treat the intervention strength w_v as a random variable. In gene silencing an inhibiting molecule (a double-stranded RNA in case of RNAi) is introduced into the cell. This usually works in a high percentage of affected cells. In the case of success, the inhibitor still has to spread throughout the cell to silence the target gene. This diffusion process is stochastic and consequently causes experimental variance in the strength of the silencing effect.

These observations suggest to assign a prior distribution $p(w_v)$ to the intervention strength. That is, we drop the assumption of having one intervention strength in all cases, but instead average over possible values of w_v. For simplicity we assume there is only a limited number of possible values of w_v, say, $w_v^{(1)}, \dots, w_v^{(k)}$, with an arbitrary discrete distribution assigned to them. Then we can express our inability to control the pushing strength in the experiment deterministically by using a mixed prior of the form

$$p(\theta_v|D) = \sum_{i=1}^{k} q_k \, p(\theta_v|D, w_v^{(k)}). \tag{3.23}$$

Here, the mixture coefficients $q_k = p(w_v^{(k)})$ are the prior probabilities of each possible pushing strength. The terms $p(\theta_v|D, w_v^{(k)})$ correspond to Dirichlet densities in the discrete case and Normal-inverse-χ^2 densities in the Gaussian case. In RNAi experiments, $w_v^{(1)}, \dots, w_v^{(k)}$ can be estimated from the empirical distribution of measured RNA degradation efficiencies in repeated assays. Mixed priors as in Eq. (3.23) are often used in biological sequence analysis to express prior knowledge which is not easily forced into a single distribution. See Durbin *et al.* [34] for details. If we substitute the prior $p''(\theta_v|D, w_v)$ in the interventional part of Eq. (3.14) with the mixture prior in Eq. (3.23), the marginal likelihood of a family of nodes is a mixture of marginal likelihoods corresponding to certain values $w_v^{(k)}$ weighted by mixture coefficients q_k.

Discussion

Our work extends structure learning from interventional data into two directions: from learning discrete networks to learning mixed networks and from learning with hard interventions to learning with soft interventions. Soft interventions are focussed on a specific target value of the variable of interest and concentrate the local probability distribution there. We proposed parametrizations for pushing discrete and continuous variables using Dirichlet and Normal-inverse-χ^2 priors, respectively. We computed the marginal likelihood of CG networks for data containing both observational and (soft) interventional cases. In Bayesian structure learning, the marginal likelihood is the key quantity to compute from data. Using it (and possibly a prior over network structures) as a scoring function, we can start model search over possible network structures. For a survey of search heuristics see section 2.3.4.

Since in biological settings the pushing strength is unknown, we proposed using a mixture prior on it, resulting in a mixture marginal likelihood. This makes the score for each network more time-consuming to compute. But in applications there is often a large amount of biological prior knowledge, which limits the number of pathway candidates from the beginning. When learning network structure we usually don't have to optimize the score over the space of all possible DAGs but are limited to a few candidate networks, which are to be compared. This corresponds to a very rigid structure prior.

Modeling interventions as soft pushing makes structure learning more robust against noise. Soft interventions handle major sources of noise inherent in real biological systems. This is a central benefit of our approach.

Beyond transcriptional networks At the end of chapter 2 we found that visualizing the correlation structure of gene expression may not give us a biologically meaningful answer. As a first reason for this shortcoming we discussed the need for interventional data. To address this issue, the present chapter introduced a novel model of interventions in Bayesian networks. But there is also a second reason, why a visualization of correlation structure on expression data may not give us the full picture. We need to have a second look at the rationale, which made us use graphical models in the first place.

The application of graphical models is motivated by the following consideration: if the expression of gene A is regulated by proteins B and C, then A's expression level is a stochastic function of the joint activity levels of B and C. Expression levels of genes are taken as a proxy for the activity level of the proteins they encode. This is the rationale leading to the application of Bayesian networks to expression data [41]. It relies on the assumption that both the regulator and its targets must be transcriptionally regulated, resulting in detectable changes in their expression. Indeed, recent large-scale analyses of the regulatory networks of *Escherichia coli* [121] and *S. cerevisiae* [74, 86] found a number of cases in which the regulators are themselves transcriptionally regulated. Simon *et al.* [123] show direct dependencies of cell cycle transcriptional regulators in yeast between different cell cycle stages. Regulators that function during one stage of the cell cycle contribute to the regulation of transcriptional activators active in the next stage. These studies show the importance of transcriptional regulation in controlling gene expression.

On the other hand, these observations cannot obscure the fact that models of correlation structure of mRNA levels have only limited explanatory value, as can be seen by the two following studies. Gygi *et al.* [54] found that correlation between mRNA and protein levels was poor in yeast. Quantitative mRNA data was insufficient to predict protein expression levels. They found cases where the protein levels varied by more than 20-fold, even if the mRNA levels stayed the same. Additionally, activation or silencing of a regulator is in most cases carried out by posttranscriptional protein modifications [71]. Thus, even knowing the correct expression state is not enough, we also need to know the activation state of the protein. In summary, activation levels of proteins cannot be approximated well by expression levels of corresponding genes. However, the next chapter will show that the situation is not hopeless. We will show that secondary effects of interventions are visible as expression changes on microarray data. Transcriptional effects allow to infer regulatory hierarchies in non-transcriptional parts of a pathway.