Markov Chain Monte Carlo Methods
for Parameter Identification
in Systems Biology Models

Theresa Niederberger
aus
Bad Reichenhall, Deutschland

2012

Erklärung

Diese Dissertation wurde im Sinne von § 7 der Promotionsordnung vom
28. November 2011 von Herrn Prof. Dr. Patrick Cramer betreut.

Eidesstattliche Versicherung

Diese Dissertation wurde eigenständig und ohne unerlaubte Hilfe erarbeitet.

München, 23.4.2012

_____

Theresa Niederberger

Dissertation eingereicht am: 23.4.2012

1. Gutachter: Prof. Dr. Patrick Cramer
2. Gutachter: Prof. Dr. Achim Tresch

Mündliche Prüfung am: 4.6.2012

Acknowledgments

Abstract

Understanding nature is a deep human desire. Therefore, experiments are carried out and measurements are performed to shed light onto what is so far unknown. But experiments only lead to an information gain which is restricted to very limited circumstances and conditions. In order to generalize information, to derive conclusions relevant to a broader scale, and to make well-founded predictions, in short, to actually "learn" something, abstract models have to be established.

Model selection, i.e., finding a suitable model class (here: defining the mathematical framework of the model) is a challenging task. A good model has to strike the balance between a sufficiently detailed description of the observations (complexity) and a good generalization performance (simplicity). More precisely, it has to be as accurate as possible without overparametrization and overfitting. Graphical models reduce the complexity by encoding dependencies among random variables, enabling the factorization of the joint probability distribution into a product of simpler "local" distributions.

After a suitable model class has been selected, the parameters describing it can be estimated based on the observations (parameter inference). Bayesian methods approach this task by including prior knowledge and maximizing the posterior probability of the parameters, considered as random variables, given the observations. Due to the inherent complexity of the chosen model class, however, this posterior distribution can't be accessed analytically in many situations. Approximation algorithms such as Markov Chain Monte Carlo sampling solve this problem and avoid complex or even infeasible calculations by drawing representative samples from the distribution of interest. In addition to a simple "best fit", they provide valuable information on the uniqueness of the solution and the variability of the parameter estimates as a function of the data.

Both model selection and parameter inference benefit from the development of increasingly faster computer systems in recent years, which facilitate the exploration of large model spaces. Basically, model selection and parameter inference are subject to three major kinds of error: The model bias arises from the fact that essentially every model class is an abstract and usually compressed version of the true physical processes involved and hence necessarily not correct. Since it is difficult if not practically impossible to identify the best model within a model class, the estimation bias quantifies the difference between the best fit and the estimate that has been obtained from the inference algorithm. The variance represents the error that is due to the stochastic nature of measurements. The extent of these errors can and must be assessed by simulations.

This thesis is at the interface of statistics and biochemistry. In order to be self contained, I included the necessary basics from both fields. The focus of this thesis is on parameter inference using Markov Chain Monte Carlo sampling, i.e., on the parameter estimation within a given model class. I introduce three approaches which have been developed for different biological applications. First, I present MC EMiNEM, a sampling scheme that combines Expectation Maximization with MCMC sampling in the class of NEMs. MC EMiNEM has been developed for the reconstruction of regulatory networks and was applied to a set of four perturbation studies on the yeast Mediator, a transcriptional cofactor. We were able to derive new insights into the functional dependencies within the complex and its interactions with gene specific transcription factors. Second, I present an analysis method for the processive degradation or synthesis of biomolecules, based on a set of ordinary differential equations. In close collaboration with Karl-Peter Hopfner and Sophia Hartung, this method was applied to quantitatively analyze RNA degradation by the archaeal exosome. The results lead to a more profound understanding of the involved kinetics, in dependence of both substrate features and the architecture of the exosome. Third, I describe a reversible-jump MCMC algorithm for simultaneous model selection and parameter inference. Here, we use the recently developed class of factor graphs to model cellular decision processes. The method has been applied to hematopoietic stem cell genealogies.

Contents

Contents

List of Figures

Differentiation of hematopoietic stem cells   **72**

Appendix   **87**

List of Tables

Part I.

Mathematical Background

1. A brief outline on statistical modeling

1.1. Model selection

A model is an abstract and usually compressed description of the observed data, in terms of a certain model class, and a set of parameters that identify the concrete model within this class. Its construction can be based on established knowledge, extrapolation from similar models or even merely on intuition. It can be rather specific, such as the 3D model for a protein, or very general, such as Michaelis-Menten kinetics which can be applied to a variety of enzymes. Depending on the availability of knowledge and data it can be simple, containing only few parameters, or rather complex. Ideally, a model enables the generalization of information gained from specific experiments under predefined circumstances and conditions to derive conclusions on broader implications, and to make well-founded predictions [50, Chapter 1.2]. This generalizability is the reason why the process of building a model from given data is commonly referred to as "learning".

Finding a suitable model is a challenging task. A model has to be appropriate for the situation it is applied to, which means that it has to answer the questions it is designed for and that it has to find a reasonable trade-off between complexity and simplicity: On the one hand it has to fit the measured data at the best, on the other hand it must be able to make general predictions. In other words, it has to represent all necessary details without overfitting [50, Chapter 4.4].

1.1.1. Overparametrization and overfitting

Overfitting occurs when the amount of (or the kind of) data used to fit a model (the training set) is insufficient. It means that the model's parameters are overly dependent on the training set and results in a poor prediction quality on other datasets [8, Chapter 3.2]. The risk of overfitting is addressed by Occam's razor which states that if two model classes are equally capable of explaining the data, it is better to choose the simpler one. This principle is supported for three reasons: First, simplicity is preferred for aesthetic reasons, second, Occam's razor has shown to be successful in practice, and third, Bayesian inference actually embeds Occam's razor and so the simpler solution is indeed more probable [65, Chapter 28]. Reasonable parametrization of the model is in line with Occam's razor: the number of parameters describing a model and the amount of experimental data have to be kept sensibly balanced. If there are not enough measurements available to distinguish between different parameter choices, the model is overparameterized (also underdetermined or ill-determined), and notoriously sensitive to overfitting.

The choice of parameters has turned out to be especially challenging in Part III. The aim of this project was to analyze the impact of various mutations in the archaeal exosome on its efficiency to degrade RNA depending on the length of the substrate. The data consisted of measurements of the total amount of RNA for each length at predefined time points, and the process of RNA degradation was modeled as a set of ordinary differential equations (ODEs). Yet, the intuitive model consisting of the pure decay rate, as well as association and dissociation of the substrate to, respectively, from the exosome turned out to be ill-specified (Section 11.3). A reparametrization of the model together with a reduction in the number of free parameters could solve the problem. If one still wishes to extend the aim of the project with regard to a more precise definition of the impact caused by the mutations, additional measurements will be required.

1.1.2. Strategies for model selection

Model selection can be performed manually based on knowledge and intuition, or automatically based on predefined criteria. Increasingly faster computer systems make it possible to explore increasingly larger model spaces. Automated model selection has been used in the reversible-jump MCMC approach applied to the hematopoietic stem cell genealogies in Part IV. In this case, parameter estimation and model selection alternate since the best model class is not known *a priori*. More details are provided in the corresponding part of this thesis, as well as in Section 2.4.

To avoid overfitting and to provide a sound and unbiased evaluation of the generalization performance of the selected model, three steps based on three different datasets are recommended. The training set is used to fit the model, i.e., to tune the hyperparameters (e.g., the weight of the sparseness prior in Part II). Then, the validation set is used to calculate the prediction error of the current model and to decide whether it is appropriate or not. After a model has been selected, the test set is used for the final assessment. The amount of data required for these steps depends on the quality of the data (signal-to-noise ratio) and on the complexity of the model. It can be reduced by cross-validation or bootstrapping methods [37, Chapter 7.2]. The prediction error of a model is measured by loss functions. An example is the squared error $L(\Theta) := \sum_{(x,y)} (y - f(x; \Theta))^2$, where y is the true target parameter, $f(x; \Theta)$ is a prediction model with parameter set Θ, and x is a data point on which the prediction is based. The aim is to minimize this loss function [37].

1.2. Graphical models

A model is an abstract representation of observations usually described by a set of parameters. A statistical model interprets the observations as random variables and assigns probability distributions $\{P(D|\theta) : \theta \in \Theta\}$ to them, where D is the observed data and θ is an unknown parameter set taken from the parameter space Θ [107, Chapter 6]. This makes the model accessible to statistical methods.
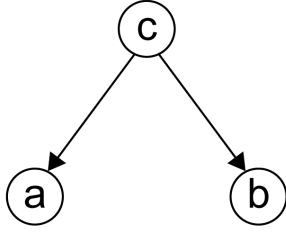
Figure 1.1: In a graphical model, nodes represent variables and edges represent the dependencies between these variables. Here, the dependency structure allows the following decomposition of the joint probability distribution: $p(a, b, c) = p(a|c) \cdot p(b|c) \cdot p(c)$. The figure has been modified from [8, Chapter 8.2].

Graphical models encode dependencies among the random variables and thus offer a way to factorize the joint probability distribution into a product of simpler "local" distributions, that depend only on a subset of variables. As a result, the inference of graphical models decomposes into inferring the dependency structure on the one hand, and learning the individual parameters of the local distributions on the other hand [8, Chapter 8]. In general, this leads to a simplification of the distributions that have to be learned, and hence reduces the amount of data needed for parameter inference. An example is provided in Fig. 1.1 where the graphical model allows the decomposition of a joint probability distribution of three random variables into three local distributions: $p(a, b, c) = p(a|c) \cdot p(b|c) \cdot p(c)$. If one assumes that every variable may take ten values, at least $10^3 = 1000$ measurements will be required to cover all possible realizations of the joint probability distribution. However, by taking into consideration the decomposition enabling the individual assessment of the local probabilities, significantly less measurements, max. $10^2 + 10^2 + 10 = 230$, are required. Repeated measurements for all possible events are the prerequisite for reliable inference, and so this reduction in the amount of required data or the considerable increase in inference quality for the same amount of data, respectively, is tremendous. Yet, this factorization comes at a cost. An increase in the number of distributions can entail an increase in the number of hyperparameters that have to be learned. Reducing this number of hyperparameters and the complexity of the probability distributions too much, however, reduces the flexibility of the model, and leads to its oversimplification. Graphical models are applied in all three parts of this thesis: Every Nested Effects Model in Part II is a graphical model by itself. In Part III, the structure of the graphical model is provided by the structure of the kinetic systems. In Part IV, factor graphs are used to represent cellular decision processes. Taking advantage of the local probability structure, fast algorithms enable the efficient estimation of parameters in this class of graphical models.

1.3. Bayesian modeling

Bayesian modeling is a special case of statistical modeling. It aims at maximizing the posterior $P(\Theta|D) \propto L(\Theta) \cdot \pi(\Theta) = P(D|\Theta) \cdot \pi(\Theta)$, by including prior knowledge on Θ given as probability distribution prior $\pi(\Theta)$. [37, Chapter 8.3]. So, defining a model in the spirit of Bayesian statistics, as it is done in all three scenarios described in this thesis, includes defining a likelihood distribution and a prior. A more elaborate motivation for the use of Bayesian methods in model selection and parameter inference is provided in Section 2.2.

Choice of the prior distribution

The prior distribution needs to be chosen with utmost care. An overly strong prior may override or bias any evidence from the data, while a weak prior may result in an unnecessarily disperse posterior distribution.

The definition of the prior was particularly important in Part III for the quantitative analysis of processive RNA degradation by the archaeal exosome: It is reasonable to assume that the decay efficiency is similar for RNA molecules of consecutive lengths. This constraint could be incorporated by defining of a smoothness prior which had a strong effect on the prediction quality. In graphical models, sparseness of edges is a common assumption which greatly simplifies the learning task. The beneficial effect of a sparseness prior is demonstrated in Part II. Apart from knowledge-driven approaches for prior specification, data-driven approaches exist. One of them is the Empirical Bayes method which was applied in Part II (see Sections 6.2.2 and 7.2.1).

Choice of the likelihood function

The choice of an error model has been crucial in Part III. There, the definition of the likelihood necessitates the specification of the expected measurement errors which are not known *a priori*. We thus developed an adaptive likelihood MCMC where the error model is updated regularly during the sampling process.

1.4. Bias and variance

Basically, three types of errors play a role in the model selection process. They are illustrated in Fig. 1.2. The model bias is the mathematical counterpart to George E. P. Box's appropriate statement: "Essentially, all models are wrong, but some are useful" [13, p. 424]. It arises from the fact that each model explains only certain aspects of reality and will in some way or other differ from the truth. Thus, the model bias describes the difference between reality and the best model in the selected model class. Since the task of finding the best model within a given model class can itself be very difficult if not practically impossible, the learning algorithm may return a model which is not identical to the best one. The estimation bias thus extends the model bias by quantifying the difference between the best fit and the average estimated model within the model class. The combination of model bias and estimation bias is referred to as bias. The bias is a systematic error, and knowing about it allows to correct for it and to keep it in mind when interpreting the results [37, Chapter 7].

In contrast, the variance is an unsystematic error that originates from the variability within the data (random fluctuation, e.g., due to measurement errors and replicates). It leads to a variation within the model space, and thus describes the expected deviation of individual predictions from the mean (standard deviation) [37, Chapter 7]. It can be assessed through repeated estimations based on re-sampled data (bootstrapping), however, it is not possible

to correct for it [50, Chapter 4.2]. Nevertheless, the quantification of a model's variance is important to assess the reliability of the predictions made by the model. To allow for this variability during parameter estimation, an appropriately chosen error model has to be defined. In Part III, this error model could not be determined beforehand and so an adaptive likelihood approach with a stepwise adaption of the error model has been developed (see Section 12.1).

A good model has to find a trade-off between its bias and its variance. High complexity leads to a good fit of the training sample, i.e., a low bias, but to a high dependence on this dataset, i.e., a high variance. In contrast, low complexity leads to more robust predictions, i.e., a low variance, but to a potentially less precise fit of the training sample, i.e., a high bias. To avoid overfitting, model design and model fitting have thus to be conducted with utmost care, and the incorporation of an independent test sample is indispensable to find a compromise between low bias and low variance [37, Chapter 7].
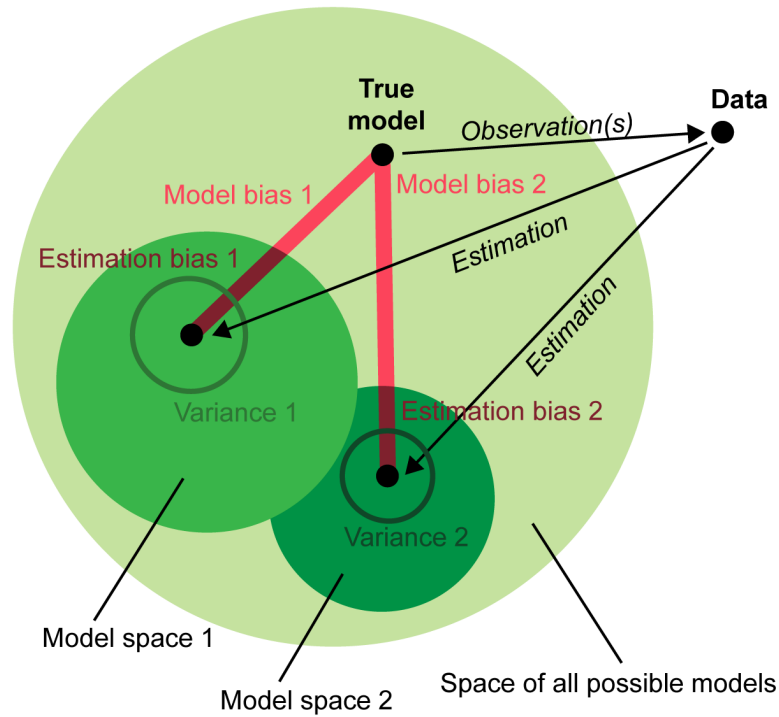
Figure 1.2.: Model bias (light red line), estimation bias (dark red line), and variance (green circles) are the three major errors in the model selection process. A detailed introduction to the topic is provided in Section 1.4. Here, they are illustrated based on a true model and two different model classes (model space 1 and model space 2), which are both more or less appropriate to describe some observations (data) generated by the true model. Different parametrization leads to a model class of higher complexity (model space 2; lower variance but higher bias) and to one of lower complexity (model space 1; lower bias but higher variance). The black dots within the small green circular areas represent the average estimate within a given model class. Depending on the quality of the estimation procedure, the average predictions can deviate significantly from the best model (transition point from the light red line to the dark red line).

2. Parameter inference

When the mathematical model structure is defined (e.g., the system of ordinary differential equations in Part III), the parameters (e.g., the decay rates in Part III) can be identified based on the measured data (e.g., the amount of RNA of different lengths at different time points in Part III). If the model is correct (and identifiable), the data is free of measurement errors, and enough data (including all essential information) is available, one can simply identify exactly one set of parameters that is able to reproduce the observed data. However, as already stated before, this is rarely the case in real-world scenarios, and mathematical methods have to be applied to derive parameter estimates which approximate the true values at the best. Estimating model parameters from noise-containing observations is called an inverse problem [50, Chapter 4.2].

2.1. The frequentist approach

The standard frequentist approach to parameter identification is to find a parameter set that maximizes a likelihood function $L(\Theta) = P(D|\Theta)$, i.e., the maximum likelihood estimate $\hat{\theta}_{\mathrm{ML}} = \arg\max_\theta L(\theta), \theta \in \Theta$. Under the assumption of independent observations with Gaussian measurement errors, this function is the sum of the squared errors, and the resulting approach is well-known as least squares estimation. The question posed by this approach is rather counterintuitive: "How should the parameters look like to make the data more probable?" [50, Chapter 4.2]. In this scenario, the likelihood is not considered as a probability distribution and the parameters are treated as fixed but unknown. The data however is treated as reproducible, and probability is seen as frequency based on a large number of observations [9, Chapter 1].

Confidence intervals

It is pointless to assign probabilities to parameters that are treated as fixed [9, Chapter 1]. This has a major influence on the interpretation of confidence intervals: A confidence interval of confidence level $x\%$ only states that, for many repeated estimations based on different datasets, $x\%$ of the calculated confidence intervals include the true parameter. A common misinterpretation is to say that the parameter is inside the interval with a probability of $x\%$. This statement would contradict the basic assumption that the parameter is fixed and thus either is inside the interval or not [65, Chapter 37].

Nuisance parameters

In some situations, only a subset Θ_1 of the model parameters are interesting. If this is the case, it is important that the remaining parameters (Θ_2, the nuisance parameters) do not interfere with the estimation process. The frequentist approach incorporates the nuisance parameters by calculating the profile likelihood

$$L_P(\Theta_1) = \sup_{\Theta_2|\Theta_1} L(\Theta_1, \Theta_2),$$

which results in

$$L_P(\Theta_1) = L(\Theta_1, \hat{\Theta}_2|\Theta_1).$$

However, simply plugging the maximum conditional likelihood values into the joint likelihood function does not take into account uncertainties about the nuisance parameters [9, Chapter 1].

Drawbacks of the approach

Typically, an inverse problem is ill-posed, i.e., a solution does not necessarily exist, and if one exists it does not have to be unique, and it may vary unstably in response to small changes in the measurements. A sound parameter estimation procedure therefore needs to address all these issues by not only producing one single parameter fit, but by providing additional information about the goodness of this fit and the variability of the estimation process [50, Chapter 4.2].

This is not met by the frequentist approach to parameter estimation: Even though the likelihood function is helpful to verify that the model can approximate the data, it only chooses the (single) best-fitting parameter set and nothing can be said about the uniqueness of the solution or the variability of the parameter estimates. On the contrary, the Bayesian approach provides a comprehensive characterization of the posterior distribution of the parameters given the data. This distribution can be used to assign probabilities to any hypothesis about the parameters or their relations [50, Chapter 4.2].

2.2. Bayesian parameter estimation

The Bayesian viewpoint to parameter estimation considers the probability as a degree of believe, and the parameters as random variables which depend on the given data and which may vary according to our prior beliefs. In other words, starting with a pre-defined prior parameter distribution, the data is used to update our beliefs, and to arrive at a posterior parameter distribution which contains all information on the parameters that can be learned from the data [9]. In particular, in Part III one can answer questions like: "Are the association and dissociation parameters for one RNA length dependent?" or "Is the parameter distribution of the catalytic efficiency narrow or wide?" in a precise way by calculating statistical measures of dependence or dispersion, respectively. If two posterior distributions from two different

experiments are available, one can even compare models, e.g., by asking for the probability that the catalytic efficiency for a given RNA length in experiment 1 is higher than in experiment 2. A comparison between least-squares fitting and Bayesian parameter estimation based on the RNA decay model is provided in Section 13.1.6.

The posterior distribution

The posterior distribution can be derived from the likelihood and the prior using the Bayes theorem [9, Chapter 1]:

$$P(\Theta|D) = \frac{P(D|\Theta) \cdot \pi(\Theta)}{P(D)} = \frac{P(D|\Theta) \cdot \pi(\Theta)}{\int P(D|\Theta) \cdot \pi(\Theta)d\Theta}$$

Due to the fact that the integral can be difficult to evaluate and that, in general, it is not necessary to know the exact posterior, the following approximation which provides the same shape information is used in most applications [9, Chapter 1]:

$$P(\Theta|D) \propto P(D|\Theta) \cdot \pi(\Theta)$$

Compared to the frequentist approach, the question posed here is rather intuitive: "Given the data, which is the most probable parameter?" [50, Chapter 4.2]. In this setting, the analogue of the maximum likelihood estimate is the maximum *a posteriori* estimate, a parameter set which maximizes the posterior. However, the Bayesian spirit is better met by taking into account the mean of the posterior distribution which minimizes the mean-squared error loss function [9, Chapter 1].

Credible intervals

Unlike the confidence interval in the frequentist approach, the Bayesian credible (or confidence) interval can easily be interpreted and offers the statement which is usually desired: Based on the posterior distribution, a $x\%$ credible interval can be calculated which actually implies that the true parameter is inside this interval with a probability of $x\%$ [9, Chapter 3].

Nuisance parameters

Unlike the frequentist likelihood, the Bayesian posterior is a probability distribution, and thus the incorporation of nuisance parameters differs significantly from the frequentist approach. Here, the marginal posterior for the parameters of interest Θ_1 can be calculated by integrating out the nuisance parameters Θ_2:

$$P(\Theta_1|D) = \int P(\Theta_1, \Theta_2|D)d\Theta_2$$

This approach allows for all uncertainties with regard to the nuisance parameters [9, Chapter 1]. Integrating out parameters can be time-consuming and thus methods like the Expectation-Maximization (EM) algorithm have been developed for a more efficient approximation. We developed an EM algorithm for Nested Effects Models (NEMs) in Part II to deal with their two-part network structure. More details are provided in Section 7.1.

2.3. The need for computational methods

The main obstacle to Bayesian parameter estimation is that generally the posterior distribution cannot be derived analytically and that even a numerical analysis is often infeasible. This long-standing problem was brilliantly solved by Markov Chain Monte Carlo (MCMC) sampling, one of the major breakthroughs in 20th century statistics. MCMC outputs a sequence of parameter sets (Markov chain) whose empirical distribution, for long sequences, approximates (converges to) the posterior distribution [9, Chapter 1].

2.4. Markov Chain Monte Carlo sampling

Markov Chain Monte Carlo sampling provides an elegant way to assess the parameters of a model, even if the corresponding posterior distribution is not accessible analytically. It outputs a sample of parameters whose empirical distribution, for long sequences, converges to the true posterior. Thus, any question that one might ask about the posterior parameter distribution can, in theory, be answered by looking at a corresponding Markov chain [9, Chapter 2].

2.4.1. Monte Carlo simulation

The idea for today's Monte Carlo simulation traces back to 1946, when Stan Ulam tried to figure out the chances to win a particular solitaire laid out with 52 cards. As calculations turned out to be complicated and exhausting, he had the idea to just play several times and count. This principle, approximating a complex combinatorial problem by the much easier process of drawing samples, is the basic idea of Monte Carlo simulations [2, Chapter 1].

Monte Carlo approaches aim at approximating a target density $p(x)$, $x \in X$ (with X being a high-dimensional space) by generating an independent and identically distributed (i.i.d.) set of samples $\{x^{(i)}\}_{i=1}^{N}$. This set of samples can then be used to estimate, for example, integrals or maxima of the target function. For simple forms of $p(x)$, straightforward sampling routines are available. In all other situations, i.e., in most real-world applications, more elaborate techniques such as Markov Chain Monte Carlo Sampling are required [2, Chapter 2].

2.4.2. Markov chains

A Markov chain is a stochastic process which yields a sequence of states where one state depends only on the directly preceding one (the so-called Markov property):

$$P(x^{(i)}|x^{(i-1)}, ..., x^{(1)}) = P(x^{(i)}|x^{(i-1)})$$

Possible transitions between the states are specified by a transition matrix

$$T(x^{(i)}|x^{(i-1)}), \text{ with } \sum_{x^{(i)}} T(x^{(i)}|x^{(i-1)}) = 1 \, \forall i$$

If $T \triangleq T(x^{(i)}|x^{(i-1)})$ remains invariant for all i the Markov chain is called homogeneous. A distribution $p(x)$ is called invariant if the transition matrix is constructed such that after several steps and for any starting point the chain converges to this distribution. This is exactly the behavior which is desired if MCMC sampling is used to approximate a posterior distribution that can't be assessed otherwise. To induce an invariant distribution, the stochastic, homogeneous transition matrix T has to be irreducible and aperiodic. Irreducibility means that any state can be reached from any other state at some point, aperiodicity means that the chain won't get caught in cycles. The detailed balance condition (or reversibility) is a sufficient but not necessary condition for the invariance of a target distribution $p(x)$:

$$p(x^{(i)})T(x^{(i-1)}|x^{(i)}) = p(x^{(i-1)})T(x^{(i)}|x^{(i-1)})$$

Thus, by ensuring detailed balance, it is possible to ensure that a target distribution $p(x)$ is invariant [2, Chapter 3].

2.4.3. MCMC sampling using the Metropolis-Hastings algorithm

MCMC sampling combines the Monte Carlo principle of approximating a distribution by drawing random samples with the principle of Markov chains, which offers a mathematical framework to ensure that the derived sample has the desired properties. In this setting, the unknown parameters are the states of the Markov chain, and a proposal function that suggests a new set of parameters based on the current one replaces the transition matrix. The main challenge is to ensure that the Markov chain and the proposal function fulfill the required properties such that the desired posterior distribution is the invariant distribution of the chain. To this end, various methods exist. One of them is the Metropolis-Hastings algorithm which I will introduce in the following. The combination of these methods allows to approximate the posterior distribution even if it is not possible to sample from it directly.

The standard algorithm

The Metropolis-Hastings algorithm has first been suggested in 1953 [70] and further extended in 1970 [38]. Let Θ be the set of unknown parameters, $q(\Theta_n \to \Theta_{n+1})$ the proposal function, $L(\Theta) = P(D|\Theta)$ the likelihood function, and $\pi(\Theta)$ a predefined prior. The Markov chain is created by starting with an initial set of parameters, and then repeatedly suggesting a new one and either accepting or rejecting it by turns. The proposal / acceptance steps are repeated until the chain has converged, and a sufficiently large sample has been derived. This procedure is delineated in the following:

---

1. Initialize Θ_0
2. Proposal step: Given Θ_n , draw a candidate Θ' from the proposal distribution $q(\Theta_n \to \Theta')$
3.1 Calculate the quantity $A = \frac{L(\Theta')}{L(\Theta_n)} \cdot \frac{\pi(\Theta')}{\pi(\Theta_n)} \cdot \frac{q(\Theta' \to \Theta_n)}{q(\Theta_n \to \Theta')}$
3.2 Acceptance step: With probability $\min(A, 1)$, let $\Theta_{n+1} = \Theta'$ (accept). Otherwise, let $\Theta_{n+1} = \Theta_n$ (reject)
4. Increment n by one and repeat steps 2. and 3. until convergence

---

The quotient in step 3.1 allows to decide upon acception / rejection of the newly suggested parameter set based on the true posterior distribution without actually requiring the normalizing factor:

$$\frac{P(\Theta'|D)}{P(\Theta_n|D)} = \frac{\frac{L(\Theta') \cdot \pi(\Theta')}{P(D)}}{\frac{L(\Theta_n) \cdot \pi(\Theta_n)}{P(D)}} = \frac{L(\Theta') \cdot \pi(\Theta')}{L(\Theta_n) \cdot \pi(\Theta_n)}$$

It can easily be shown that a chain generated by this mechanism fulfills all requirements (detailed balance with respect to $P(\Theta|D)$, aperiodicity, and irreducibility), and actually converges to the desired posterior distribution [2, Chapter 3].

Requirements, challenges and pitfalls

A MCMC approach has to be designed such that its invariant distribution is the target distribution, and that it converges quickly to this distribution [2, Chapter 3]. In the following I will discuss some practical implications.

Initialization   The Markov Chain has to converge to the invariant distribution independent of the initially chosen parameter set [2, Chapter 3]. When setting up a MCMC sampling approach, this has to be verified in simulation runs.

Choice of the proposal function   An appropriate choice of the proposal function is crucial for the convergence properties of the Markov chain. If the proposal is too wide, the attempted jumps will be too large and the rejection rate might be very high. This would result in high correlations between the states which disagrees with the Markov property. If, in contrast, the proposal is too narrow, the chain will not be able to explore the whole parameter

space [2, Chapter 3]. A method to assess this so-called mixing behavior of the chain is introduced in Section 13.2.2 and Fig. 13.15 in Part III. In this application, the proposal function is adjusted by setting the standard deviation of a log-normal distribution appropriately. In Part II, the width of the proposal function corresponds to the number of signals graph edges that are changed in one sampling step (see Section 7.2.2).

Convergence speed, chain length and burn-in phase   The convergence speed of the Markov chain is measured by the number of steps it takes for the chain to reach its stationary distribution (the so-called burn-in phase). The standard visual control is offered by a convergence plot which displays the trace of each parameter along the sampling procedure, see Fig. 13.5 in Part III for an example. After convergence, in the so-called stationary phase, the variation of the chain does not decrease any further. Only the stationary phase reliably approaches the probability distribution, and so the burn-in phase is discarded.

The chain has to be long enough to converge to the target distribution, and to produce enough samples for the subsequent analysis. At the same time, the number of steps are obviously subject to computational restrictions. A reasonable trade-off has to be found in simulation runs.

2.4.4. Other MCMC approaches

MCMC without likelihood

MCMC sampling based on the Metropolis-Hastings algorithm allows to approximate a posterior distribution even if it is not accessible analytically. Yet, it must still be possible to calculate the likelihood of the parameters. For situations where this is not the case, a Markov Chain Monte Carlo without likelihoods approach has been developed [66]. In this approach, the likelihood of the parameters is replaced by the quality of so-called summary statistics. The summary statistics are calculated from datasets that are simulated based on the parameters in question, and describe meaningful features of the data. Comparing them to the summary statistics that have been calculated from the observed data allows to approach the probability of the underlying parameters.

We tried this approach for the analysis of the hematopoietic stem cell differentiation process (Part IV) based on various summary statistics (e.g., the branch size or the relative frequency of double-death siblings). Yet, the approach actually described in the corresponding part of this thesis turned out to be more successful.

Reversible Jump MCMC sampling

The Metropolis-Hastings approach as described above can only be used when model selection has already been completed. If the choice of the best model should be incorporated into the sampling process, the method has to be extended. The reversible jump algorithm yields a Markov chain where $p(m, \Theta_m)$ is the invariant distribution, with $M = \{M_m\}_{m=1,...,N}$ being a

family of models and Θ_m being the corresponding parameter set. It includes steps where the model class is changed (so-called model jumps), as well as steps where only the parameters within the same model class are updated. The main obstacle is that probabilities for model classes with different dimensions can't be compared directly, which has to be considered in the acception step. An efficient solution for this problem has been developed in 1995 [33]. The acception step is adapted such that it allows jumps between different models. [2, Chapter 3]. More details are provided in the Appendix, Section C.1.

This approach was used in Part IV where the MCMC sampling includes jumps between the selective and the instructive scenario for hematopoietic stem cell differentiation. In the selective scenario, differentiation is due to varying cell death rates, while the instructive scenario implies varying differentiation rates. The respective other parameter is the same for all lineages that are included in the model.

3. Importance of simulation

Simulations are of tremendous importance both during model selection and parameter inference. Whether a model is appropriate and whether the designed parameter estimation process (e.g., MCMC sampling) works properly can't be evaluated based on the observed data only. In a simulation scenario, a realistic "true" parameter set is chosen, and noise-containing measurements are simulated based on this set. In this way, as many datasets as desired can be produced. Some major advantages of simulation runs prior to the actual analysis of the observed data are summarized in the following:

Model Selection

- If it is unclear whether all parameters are identifiable in the model, simulations help to reveal parameter dependencies. See Part III, Section 13.1.1 for an example.

- Bias and variance can be assessed if data exists for which the true parameters are known. See Part III, Section 13.1.4 for an example.

Parameter inference / MCMC sampling

- Simulations serve to assess the convergence properties of the Markov chain. Multiple, differently initialized Markov chains are run in order to guarantee a sufficiently fast convergence as well as an appropriate mixing of the chain. Importantly, the length of the burn-in phase can be determined, and independence of the initialization can be tested.

- The parameters of the proposal function can be determined.

- The prediction quality of the approach (e.g., sensitivity and specificity) can only be determined if the true parameters are known. See Part II, Fig. 8.5 for an example.

Part II.

MC EMiNEM maps the interaction landscape of the Mediator

4. Introduction

The Mediator is a highly conserved, large multiprotein complex that is involved essentially in the regulation of eukaryotic mRNA transcription. It acts as a coactivator by integrating regulatory signals from gene-specific activators or repressors to the RNA Polymerase II. The internal network of interactions between Mediator subunits that conveys these signals is largely unknown.

Active interventions into the cellular system followed by phenotypic measurements, as opposed to purely observational data, provide insight into the functions and interactions of the respective gene products. Along this line, perturbation experiments have been carried out with low-dimensional readouts (such as cell viability or growth [31, 108]) as well as with high-dimensional phenotypes (such as genome-wide expression or DNA binding measurements [41, 43]). While the reconstruction of regulatory networks from observational high-dimensional gene expression data has been investigated thoroughly (e.g., [5, 88, 89]) the statistical analysis and interpretation of perturbation data is an active field of research [28, 109]. Here, we introduce MC EMiNEM, a novel method for the retrieval of functional dependencies between proteins that have pleiotropic effects on mRNA transcription. MC EMiNEM is an efficient and robust learning algorithm for Nested Effects Models (NEMs), a class of probabilistic graphical models that extends the idea of hierarchical clustering. MC EMiNEM combines mode-hopping Monte Carlo (MC) sampling with an Expectation-Maximization (EM) algorithm for NEMs. A meta-analysis of four Mediator perturbation studies in *Saccharomyces cerevisiae* provides new insight into the Mediator signaling network. In addition to the known modular organization of the Mediator subunits, MC EMiNEM reveals a hierarchical ordering of its internal information flow, which is putatively transmitted through structural changes within the complex. We identify the N-terminus of Med7 as a peripheral entity, entailing only local structural changes upon perturbation, while Med19 and the C-terminus of Med7 appear to play a central role. MC EMiNEM associates Mediator subunits to most directly affected genes, which, in conjunction with gene set enrichment analysis, allows us to construct an interaction map of Mediator subunits and transcription factors.

The content of this part has been published in Niederberger *et al.* [77]. Some sections refer to the supplementary material of this paper which are provided in digital form along with this thesis. The MC EMiNEM method is freely available as a part of the **R**/*Bioconductor* package *nem* [25, 30, 44]. All datasets have been provided by members of the Cramer lab at the Gene Center.

5. Biological background

Phenotypic diversity and environmental adaptation in genetically identical cells are achieved by an exact tuning of the cell's transcriptional program. A variety of components contributes to this program, including the polymerase, general transcription factors, coactivators, gene specific transcription factors, and promoter elements. Unraveling parts of the complex network of involved components and associated interactions is a challenging task. Here, we shed light on the role of the Mediator complex in transcription regulation in yeast.

5.1. The yeast Mediator complex

The Mediator, first discovered in 1994 [49,51], is a large multiprotein complex which is highly conserved in eukaryotes [11]. It is a coactivator acting as an interface between gene-specific transcription factors (TFs) and the core transcription machinery (e.g., Polymerase II (Pol II)), and it is required for basal transcription as well as for activated transcription or repression [17,53,58]. Despite its importance and even though, in the last years, many successful efforts have been made to gain insight into both structural and functional aspects [10,45,55,56], large parts of its structure and function are still unknown. This is mainly due to the large size of the Mediator, as well as to its complexity and flexibility.

Structure

Yeast Mediator consists of 25 subunits with a total molecular weight of more than 1 MDa. It is organized in 4 different modules (head, middle, tail, and kinase module) which are supposed to contribute in different ways to the overall function of the Mediator. A schematic representation of the whole complex is provided in Fig. 8.9, already including the results of this study.

Recently, an excellent review of the state-of-the-art understanding of both Mediator structure and function has been published [58]. It states that, at the moment, atomic structures are available for (parts of) 13 subunits which is less than 20% of the total structure. The head module is best characterized while only few is known about the tail module. It is striking that most of the folds observed in Mediator do not appear in other parts of the transcription machinery and that some of them are duplicated within the Mediator (e.g., a four-helix bundle in Med11/Med22 and Med7/Med21), suggesting the existence of common building blocks. The structural studies also reveal the existence of functional submodules and the corresponding flexible linkers connecting them to the rest of the Mediator. These include for example Med7N/Med31 [56] and Med8c/Med18/Med20 [59]. Furthermore, the review

discusses evidence for extensive structural changes of the Mediator complex upon activator and Pol II binding. These structural changes seem to vary for different activators and are supposed to promote Pol II binding as well as additional interactions with transcription-related proteins. In particular, interactions with TBP and TFIIH have been reported suggesting that the Mediator stabilizes the preinitiation complex.

Function

The Mediator contains so-called activator-binding domains (ABDs) which interact with the transactivation domains (TADs) of the transcriptional activators. It has been shown that some subunits contain several ABDs and that some TADs can interact with diverse ABDs which is supposed to be enabled by the conformational flexibility of the TADs [58]. This variance of possible interactions induces a great diversity of gene-specific effects, some of which are reviewed in [10].

The diverse roles of the Mediator in transcription regulation are supposed to include transcription initiation (facilitation of the preinitiation complex (PIC) formation by Pol II recruitment to the core promoter, PIC stabilization), promotion of transcription elongation (by elongation factor recruitment), or transcript processing (by stimulation of Pol II CTD phosphorylation) [10, 17]. The tail module is thereby believed to establish the contact to the gene-specific transcription factors while the head and middle module apparently contact Pol II [93]. Consequently, the head module is highly conserved, whereas the tail module is the most evolutionary divergent one. This is in line with a high structural and functional variability of transcription factors among eukaryotes [17]. The role of the kinase module is unclear. It is not necessarily part of the complex and has long been considered as repressive since studies showed that its presence prevents the binding of Mediator and Pol II [22]. Recent studies, however, suggest activating roles, in particular, with respect to transcription elongation and the release of paused Pol II [17].

5.2. A meta-analysis of four Mediator perturbation studies

We combined expression profiles of *S.cerevisiae* Mediator subunit deletion mutants dMed2, dMed15, dMed20, dMed31 with data from intervention studies. Those comprise mutations of Med7 (N- and C-terminal deletion), and point mutants of Med10, Med19, and Med21 (for more details on the data, please refer to [77]). The raw data is available at ArrayExpress. Although there exist even more high-quality gene expression data of Mediator mutants (e.g., [3, 59]), we restricted our analysis to experiments that were obtained on the Affymetrix yeast 2.0 array under similar environmental conditions. Some data are redundant in different experiments which enabled us to correct for batch-specific effects and to remove outlier genes (for data pre-processing, see Appendix Section A.4.1). After removing uninformative genes, this results in a total of 9 perturbations and ~2500 measurements.

6. A model for the Mediator signaling network

6.1. Nested Effects Models

Nested Effects Models (NEMs) are probabilistic graphical models designed for the analysis of gene expression perturbation screens $[1, 24, 26, 67, 68, 102, 105, 110]$ (see [27] for a summary) by reconstructing the dependency structure of the perturbation signals. They perform particularly well if this structure is hierarchical [68] and have so far been applied successfully to the ER-α pathway of human MCF-7 breast cancer cells [27] and to a signaling pathway in *Drosophila melanogaster* [67].

6.1.1. A two-part graphical model

The graph underlying a NEM contains two types of nodes: the perturbed entities (the signals $\mathcal{S}$, here: the Mediator subunits) and the genes for which expression has been measured (the effects $\mathcal{E}$). The edges of that graph describe the flow of regulatory information between the nodes. NEMs split this flow into two parts: the signals graph Θ containing the edges between the perturbed entities, and the effects graph H describing the assignment of the effect nodes to the signal nodes. We identify the graphs Θ and H with their respective adjacency matrices $\Theta \in \{0,1\}^{\mathcal{S} \times \mathcal{S}}$, $H \in \{0,1\}^{\mathcal{S} \times \mathcal{E}}$. The experimental data is summarized in an $\mathcal{S} \times \mathcal{E}$ matrix $D = (D_{jk})$, where D_{jk} corresponds to the expression data obtained from measurements of effect k upon perturbation of signal j. NEMs aim at reconstructing the signals graph, assuming a particularly simple regulatory structure: The perturbation of a signal j implies the perturbation of other signals that are children of j. This in turn perturbs the effect nodes that are the children of the perturbed signals in the effects graph (see Fig. 6.1). In other words, the NEM predicts an effect of gene k upon perturbation in signal j exactly if there is a two-step path from j to k, i.e., if $(\Theta H)_{jk} > 0$.

6.1.2. Parametrization and probability model

These binary predictions of our model are then linked to the actual measurements by specifying a probability model for the individual effects gene measurements,

$$p_{jk} = P(D_{jk}|j \text{ has an effect on } k) = P(D_{jk}|(\Theta H)_{jk} > 0) \text{ , and}$$
$$q_{jk} = P(D_{jk}|j \text{ has no effect on } k) = P(D_{jk}|(\Theta H)_{jk} = 0)$$
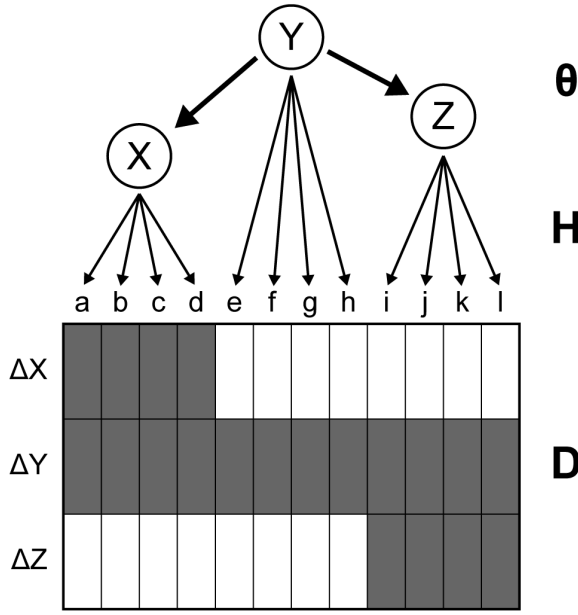
Figure 6.1: NEMs - an example. $\mathcal{S} = \{X, Y, Z\}$, $\mathcal{E} = \{a, b, c, ..., l\}$. Shaded matrix fields $D_{j,k}$ correspond to an expression change of effect gene k upon perturbation of signal j, white fields indicate no change in expression. The edges $Y \to X$ and $Y \to Z$ cause an effect in genes directly attached to signal X and Z respectively, when Y is perturbed.

There is extensive literature on the estimation of these two distributions, see [91, 97]. We adhere to the method proposed in [102]. Consequently, a NEM is parametrized by the tuple $(\Theta, H) \in \mathcal{M}_{\mathcal{S}} \times \mathcal{M}_{\mathcal{E}}$, where $\mathcal{M}_{\mathcal{S}}$ is the space of binary $\mathcal{S} \times \mathcal{S}$ matrices with unit diagonal, and $\mathcal{M}_{\mathcal{E}} \subset \{0, 1\}^{\mathcal{S} \times \mathcal{E}}$ is the space of effects graphs. We assume that the effects graph is *sparse*, such that each effect is linked to at most one signal (i.e., each column of $H \in \mathcal{M}_{\mathcal{E}}$ equals either a unit base vector of dimension n, or the null vector). It is convenient to transform the data matrix D into the log-odds matrix $R = (R_{jk}) = \log(\frac{p_{jk}}{q_{jk}})$. According to [102], the log posterior of the signals graph is given by

$$\log P(\Theta, H | D) = \text{trace}(\Theta H R^T) + \log \pi(\Theta, H) + \text{const} \tag{6.1}$$

For a derivation of Eq. 6.1, see also the Appendix, Section A.1.

6.1.3. Applying NEMs to the Mediator

In this application, the signals $\mathcal{S}$ correspond to the perturbed Mediator subunits, while the effects $\mathcal{E}$ correspond to the genes for which expression has been measured. The distinction between signals graph and effects graph allows the selective optimization of the regulatory structure among the Mediator subunits, and to make use of the underlying attachment of effects to signals at the same time. Due to an expected hierarchical structure of the transcriptional effects upon Mediator subunit perturbation (see Section 8.2.2, first paragraph), NEMs are the suitable model class for this analysis.

6.2. Prior choice

We assume edge-wise independent priors, $\pi(\Theta, H) = \pi^{\mathcal{S}}(\Theta) \cdot \pi^{\mathcal{E}}(H)$, and $\pi(\Theta) = \prod_{i,j} \pi^{\mathcal{S}}(\Theta_{ij})$, $\pi^{\mathcal{E}}(H_{\bullet k}) = \prod_k \pi^{\mathcal{E}}(H_{\bullet k})$.

6.2.1. The signals graph prior

In the absence of prior knowlege, a uniform prior is chosen (i.e., edge frequency $= 0.5$).

6.2.2. The effects graph prior

It is not obvious how the effects graph prior should be defined. Being most conservative, $\pi^{\mathcal{E}}$ can be chosen uniform, i.e., $\pi^{\mathcal{E}}(H) = \text{const}$ for all effects graphs $H \in \mathcal{M}_{\mathcal{E}}$. The posterior $P(\Theta|D)$ is then proportional to the marginal likelihood $P(D|\Theta)$. On the other side, upon availability of precise prior knowledge, $\pi^{\mathcal{E}}$ can be chosen deterministic, i.e.,

$$\pi^{\mathcal{E}}(H) = \begin{cases} 1 & \text{if } H = H_{\text{prior}} \\ 0 & \text{otherwise} \end{cases}$$

for some fixed adjacency matrix H_{prior}. In this case, the posterior is proportional to the full likelihood $P(D \mid H_{\text{prior}}, \Theta)$. As a trade-off between these two extremes, we initialize $\pi^{\mathcal{E}}$ in a data-driven fashion (based on R), namely

$$\pi_k^{\mathcal{E}}(H_{\bullet k} = v) \propto \begin{cases} \frac{p_{jk}}{p_{jk}+q_{jk}} = (1 + \exp R_{jk})^{-1} & \text{if } v = e_j, \ j \in S \\ \text{mean}(\frac{p_{jk}}{p_{jk}+q_{jk}} | j \in \mathcal{S}) & \text{if } v = 0 \end{cases}, \ k \in \mathcal{E} \qquad (6.2)$$

This prior will be updated regularly during the MCMC sampling in an Empirical Bayes procedure (see Section 7.2.1 for more details).

6.3. Structure learning in NEMs

The problem of structure learning in probabilistic graphical models is generally computationally hard (see [60]). A range of methods has been proposed for the maximization of Eq. 6.1. It has been observed that it is very difficult to estimate the effects graph H reliably. This is not surprising, since the adjacency matrix H has the same dimensions as the data matrix D. The main interest being the reconstruction of the signals graph Θ, several approaches try to maximize the (marginal) structure posterior $P(\Theta|D)$ by integrating out the hidden parameters H (for a methods review, see [27]). This marginalization however is a time consuming step that increases the complexity of the respective algorithms by at least a factor of $|\mathcal{E}|$, making the analysis of larger effects sets (such as in microarray studies) slow or even impossible. We avoid this drawback and develop an efficient Expectation-Maximization (EM) algorithm for the optimization of the NEM structure posterior (EMiNEM), which, even for large expression

datasets, is able to detect a local maximum within seconds. Since the landscape of the structure posterior is rugged (Fig. 6.2), we combine EMiNEM with mode-hopping Markov Chain Monte Carlo sampling (MC EMiNEM) for an efficient optimization of the structure posterior.
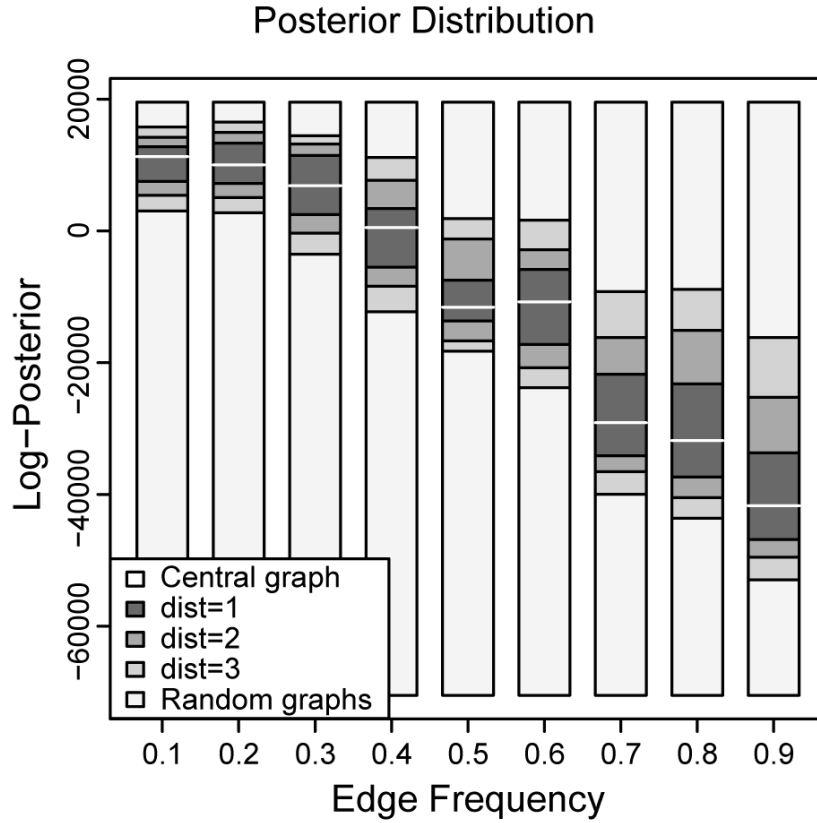
Figure 6.2.: NEM posterior distribution. This figure illustrates how the likelihood varies when only few edges (here: 1 to 3) are changed, based on randomly sampled fixed graphs (white lines) and relative to a fixed representative random graph sample (light gray). The underlying data is the real Mediator perturbation data, where Med10 and Med21 are combined to one signal node (i.e., $|S| = 9$), as a prior for the effects graph the data-driven prior has been used (according to the initialization of the MCMC sampling). On the x-axis, different graph densities are compared. The strong variation within very similar graphs demonstrates how rugged the landscape is. Given that the underlying data of this figure is the real data, the observed decrease of likelihoods following the increase of edge frequency yields extra information: It shows that the Mediator graph we are looking for tends to be rather sparse, which is a confirmation for the choice of the earlier mentioned sparseness prior during MCMC sampling.

7. Parameter estimation by EM and MCMC sampling

7.1. Finding a maximum *a posteriori* estimate with the EM algorithm

Throughout this section, the data D respectively the matrix R is considered given and fixed. We want to find the maximum *a posteriori* estimate $\hat{\Theta}$ for the signals graph,

$$\hat{\Theta} = \arg\max_{\Theta} P(\Theta|D) = \arg\max_{\Theta} \sum_{H \in \mathcal{M}_{\mathcal{E}}} P(\Theta, H|D) \tag{7.1}$$

This is the classical situation in which Expectation-Maximization is applicable [19]. For excellent introductions to the EM-algorithm, we recommend the tutorials of [18, 72, 76]. Briefly, given some guess Θ^t for $\hat{\Theta}$, the EM algorithm describes how to find an improved guess Θ^{t+1} such that the sequence $(P(\Theta^t|D))_{t=1,2,\ldots}$ is monotonically increasing, and converges (under mild additional assumptions that are met in our case) to a local maximum of $P(\Theta|D)$.

The expectation (E-)step of the EM algorithm involves calculating the expected log-posterior with respect to the distribution of H, given the current guess Θ^t:

$$Q(\Theta; \Theta^t) = \mathbb{E}_{P(H|D,\Theta^t)}\left[\log P(\Theta, H|D)\right] \tag{7.2}$$

The maximization (M-)step of the EM algorithm then consists of finding the maximizer $\Theta^{t+1} = \arg\max_{\Theta} Q(\Theta; \Theta^t)$. This is usually a much easier task than solving Eq. 7.1 directly. We derive an analytical solution, which leads to an efficient closed-form update step for Θ^{t+1}:

$$\Theta_{ab}^{t+1} = \begin{cases} 1 & \text{if } \sum_{k \in \mathcal{E}} R_{ak} \pi_{bk}^{\mathcal{E}} \exp((R^T \Theta^t)_{kb})(A_k)^{-1} + \tau_{ab} > 0 \\ 0 & \text{otherwise} \end{cases} \quad \text{for } a, b \in \mathcal{S} \tag{7.3}$$

A precise definition of the variables contained in Eq. 7.3, together with a detailed derivation of this formula is deferred to the Appendix, Section A.1, as it involves elementary but tedious calculations.

The EM algorithm is guaranteed to find a local maximum which, for unimodal distributions, equals the global optimum. In practice, the posterior landscape $P(\Theta|D)$ can be very rugged (see also Fig. 6.2). The outcome of the EM algorithm may therefore strongly depend on its
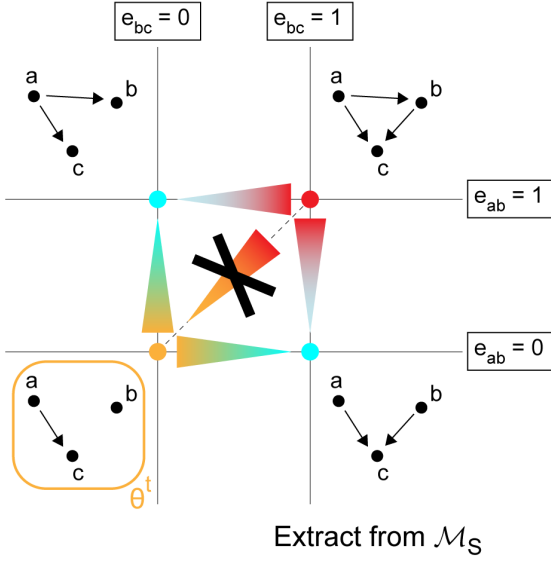
Figure 7.1: Pitfalls: local maxima. Considering an extract of $\mathcal{M}_{\mathcal{S}}$, where Θ includes the edge $a \rightarrow c$, two different states are possible: either both edges $a \rightarrow b$ and $b \rightarrow c$ are missing (medium probability, indicated by orange) or both of them exist (high probability, indicated by red). A graph which includes only one of them has a low probability (indicated by blue). Thus, based on a $\Theta^t = \{a \rightarrow c\}$, EMiNEM is not able to cross the low-probability states to arrive at the high probability state, changing only one edge at the same time.

initialization, and it may be far from the global optimum (see also Fig. 7.1). This raises the need to explore the set of local maxima provided by EMiNEM.

7.2. Sampling of the signal posterior's local maxima

Markov Chain Monte Carlo sampling offers a way to reliably explore the set of local maxima derived by EMiNEM. Similar so-called mode hopping optimization approaches have been established in [62,75,90,106], with applications in areas such as protein folding [15], nanocluster structure analysis [48] and reconstruction of signaling pathways [47]. A theoretical motivation is provided in Appendix Section A.2.1. For a general introduction to Markov Chain Monte Carlo sampling and its requirements and challenges (in particular, the importance of the features discussed in the following), please refer to Section 2.4.

The basic MC EMiNEM procedure is illustrated in Fig. 7.2. It deviates from the classical Metropolis-Hastings MCMC approach by adding an EM step to every acception/rejection step to restrict the estimation of parameters to the local maxima. In this context, $\Theta \in \mathcal{M}_{\mathcal{S}}$ refers to an element of the general signals graph space, while $\hat{\Theta} \in \mathcal{N}$ refers to an element of the space of local maxima. The details of the implementation are explained in Section 7.2.2.

7.2.1. An Empirical Bayes method for the effects graph prior

As outlined in Section 6.2.2, the effects graph prior is initialized in a data-driven manner. However, to the degree to which the Markov chain converges to the desired posterior distribution in the sampling procedure, we gain information on the signals graph structure. To incorporate this information into the remaining sampling steps, the prior is updated on a regular basis: In an Empirical Bayes approach, we iteratively estimate $P(\Theta|D)$ and $P(H|D)$, and use these distributions as priors for the estimation of the other quantity, respectively.
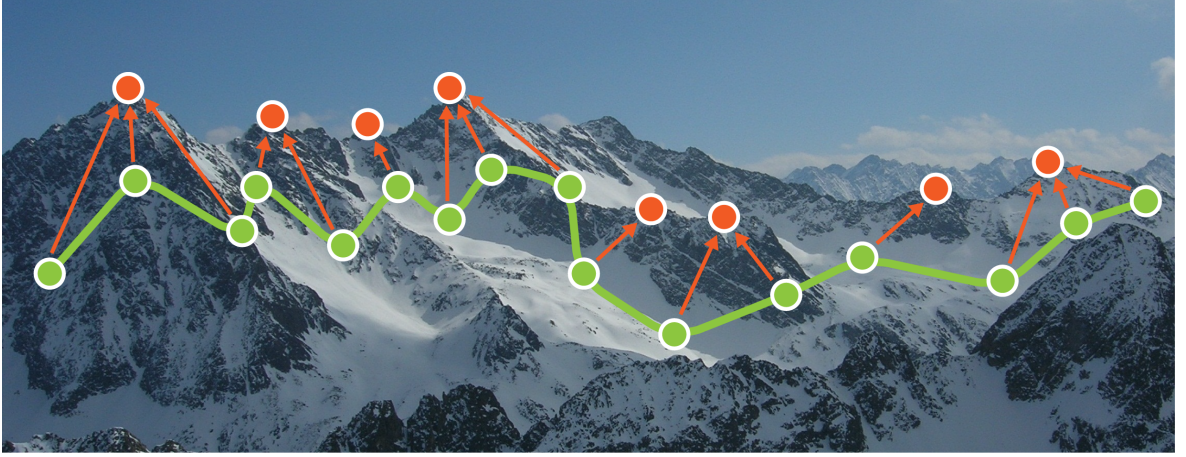
Figure 7.2.: The search strategy used by MC EMiNEM can be perfectly illustrated based on this mountain view from the top of the Hintere Karlesspitze in the Stubai Alps. The green dots represent the sampled signals graphs $\Theta \in \mathcal{M}_{\mathcal{S}}$ forming the underlying Markov chain (the green line). At every step the EM algorithm is applied to identify the corresponding local maxima $\hat{\Theta} \in N$ (the red dots). The sequence formed by the local maxima (one for every Θ in the Markov chain) is then used to assess the posterior distribution. This combination of Expectation Maximization and MCMC sampling offers a way to restrict the sequence derived from the sampling process to the most informative parameters.

Our Empirical Bayes procedure is:

1. Initialize $\pi^{\mathcal{E}}$ in a data-driven fashion (Eq. 6.2); choose $\pi^{\mathcal{S}}$ uniform.

2. Generate a representative sample $(\hat{\Theta}_i)_{i=1,2,\dots}$ from $\hat{P}(\Theta|D)$ by mode-hopping MCMC, given the prior distributions $\pi^{\mathcal{E}}$ and $\pi^{\mathcal{S}}$.

3. Replace $\pi^{\mathcal{E}}(H)$ by $\sum_j P(H|\hat{\Theta}_j, D)$, which is taken as an approximation for $P(H|D) = \sum_{\Theta \in \mathcal{M}_{\mathcal{S}}} P(H, \Theta|D) = \sum_{\Theta \in \mathcal{M}_{\mathcal{S}}} P(H|\Theta, D) \cdot P(\Theta|D)$. For more details, see Appendix Section A.2.2.

4. Repeat steps 2 and 3 until convergence (see Section 7.2.2 for details).

7.2.2. Implementation of the sampling procedure

Initialization

The chain is initialized with a randomly sampled signals graph Θ_{init}, based on a sparse edge frequency $(p_{\mathrm{edge}} = \frac{1}{|\mathcal{S}|})$.

Proposal function

Suggesting a new signals graph Θ' is based on the signals graph Θ_n of the previous step. Thus, the proposal function is independent of the local maxima. Simulation has demonstrated (results not shown) that randomly selecting $1.5 \cdot |S|$ edges and replacing them according to the predefined, sparse edge frequency $(p_{\mathrm{edge}} = \frac{1}{|\mathcal{S}|})$ yields a sufficient acceptance rate as well as a good mixing behavior of the chain.

Acception / Rejection

The newly suggested parameter is accepted (and added to the chain), if

$$
\begin{aligned}
\log(u) < \quad & \min(0, \left(\log L(\hat{\Theta}') + \log \pi(\hat{\Theta}') + \log q(\Theta' \to \Theta_n) \right) \\
& - \left(\log L(\hat{\Theta}_n) + \log \pi(\hat{\Theta}_n) + \log q(\Theta_n \to \Theta') \right) \\
& + w_{\text{sparse}} \cdot \left(\log \pi_{\text{sparse}}(\Theta') - \log \pi_{\text{sparse}}(\Theta_n) \right)) \quad , \text{ with } u \sim \mathcal{U}(0,1) \ ,
\end{aligned}
$$

otherwise it is rejected and the old parameter is added once again. Note that Θ is the signals graph suggested by the proposal function, while $\hat{\Theta}$ is the corresponding local maximum derived by EMiNEM, as explained in the main text. Thus, due to the inclusion of the EM algorithm, the suggestion of parameters refers to the whole signals graph space $\mathcal{M}_{\mathcal{S}}$ while the evaluation of the proposal refers to the space $\mathcal{N}$ of local maxima.

An additional prior $\pi_{\text{sparse}}(\Theta)$ for sparsity of the suggested graph is included and the corresponding weighting parameter $w_{\text{sparse}} = 0.5$ has been determined in simulation runs (variation of w_{sparse} did not change the results qualitatively, data not shown). For reasons of clarity, the sparsity prior is shown separately in an extra line.

Chain length, Empirical Bayes and burn-in phase

Each chain consists of 60000 steps. According to the Empirical Bayes procedure, the effects graph prior is updated every 5000 steps, which has proved to be most suitable in simulation runs (data not shown). Determining the burn-in phase is trivial: traceplots of the simulation runs showed that after well less than the 60000 steps the chain converges to one final $\hat{\Theta}$, i.e., the MCMC sampling can be seen as an additional EM algorithm (see Fig. 8.1). The MCMC runs of the Mediator data showed the same behavior (see Fig. 8.6). Thus, any information drawn from this final part of the sequence is good. However, for reasons of consistency, and since the effect gene attachment is updated every 5000 steps, only parameters according to the final attachment, i.e., the 5000 last parameters of the sequence are retained.

Best fit

The Markov chain provides an approximation of the posterior distribution of the parameters. We extract one "resulting" signals graph from this chain by weighting all edges by their frequency in the last 5000 steps and only retaining those that appear in at least 50%. In general, this marginalization might result in a loss of information because dependencies between edges are not considered any more. However, since the Markov chain in our case converges very fast to a unique, dominating signals graph which will then be extracted as the resulting graph, there basically is no marginalization and so this problem does not arise here.

8. Results & Discussion

8.1. Simulation

Datasets have been simulated as explained in Section A.3.1.

8.1.1. Assessment of the MCMC sampling behavior

Independence of initialization.

For six simulated NEMs, randomly chosen from two parameter settings (one with $|\mathcal{S}| = 8$ and $\beta - \mathrm{level} = 49\%$, the second with $|\mathcal{S}| = 11$ and $\beta - \mathrm{level} = 20\%$) 10 runs have been performed, each initialized with a different signals graph. For all six datasets, the ten results where the same, i.e., independent of initialization (data not shown).

Convergence

The convergence of the Markov chain has been verified in simulation runs. Traceplots for the example mentioned in Appendix Section A.3.1 (Fig. A.1, Fig. A.2) are shown in Fig. 8.1 (all edges) and Fig. 8.2 (selected edges). It is apparent that the sampled graphs comprise significantly more edges than the local maxima and that they vary strongly throughout the whole sampling process, while the local maxima vary slower in a more restricted model space and converge in the second half of the Markov chain. This is in line with the construction of the method and what has been discussed previously: Crucial for the success of the parameter estimation is the convergence of the Markov chain of local maxima, on which the calculation of the likelihood is based. Since several signals graphs from the underlying sequence of sampled graphs can yield the same local maximum they are not distinguishable, as long as they are in accordance with the general sparseness assumption.

Attachment of effects

The development of the attachment of effects to signal nodes during the Empirical Bayes procedure, again for the examples in Fig. A.1 and Fig. A.2, is visualized in Fig. 8.3. Obviously, some effects turn out to be rather deterministic, while others remain indecisive until the end of the Markov chain. The attachments predicted by MC EMiNEM are compared to the true ones in Fig. 8.4. While most of them agree very well, the attachments to signal node d are poorly reproduced. This corresponds to the missing edges in the predicted signals graph, which is depicted in Fig. A.2.

Sampled Graphs   **Local Maxima**

MCMC steps

Edges   Edges
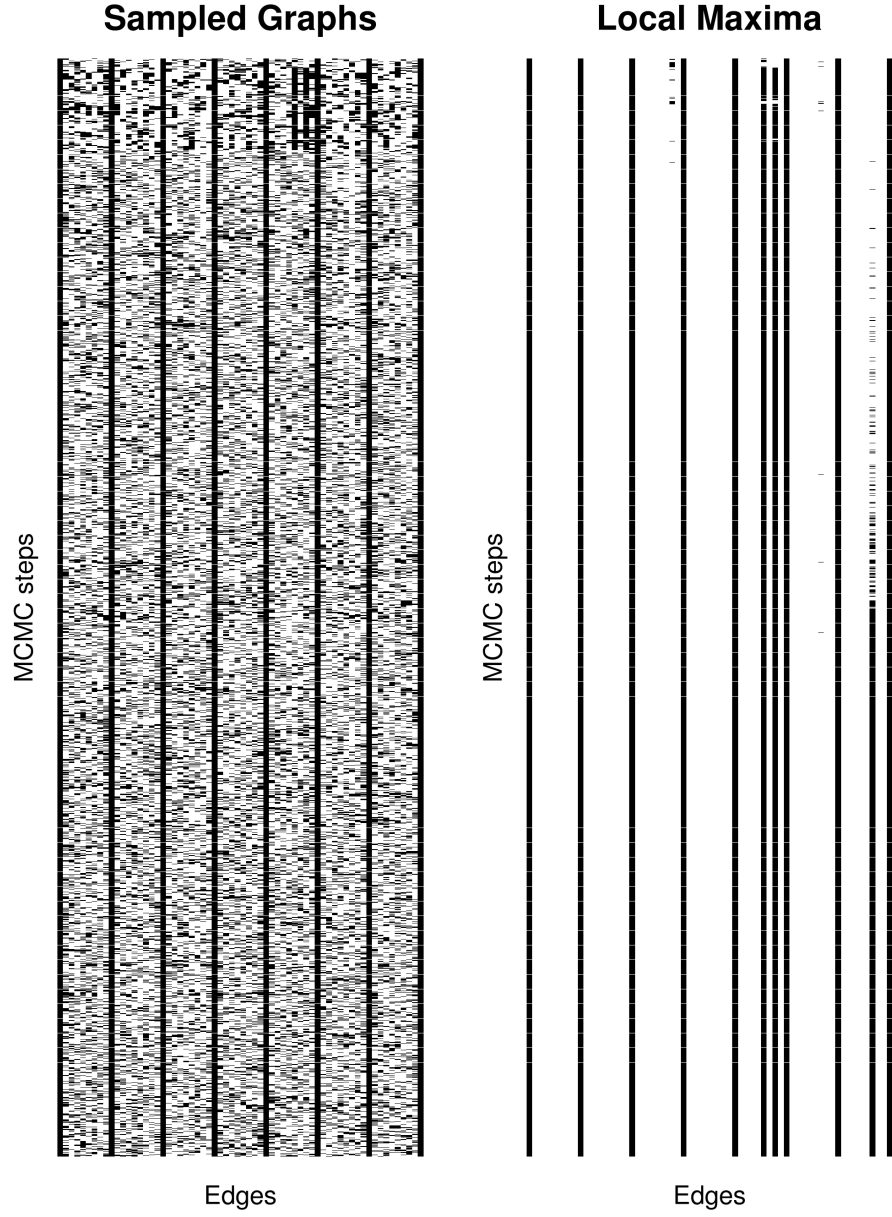
Figure 8.1.: Traceplot of all edges, simulation. Here, $|\mathcal{S}| = 8$ and $\beta - \text{level} = 49\%$. The left panel shows the traceplot for the sampled graphs $(\Theta_i)_{i=1,2,\ldots}$, the right panel shows the traceplot for the corresponding local maxima $(\hat{\Theta}_i)_{i=1,2,\ldots}$. The MCMC steps are depicted on the y-axis (from top to bottom), individual edges on the x-axis, thus, one line in the traceplot corresponds to the signals graph of the corresponding MCMC step. Black fields indicate the presence, white fields the absence of a given edge in a given MCMC step. Completely black columns represent self-loops, which are defined to be present in the mathematical formulation and included here for reasons of clarity. The sampled graphs comprise more edges and vary stronger, as compared to the sequence of local maxima. A discussion of this behavior is provided in the main text (Section 8.1.1).

Local Maxima

Sampled Graphs

Figure 8.2.: Traceplot of selected edges, simulation. Here, $|signals| = 8$ and $\beta - \text{level} = 49\%$. The upper panel shows the traceplots of selected edges in the sequence of local maxima $(\hat{\Theta}_i)_{i=1,2,...}$, the lower panel shows the traceplot of these edges in the sequence of the underlying sampled signals graphs $(\Theta_i)_{i=1,2,...}$. On the x-axis, extracts of the MCMC steps at the beginning $(1-1500)$ and the end $(57500-60000)$ of the chain are depicted. Selected edges (edges that vary in the sequence of local maxima) are depicted in different colors. Stacked on the y-axis are values of 0 and 1 for each edge, corresponding to the absence and presence of the edge at a given MCMC-step. As in Fig. 8.1, the sampled graphs comprise more edges and vary stronger, as compared to the sequence of local maxima. A discussion of this behavior is provided in the main text (Section 8.1.1).
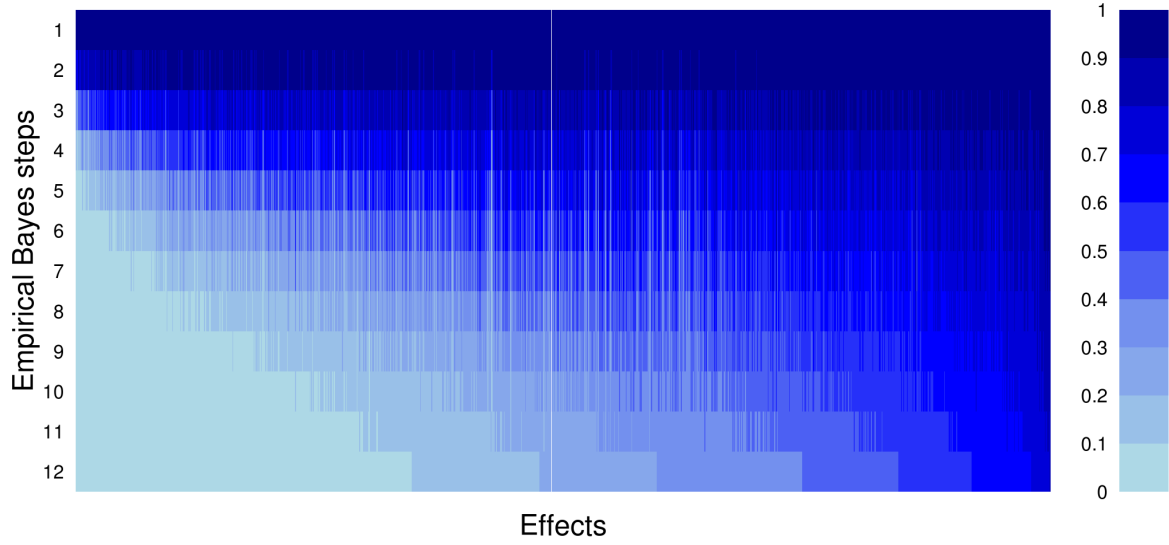
Figure 8.3.: Development of attachment entropy, simulation. For each effect j in each Empirical Bayes step l, the Shannon Entropy is calculated as $-\sum_{j \in \mathcal{S}} H_{jk}^l \cdot \log_2 H_{jk}^l$. On the y-axis, the Empirical Bayes steps are depicted (from top to bottom), on the x-axis, the effects are listed. The colors indicate the entropy, relative to the maximal one (when, for a given effect, the attachment probability is the same for any signal node (or no signal node at all)). Obviously, even though the initial prior for the effects graph is calculated according to the data matrix, the entropy is still very high. During the Empirical Bayes procedure, some effects turn out to be rather deterministic, while others remain flexible until the end of the Markov chain.
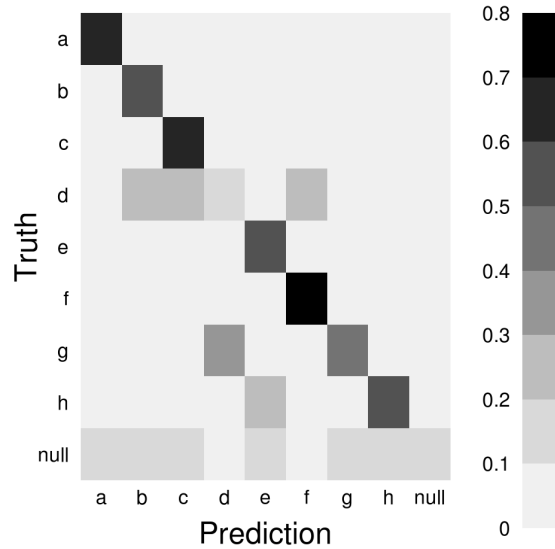
Figure 8.4.: Prediction quality for the effects graph. Here, an entry in row i and column k depicts the probability that an effect, attached to signal node i in the simulated model, is attached to signal node k in the predicted model (i.e., rows correspond to the true attachment, columns to the predicted one). Light gray corresponds to low probabilities, dark gray to high ones (see the scale on the right-hand side). The predicted attachment corresponds very well to the true one in most cases, except for effects attached to signal node d – which corresponds very well to the missing edges including d during the prediction the signals graph (Fig. A.2).

8.1.2. Prediction quality

The prediction quality was assessed in seven parameter settings for different noise levels and different numbers of signal nodes, with 1200 observed effect genes and a total number of $0.7 \cdot |\mathcal{S}|$ edges in the signals graph. For each of these scenarios, 50 NEMs were randomly sampled (for details see Appendix Section A.3.1). In each case, data were generated and afterwards analyzed with various methods: a simple EMiNEM approach without Markov Chain Monte Carlo sampling, the original NEM score [67], the Nessy method [102], and a random sampling approach (for details on the competing methods, see Appendix Section A.3.2). For all methods, the sensitivity strongly depends on the noise level and the number of signal nodes (Fig. 8.5A). MC EMiNEM performs best throughout all tested parameter settings, except for low noise where Nessy achieves a similar sensitivity. The specificity of all methods is very high, with a value above 98% in all scenarios.

8.1.3. Influence of the Empirical Bayes procedure

Our approach attempts to maximize the marginal posterior $P(\Theta|D)$. This quantity implicitly depends on the effects graph prior $\pi^{\mathcal{E}}(H)$. Therefore, we seek a prior for which the true signals graph Θ_{true} scores on the top end of the distribution $P(\Theta|D)$. It has been shown that NEM models are asymptotically consistent and identifiable [102], i.e., given the true effects graph as a deterministic prior $\pi^{\mathcal{E}}_{\text{true}}$, the true signals graph will score best. Thus, a well-chosen effects graph prior might greatly improve the prediction outcome. We tested the following priors: a deterministic prior according to the true effects graph, our Empirical Bayes prior (see Section 7.2.1), the data-driven prior used for the initialization of the MCMC sampling (see Section 6.2), and a uniform effects graph prior. The quality of an effects graph prior is assessed in two ways: First, we calculate the average L^1-distance between the prior $\pi^{\mathcal{E}}(H_{\bullet k})$ to the true prior $\pi^{\mathcal{E}}_{\text{true}}(H_{\bullet k})$, where $k \in \mathcal{E}$, and normalize it by dividing through the maximum gene-wise L^1-distance, which is 2. Secondly, we calculate the position of $P(\Theta_{\text{true}}|D)$ within the marginal posterior distribution $P(\Theta|D)$. Each posterior distribution was approximated by the empirical distribution of $P(\Theta|D)$ for a random sample of 5000 signals graphs. This was done for the 50 NEM samples that were generated in the most realistic simulation scenario (11 nodes, $\alpha = 0.05$, $\beta = 0.49$, see Fig. 8.5A. The results show that the Empirical Bayes prior approaches the true prior better than the other methods, according to the L^1-distance. Furthermore, the resulting posterior is better able to distinguish between signals graphs and to identify the true one (the true graph is located at the 99.1%, 99.4%, and 99.9% quantile for the uniform, data-driven and Empirical Bayes prior, respectively, and at the maximum for the true effects graph; see Fig. 8.5B.
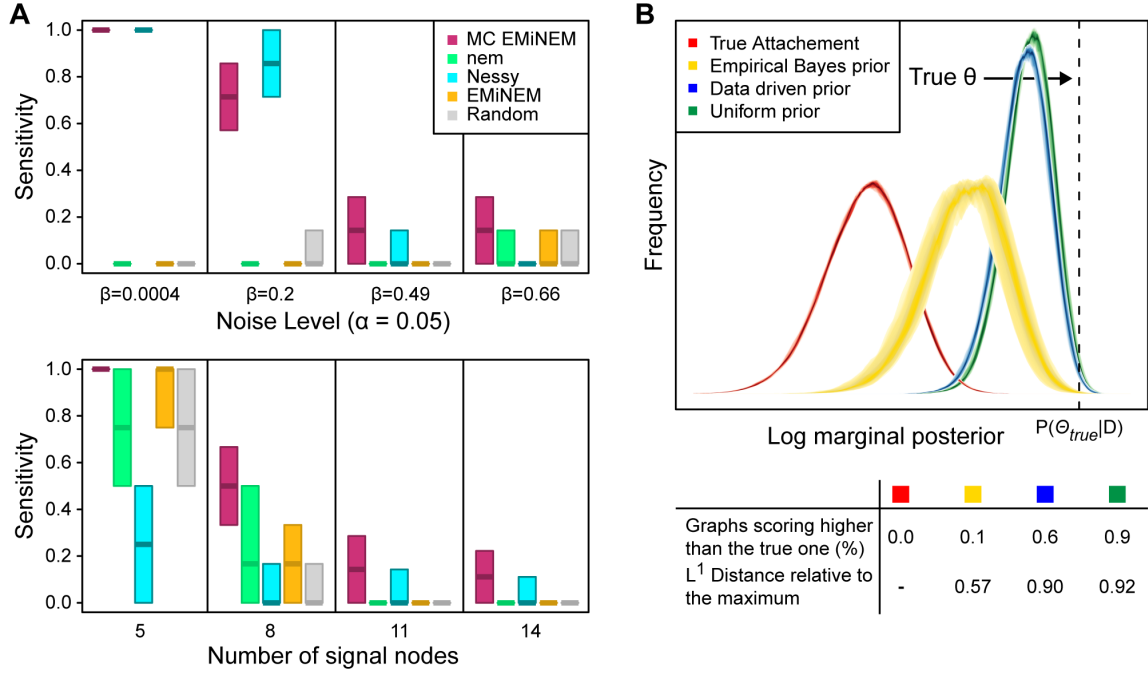
Figure 8.5.: (A) Prediction quality. Comparison of the sensitivity of MC EMiNEM and four alternative methods for four different noise levels (top) and four different signals graph sizes (bottom). The sensitivity is depicted on the y-axis, each frame corresponds to one parameter setting. Top: For a signals graph of 11 nodes, noisy datasets were generated such that for an optimal test with a type-I error (α-level) of 5%, a type II error (β − level) of $0.04\%, 20\%, 49\%$, and 66% would be achieved, respectively. Bottom: For a noise level corresponding to an error level of ($\alpha = 5\%$, $\beta = 49\%$), signals graph sizes of $|\mathcal{S}| = 5, 8, 11, 14$ are investigated. We expect our application to range within the four central scenarios. **(B)** Influence of the Empirical Bayes procedure. Here, for the standard setting $|\mathcal{S}| = 11$ and ($\alpha = 5\%$, $\beta = 49\%$). The x-axis shows the calculated marginal posterior values $P(\Theta|D)$ centered at $P(\Theta_{\text{true}}|D)$ (indicated by the dashed vertical line), on the y-axis the frequency is displayed. In the table, the percentages of signals graphs scoring higher than Θ_{true} are provided, as well as the L^1-distances (relative to the maximum).

8.2. Application: The signaling network of the yeast Mediator

8.2.1. Assessment of the MCMC sampling behavior

As for the simulation, the MCMC sampling behavior has also been assessed for the Mediator data. Convergence traceplots are shown in Fig. 8.6 (all edges) and Fig. 8.7 (selected edges). The development of the attachment of effects to signal nodes during the Empirical Bayes procedure is visualized in Fig. 8.8. Basically, the Markov chains for the Mediator data show the same behavior as for the simulated data (see Section 8.1.1). Due to more distinct entries in the ratio matrix, however, the variance within the chains is reduced and the preferred attachment of effects is clearer.
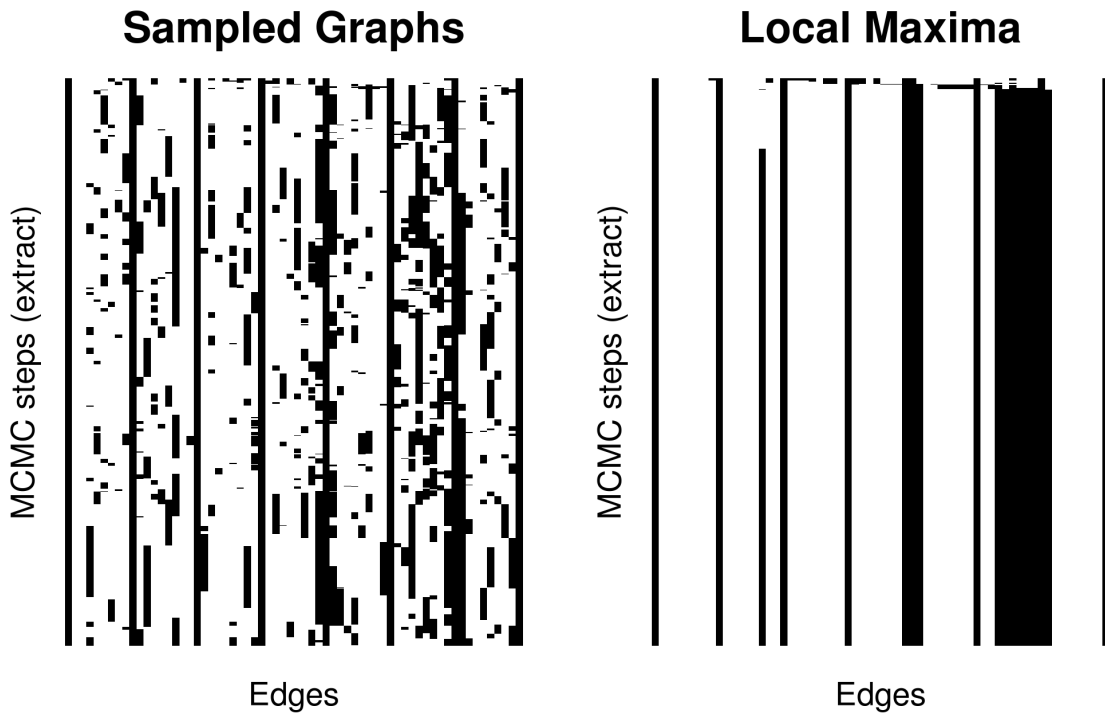
Figure 8.6.: Traceplot of all edges, Mediator data. Only the first 5000 MCMC steps are shown, since the chain converges very fast to one final signals graph (see also Section 7.2.2). The left panel shows the traceplot for the sampled graphs $(\Theta_i)_{i=1,2,...}$, the right panel shows the traceplot for the corresponding local maxima $(\hat{\Theta}_i)_{i=1,2,...}$. The MCMC steps are depicted on the y-axis (from top to bottom), individual edges on the x-axis, thus, one line in the traceplot corresponds to the signals graph of the corresponding MCMC step. Black fields indicate the presence, white fields the absence of a given edge in a given MCMC step. Completely black columns represent self-loops, which are defined to be present in the mathematical formulation and included here for reasons of clarity.. The sampled graphs comprise more edges and vary stronger, as compared to the sequence of local maxima. A discussion of this behavior is provided in the main text (Section 8.2.1).

Local Maxima

Sampled Graphs

Figure 8.7.: Traceplot of selected edges, Mediator data. The upper panel shows the traceplots of selected edges in the sequence of local maxima $(\hat{\Theta}_i)_{i=1,2,\ldots}$, the lower panel shows the traceplot of these edges in the sequence of the underlying sampled signals graphs $(\Theta_i)_{i=1,2,\ldots}$. On the x-axis, extracts of the MCMC steps at the beginning $(1-1000)$ and the end $(59000-60000)$ of the chain are depicted. Selected edges (edges that appear in > 40 MCMC steps in the sequence of local maxima) are drawn in different colors. Stacked on the y-axis are values of 0 and 1 for each edge, corresponding to the absence and presence of the edge at a given MCMC-step. As in Fig. 8.6, the sampled graphs comprise more edges and vary stronger, as compared to the sequence of local maxima. A discussion of this behavior is provided in the main text (Section 8.2.1).
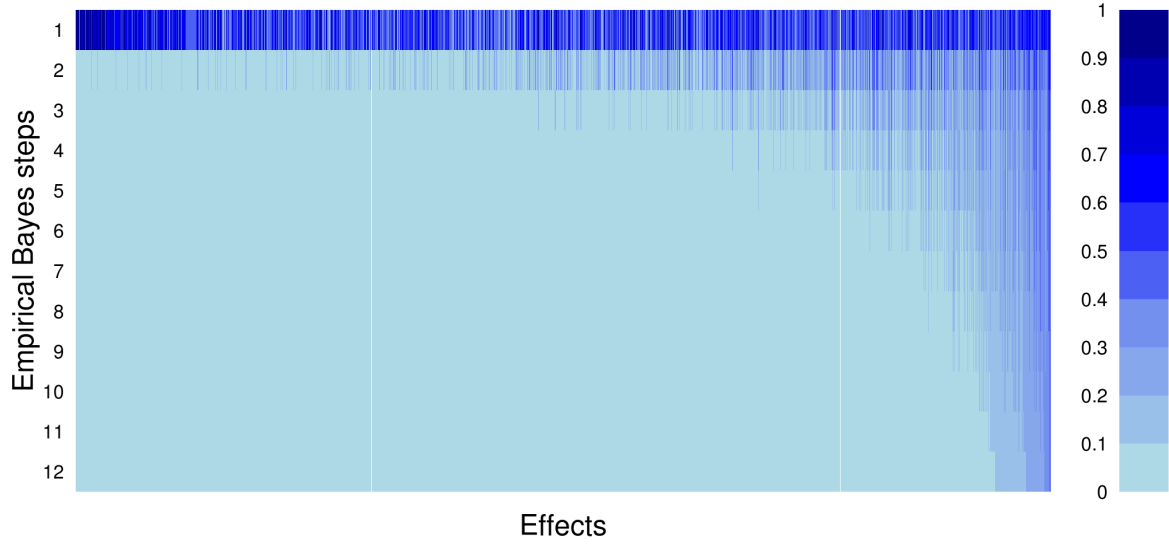
Figure 8.8.: Development of attachment entropy, Mediator data. For each effect j in each Empirical Bayes step l, the Shannon Entropy is calculated as $-\sum_{j\in\mathcal{S}} H_{jk}^l \cdot \log_2 H_{jk}^l$. On the y-axis, the Empirical Bayes steps are depicted (from top to bottom), on the x-axis, the effects are listed. The colors indicate the entropy, relative to the maximal one (when, for a given effect, the attachment probability is the same for any signal node (or no signal node at all)). Obviously, compared to the simulation results, the overall entropy is already much lower in the initial effects graph prior. Furthermore, most effects showing a high entropy in the first step converge to a preferred attachment very fast, only few edges show no preferences.

8.2.2. Results

MC EMiNEM predicts a robust Mediator subunit network

The perturbation of a central Mediator subunit can have severe consequences on the structure of the whole Mediator complex. It may cause the loss of whole modules or specific submodules [59, 99, 111]. The perturbation of a peripheral component might have only local effects on the Mediator structure and, consequently, have fewer effects on transcription. From the structural organization of the Mediator, we therefore expect a hierarchy of transcriptional effects upon subunit perturbations, which makes NEMs a suitable tool for their analysis. As a result of a NEM analysis, we expect the central Mediator subunits that have widespread effects upstream in the signals graph, whereas the more peripheral components should lie downstream. Due to its role as a general transcription factor involved in the formation of the transcription initiation complex, a perturbation of the Mediator can entail global changes in gene expression [39]. Such effects are completely removed by our normalization procedure and can therefore not be detected. Our focus in the present study, however, is on effects that are due to the interaction of the Mediator with gene-specific transcription factors. These effects are restricted to the target genes of the interacting transcription factors. They superimpose to the possible global effects of a Mediator perturbation, and hence become visible only after removal of the global effects.

A straightforward application of the MC EMiNEM algorithm led to identical results in nine out of ten independent MCMC runs; the tenth run differed only by one edge (see Appendix, Fig. A.3 and Fig. A.4). The runs revealed a bi-directional edge assigned to the Med10 and Med21 nodes, which means that these two subunits are indistinguishable in terms of their intervention effects. Their attached effect genes are interchangeable without affecting the model's likelihood. Therefore, according to [102], we combine the two subunits and treat
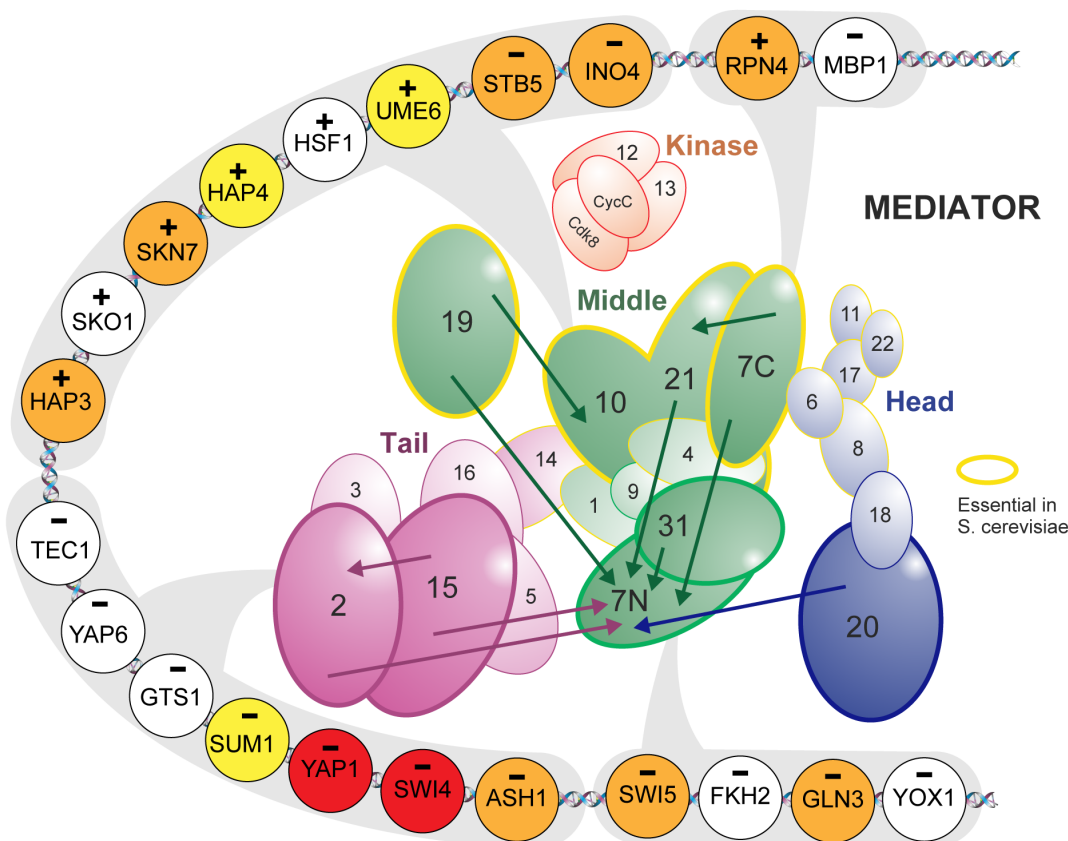
Figure 8.9.: Mediator network inferred by MC EMiNEM, with associated transcription factors (the basic Mediator cartoon was modified from [54]). The numbers of the Mediator subunits correspond to the unified Mediator nomenclature [12] and subunits that are part of this study are enlarged and have saturated colors. The two subunits Med10 and Med21 were merged as explained in the main text. The N-terminus and the C-terminus of Med7, which are represented by two individual perturbations in this study, are shown separately. Physically, they are connected by a flexible linker [55]. The arrows between the Mediator subunits show the signals graph of our MC EMiNEM analysis, arrow colors correspond to the module they originate from. The TFs surrounding the Mediator are the outcome of a gene set enrichment analysis of the MC EMiNEM effects graph. TFs are grouped into gray areas which link them to the Mediator subunit for whose target genes they are enriched. For each TF, minus, respectively plus signs indicate whether their targets are down-, respectively upregulated upon perturbation of the corresponding Mediator subunit. The results of the gene set enrichment analysis were compared to known interactions between TFs and Mediator subunits in BioGRID [95, 96]). Red: the interaction with the corresponding Mediator subunit is known; orange: an interaction with a Mediator subunit in the same module is known; dark yellow: confirmed interaction with the Mediator; white: no known interaction.

them as one node (see Appendix Section A.4.1). When Med10 and Med21 were combined, ten independent MC EMiNEM runs gave identical signals graph predictions (Fig. 8.9 and Appendix, Fig. A.5). The corresponding attachment of effects to signal nodes is provided in the Supplementary file 3 of [77], which is provided in digital form along with this thesis.

MC EMiNEM confirms the Mediator architecture

The predicted Mediator network (the signals graph in Fig. 8.9) agrees well with current knowledge about the Mediator structure [45, 55]: When removing the downstream Med7N node, the signals graph is separated into three connected components that reflect the modular organization of the Mediator (middle module: Med7C, Med19, Med10Med21, Med31; head module: Med20; tail module: Med2, Med15). While the overall module organization of the Mediator can also be recovered from a simple clustering analysis (see Fig. A.6), MC EMiNEM reveals a much finer structure by assigning a directionality to each edge. Med7N is downstream of all other nodes, indicating that among all perturbations that were applied, it has the least effects on transcription. It shows that there is a set of effects genes (attached to Med7N in the NEM) whose transcription depends on an entirely intact Mediator complex. The middle module component consists of a Med7C, Med10Med21 and Med19 upstream part, and a Med31, Med7N downstream part. Again, this conforms to its physical architecture: Med7C/Med10Med21 and Med7N/Med31 form stable complexes [55]. We conclude that the former are central architectural components, whereas the latter are peripheral. Indeed, Med7N/Med31 are only weakly attached to the middle module, and easily dissociate from it,
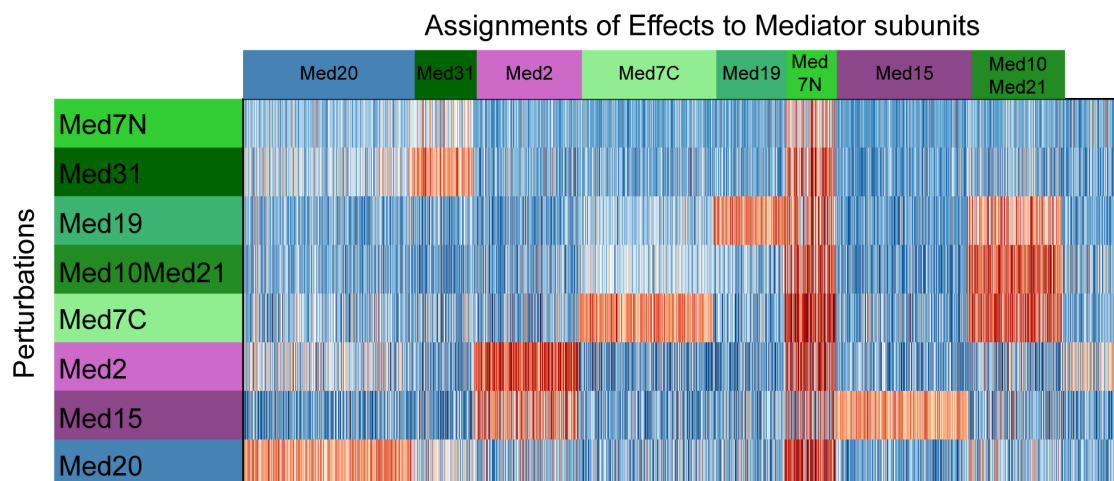
Figure 8.10.: Effects graph inferred from the Mediator data. Shown are the log-odds ratios which serve as MC EMiNEM's input. Genes that are likely to change in a given condition are depicted in red,and they are blue otherwise. Color saturation indicates the absolute value of the log-odds ratio (cf. Appendix, Fig. A.5). Rows correspond to Mediator perturbation experiments, columns correspond to genes, sorted according to their attachment to Mediator subunits. Mediator subunits are colored as in Fig. 8.9 and Fig. 8.11.

whereas Med7C/Med10Med21 are essential for its architecture [55]. The position of Med19 yet is still unclear [4, 98]. In our model, however, Med19 is clearly placed in the center of the middle module. The tail module interacts with gene-specific transcription factors and is structurally less analyzed [58]. The NEM includes an edge from Med15 to Med2 and thus suggests a more central role for Med15 than for Med2, because the effects upon perturbation of Med2 are a subset of the respective Med15 effects (see Fig. 8.10 and Fig. A.5).
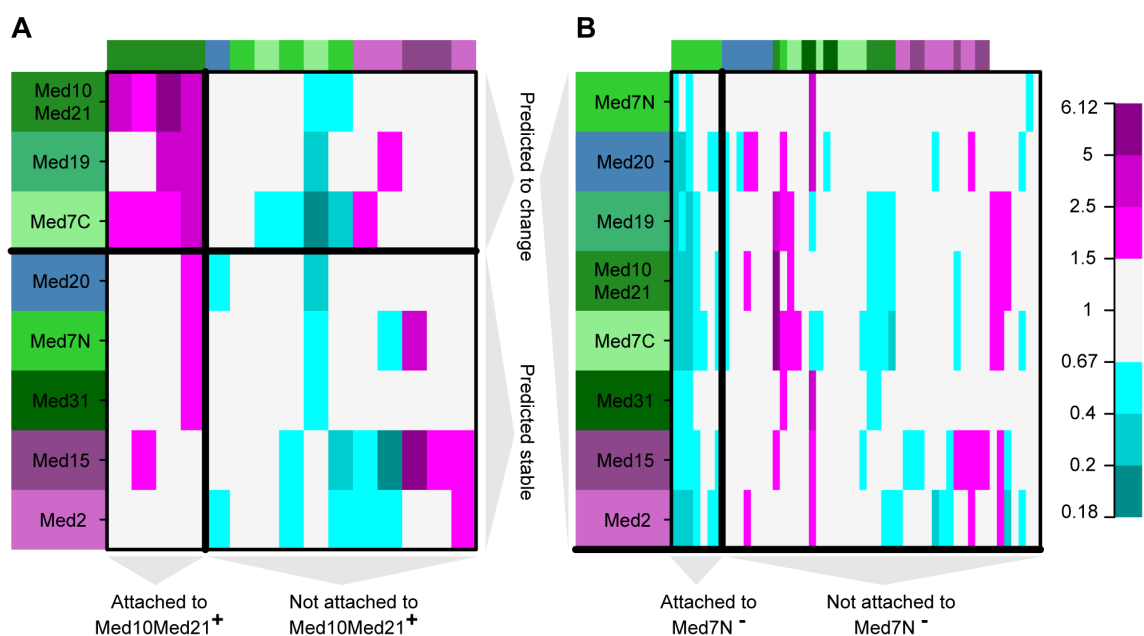
Figure 8.11.: Gene set enrichment analysis. **(A)** Expression changes of the target genes of SKO1 across all experiments. Experiments correspond to rows; the respective Mediator subunit perturbations are indicated by the colored boxes to the left of the heat map (coloring is in accordance with the Mediator module structure in Fig. 8.9). Target genes correspond to columns. If a target gene is attached to a Mediator subunit in the MC EMiNEM effects graph, this is indicated by a colored box on top of the respective column, using the same color code as for the experiments. Expression changes relative to wild type are color coded by the panel on the right. In the gene set enrichment analysis, SKO1 target genes were found enriched for upregulated genes attached to the Med10Med21node in the MC EMiNEM effects graph. These genes lie to the left of the bold vertical line in the heat map. Briefly, our Mediator NEM model predicts that they should also change their expression in the Med19 and Med7C perturbations, which lie above the bold horizontal line. Ideally, the expression changes in the upper left corner defined by the two bold lines should be strong and consistent, while those in the remaining part should be weaker and less consistent. **(B)** Same plot as (A), for the target genes of SWI5. Since SWI5 targets are enriched for downregulated genes attached to Med7N, and Med7N is downstream of all other nodes in the signals graph, we expect consistent expression changes of the Med7N attached genes across all perturbations.

MC EMiNEM provides a map of specific transcription factor - Mediator interactions

Apart from an estimate of the internal flow of regulatory information in the signals graph, MC EMiNEM returns a posterior probability of the attachment of effect genes to specific Mediator subunits (Fig. 8.10). The attachment of effects genes to signal nodes in the NEM framework does not necessarily represent a physical/direct interaction of the Mediator with the DNA. In the case of the Mediator it is sensible to assume that the coupling is mediated by transcription factors (TFs). We extend the analysis of our Mediator network and infer the transcription factors by which this coupling has been achieved (cf. [105]). We group the effect genes according to their attachment to signal nodes and according to the direction of expression change upon perturbation. A gene set enrichment analysis for these 16 groups then reveals interactions of gene-specific TFs with specific Mediator subunits / submodules. We used the MGSA algorithm for the enrichment analysis [6], based on the gene-TF assignment by Fraenkel [64] (see also Appendix Section A.4.3). Although the attachment of individual effects to Mediator subunits is notoriously variable (see Fig. 8.3 and 8.8), the gene set enrichment approach lends its robustness from combining evidence from many attached genes. The result is a map of TF-Mediator interactions, summarized in Fig. 8.9 and listed in Appendix Table A.1.

The 21 TFs-Mediator subunit interactions mapped by MC EMiNEM were validated using the BioGRID database [95]. Two interaction pairs were known from the literature (YAP1-Med2, SWI4-Med2). Another eight TFs were known to interact with a Mediator subunit from the same module as the predicted interacting subunit ([GLN3/SWI5]-Med7N, RPN4-Med7C, [SKN7/STB5/INO4/HAP3]-Med10Med21, ASH1-Med2). An interaction with the Mediator has been described for three more TFs ([UME6/HAP4]-Med10Med21, SUM1-Med2), and eight predicted interactions were new (MBP1-Med7C, [HSF1/SKO1]-Med10Med21, [TEC1/-YAP6/GTS1]-Med2, [FKH2/YOX1]-Med7N).

All target genes of TFs associated with the tail module show downregulation after perturbation, consistent with the tail's function to contact gene specific transcription factors [17]. The same holds for the target genes of TFs associated with Med7N. This is expected, as the genes attached to Med7N are those that show an effect in all perturbations (Fig. 8.10) and therefore presumably require a completely intact Mediator. The target genes of TFs associated to the rest of the middle module show expression changes in both directions, in accordance with the middle module described as an ambiguous regulator [104].

Fig. 8.11A offers a TF-centric excerpt on the MC EMiNEM map from Fig. 8.10. It drills in to the target genes of SKO1, which are enriched in the set of upregulated genes attached to Med10Med21. SKO1 is both a transcriptional activator and repressor and forms a complex with the general repressor TUP1 (Saccharomyces Genome Database [16]). TUP1 in turn targets Med21p [34]. A Mediator complex lacking this subunit might thus not be able to forward repressive signals, resulting in upregulated target genes of SKO1.

The transcriptional activator SWI5 has a large number of physical interactions with Mediator subunits from various modules (Med15, Med17, Med18, Med22, [16]). This suggests that any change in the Mediator structure affects its interaction with SWI5. Consequently, target genes of SWI5 should change their expression upon any Mediator subunit perturbation. Fig. 8.11B confirms this behavior of the SWI5 targets: MC EMiNEM associates SWI5 to Med7N, because SWI5 targets are enriched in the set of downregulated genes attached to Med7N, and these are consistently downregulated in all perturbations.

Similar analyses were carried out for all TFs in the MC EMiNEM map (see Supplementary file 2 of [77]; lists of genes that contribute to the respective TF enrichments are provided in the Supplementary file 4 of [77]; both files are provided in digital form along with this thesis). The most striking observation is that the sign of a gene's expression change is consistent in virtually all perturbations for which MC EMiNEM predicts an effect. Since our model is completely blind with respect to the sign of regulation, the consistency in the direction of the expression changes provides compelling evidence that the signals graph reflects regulatory dependencies between Mediator subunits which are likely to be caused by structural changes.

8.2.3. Summary & Outlook

The reconstruction of interaction networks from high dimensional perturbation effects is still a challenge. We have developed MC EMiNEM, a method for the learning of a Nested Effects Model. We introduced two major improvements, namely an Expectation-Maximization algorithm for the very fast detection of local maxima of the posterior probability function. Mode hopping Markov Chain Monte Carlo sampling was then used for the efficient exploration of the space of local maxima. We applied MC EMiNEM to a combination of proper and public gene expression data obtained from Mediator subunit perturbations. It turned out that MC EMiNEM does not only shed light on structural dependencies of Mediator subunits, it also identifies interactions of gene-specific transcription factors with Mediator subunits. Our findings are consistent with the state-of-the-art knowledge about the Mediator architecture and function. Hierarchical clustering has proved tremendously useful for the analysis of expression data obtained from observational experiments. We have established MC EMINEM for the analysis of expression data from intervention experiments, as the appropriate counterpart to clustering.

In the future, it would be interesting to revise the construction of the data-driven prior and the selection of effect genes. As for the data-driven prior, the putative probability that an effect is attached to none of the signals (i.e., that it's attached to the null node) is currently set to its average probability of being attached to any of them. Two additional constructions seem promising, however: First, setting this probability to the average of the attachment probabilities of all effects to all signals (i.e., to the same value for all effects) would push genes that show weak effects to the null node, while genes showing strong effects would be pulled away from it. Second, it would be intuitive to set the odds-ratio for the null-node attachment to 1 and to calculate the probability accordingly. By definition, a gene attached to the null node shows the same reactions upon any perturbation and thus the probability for the data is the same, independent of the underlying model.

A restriction of effects taken into account for the estimation process (for example only effects k with $\max_{j \in \mathcal{S}}(R_{jk}) > r$ for a predefined threshold r are considered) could help to reduce noise and to make the prediction more reliable (based on a more significant set of effects). However, it could also lead to a loss of information with respect to the gene set enrichment analysis.

As can be seen from this discussion, the best way to design the estimation process is not always obvious, if there is a "best way" at all. A careful assessment of the possibilities that are worth being considered and a reasoned decision are thus crucial.

Part III.

Quantitative analysis of processive RNA degradation by the archaeal exosome

9. Introduction

RNA exosomes are large multi-subunit protein complexes involved in controlled and processive 3' to 5' RNA degradation. They form large molecular chambers and harbor multiple nuclease sites as well as RNA binding regions. This makes a quantitative kinetic analysis of RNA degradation with reliable parameter and error estimates challenging. Consequently, RNA degradation by RNA exosomes has scarcely been studied and little is known about the features that contribute to efficient RNA degradation.

We propose the combination of a differential equation model with Bayesian Markov Chain Monte Carlo (MCMC) sampling for a more robust and reliable analysis of such complex kinetic systems. Using the exosome as a paradigm, it is shown that conventional "best fit" approaches to parameter estimation are outperformed by the MCMC method. The parameter distribution returned by MCMC sampling allows a reliable and meaningful comparison of the data from different time series. In the case of the exosome, we find that the cap structures of the exosome have a direct effect on the recruitment and degradation of RNA and that these effects are RNA length-dependent. The described approach can be widely applied to any processive reaction with a similar kinetics, like the XRN1-dependent RNA degradation, RNA/DNA synthesis by polymerases and protein synthesis by the ribosome.

The content of this part has been published in 2010 (Hartung, Niederberger *et al.* [36], focus on biology) and in 2011 (Niederberger, Hartung *et al.* [78], focus on model selection), and the work has been done in close collaboration with Karl-Peter Hopfner and Sophia Hartung. My contribution to this work is the development of a reliable method for the analysis of processive reactions that are described by ordinary differential equations (ODEs)., i.e., the development of the Markov Chain Monte Carlo sampling approach (see Chapter 12). In addition, the application of the method to RNA degradation by the archaeal exosome required a reparametrization of the model (i.e., the introduction of the catalytic efficiency, see Section 11.3) and an appropriate choice of the prior distribution (see Section 11.4.3). The data has been generated and the basic model (the system of ODEs) has been developed by Sophia Hartung and Karl-Peter Hopfner. Together with a maximum likelihood parameter fit, the data and basic model have already been published in the PhD thesis of Sophia Hartung [35]. A discussion of alternative model classes that were not able to explain the data sufficiently has been published there, they are not within the scope of this thesis. The interpretation of the biological results has been done by and under the lead of Karl-Peter Hopfner.

10. Biological background

In all three domains of life, RNA degradation, i.e., nucleolytic cleavage, is associated with essential cellular processes such as RNA maturation, RNA quality control and RNA turnover. During RNA maturation, the precursor to any type of RNA is transformed into is functional, mature form, e.g., by 3' polyadenylation, 5' capping or splicing. RNA quality control, e.g., the non-sense-mediated or the non-stop decay pathways, ensures that dysfunctional RNA molecules are degraded quickly. RNA turnover maintains a balance between RNA synthesis and decay. Deviations in any of these processes may have serious impacts on the functionality of a cell [46].

10.1. The archaeal exosome

The exosome is a 3'-5' exoribonuclease involved in RNA maturation, RNA quality control and RNA turnover in eukaryotes and archaea. It has first been identified in yeast in 1997 [73] and later on in archaea [23, 52]. It is closely related to the bacterial PNPase, which is also present in mitochondria and chloroplasts. The similarity of the core complex between all three domains of life indicates the existence of a common ancestor, and a strong evolutionary conservation [40, 86].

Structure

The central chamber of the 250 kDa archaeal exosome, a hexameric ring with a length of 50-60Å (corresponding to $\sim 7-9$ nucleotides), is formed by a trimer of heterodimers (($Rrp41{:}Rrp42)_3$). Rrp41 and Rrp42 have been detected in connection with rRNAs (rRNA processing proteins) first and possess RNase PH like domains enabling phosphorolytic ribonuclease activity. Even though only the catalytic center of Rrp41 is still active, an intact dimer of Rrp41 and Rrp42 is necessary for the correct positioning and binding of the RNA molecule. The existence of three active sites in one exosome molecule ensures a high processivity during RNA degradation.
The central chamber might be entered by two sides. On the one opposite the active sites, a trimer of either Csl4 or Rrp4 forms a flat, multidomain cap. Each Rrp4-molecule possesses one S1 and one KH protein binding domain, while each Csl4-molecule possesses one S1 and one zinc-ribbon RNA binding domain. The three S1 domains, very similar in structure and location in both Rrp4 and Csl4, frame a central pore, denoted S1 pore (18Å with the Csl4 trimer, 15Å with the Rrp4 trimer). The RNA binding domains, their positive surface potential and a negative charge distribution at the opposing opening of the hexameric ring indicate

that the Csl4 and Rrp4 caps recruit the RNA molecules and regulate their access to the active center via the S1 pore. The nucleotides resulting from the cleavage are supposed to exit by the second, larger pore (denoted PH pore). The structure and location of the KH and zinc-ribbon RNA binding domains differ strongly, which is assumed to enable the recognition of different RNA types or cofactors by the Csl4 and the Rrp4 cap, respectively. Their peripheral position in the cap as compared to the central position of the S1 binding domains support this assumption. Even though only Rrp4 or Csl4 homotrimers have been crystallized so far, heterotrimers are structurally possible *in silico* and might form i*n vivo.*

A four nucleotide long RNA binding pocket in Rrp41 guides the RNA to the active site, whose entrance is oriented contrary to the S1 pore. The only 8-10Å wide neck restricts the access to the wider central chamber to molecules without secondary structure and enables the advancement of one base at a time. Thus, additional cofactors such as RNA helicases are needed to unwind structured or double-stranded RNA molecules.

The architecture of the eukaryotic exosome is more complex but nevertheless very similar to the archaeal one. It contains a larger number of subunits whose assembly is both species- and compartment-dependent.

This paragraph is based on [14, 40, 86].

Function

The exosome is involved in total RNA degradation as well as in RNA maturation, where only a limited number of nucleotides is cleaved. It is still not clear how the distinction between the two situations is possible, though it is assumed that additional cofactors and signals play an important role. The fact, that only single-stranded, unfolded RNA substrates may enter the central chamber suggests a model where regions of secondary structure or protein:RNA complexes prevent the RNA substrate from further degradation and make an accurate trimming of the 3' end possible. In addition, the selective unwinding of structured RNA, for example by the eukaryotic TRAMP or SKI complexes, allows the complete degradation of the RNA substrate. Highly unstructured poly(rA)-tails may facilitate the initial threading of RNA into the central chamber.

The RNase PH like domains enable phosphorolytic ribonuclease activity, which means that the exosome cleaves the RNA using inorganic phosphate and releasing mononucleotide 5'-diphosphate products. In an environment with a high nucleotide diphosphate concentration the inverse reaction (i.e., polymerization) is possible.

This paragraph is based on [40].

10.2. Experimental setup

The following analysis is based on 7 variants respectively mutants of the *Archaeoglobus fulgidus* exosome, which are depicted in Fig. 10.1. For detailed descriptions about their preparation and the introduction of the mutations, please refer to [14, 36]. Since the reactions take place in an excess of inorganic phosphate (10mM phosphate compared to only 3.6 mM ADP at the time all RNA molecules are totally degraded), we may assume that no polymerization takes place. The RNase assays have been performed with 5'-radioactively labeled poly(rA)-oligoribonucleotides. The data consists of time series measurements of the amount of RNA of different lengths (from 30 base pairs to 3 base pairs), which were resolved on a denaturing polyacrylamide gel and quantified by phosphorimaging. The respective time points vary among the exosome variants. The initial amount of RNA at time point 0 is 120nM in all cases, the initial amount of the exosome at time point 0 is 30nM (Csl4 exosome, Rrp4 exosome, interface mutant), 60nM (capless exosome, Csl4 R65E mutant, Csl4 Y70A mutant) or 120nM (crosslink mutant). The measurements for the Rrp4 cap and Csl4 cap exosomes are illustrated in Fig. 10.2.
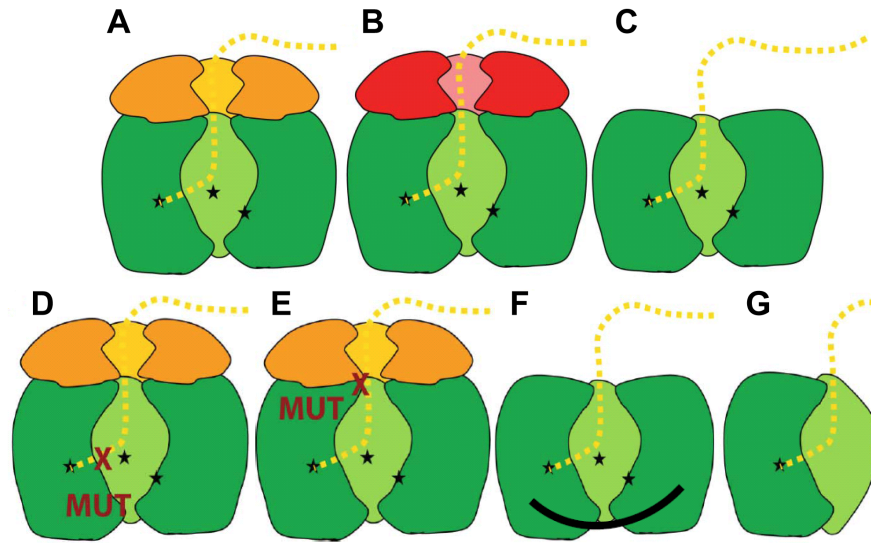
Figure 10.1.: This figure provides a schematic representation of the variants respectively mutants used in this study: (A) The exosome with a Csl4-cap (wild type) (B) the exosome with a Rrp4-cap (wild type) (C) the capless exosome (wild type) (D) the Csl4-exosome with a Y70A mutation (a tyrosine replaced by an alanine close to the active site) in Rrp42 (E) the Csl4-exosome with a R65E mutation (an arginine replaced by a glutamic acid in the neck) in Rrp41 (F) a rigidified crosslink mutant (G) an interface mutant which can't form the ring structure any more, resulting in three stable Rrp41:Rrp42 heterodimers. This figure has been modified from [35].
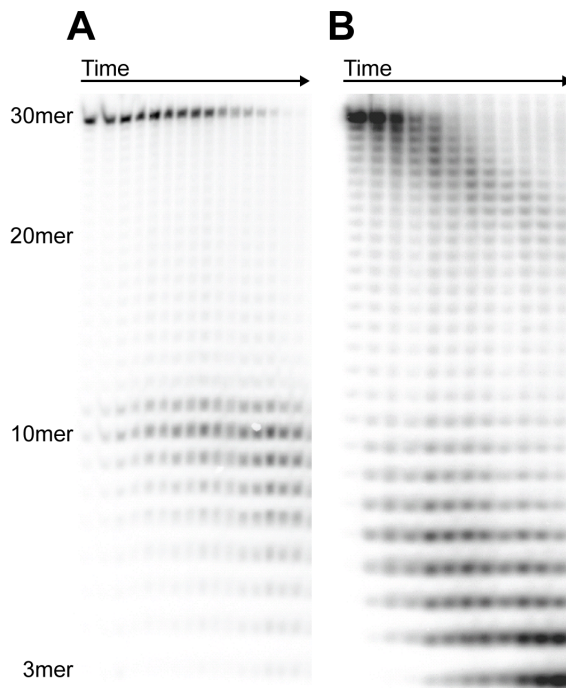
Figure 10.2: The denaturing polyacrylamide gel for the Csl4-exosome (A) and the Rrp4-exosome (B). At time point 0 only 30mer RNAs are present, at the final time point mainly short RNAs are present. The gel already shows length dependencies as well as differences between the degradation efficiencies of exosomes with different cap structures: While the Csl4-exosome obviously needs some time to get started, it is quit fast for long RNAs and than slows down significantly for short RNAs (an accumulation of RNAs is visible). The Rrp4-exosome on the other hand shows a quick start but slows down for long RNAs (again, an accumulation of RNAs is visible), accelerates for RNAs of medium length and slows down again for short RNAs. A discussion of these observations will be provided in Section 13.2.3. This figure has been modified from [35].

11. A model for processive RNA degradation

Our model for RNA degradation by the archaeal exosome involves four different reactions, namely the association of free RNA to the exosome, the dissociation of bound RNA from the exosome, the cleavage of bound RNA, and the inverse reaction, i.e., the polymerization of bound RNA. Please note that "bound" refers to the active center, not to the whole exosome complex. A more complex model class might include this distinction, however this would lead to clear overparametrization given the available data (see also Section 11.3).

11.1. Basic parametrization

The model we apply to RNA degradation is described by the following parameters: the association rates $k_{a,j}$, the dissociation rates $k_{d,j}$, the cleavage rates $k_{c,j}$, and the polymerization rates $k_{p,j}$, each for RNAs of lengths j, $j \in J = \{4, 5, ..., 30\}$, which results in a total of $4 \cdot 27 = 108$ parameters to be estimated. The 3mer RNA is considered as the final product and thus excluded from the set of parameters. Since the reaction takes place in an excess of inorganic phosphate (10mM phosphate compared to only 3.6 mM ADP at the time all RNA molecules are totally degraded) we may assume that no polymerization takes place, i.e., $k_{p,j} = 0$ for all j. Consistently, we saw no synthesis of longer RNAs in our reactions. This reduces the number of parameters to $3 \cdot 27 = 81$ ($\Theta = \{k_{a,j}, k_{d,j}, k_{c,j} | j \in J\}$). A scheme for the simplified kinetic model is presented in the following:

$$[\text{Exosome:RNA}_{j+1}] \underset{k_{a,j+1}}{\overset{k_{d,j+1}}{\rightleftarrows}} [\text{Exosome}] + [\text{RNA}_{j+1}]$$

$$\downarrow k_{c,j+1}$$

$$[\text{Exosome:RNA}_{j}\ ] \underset{k_{a,j}}{\overset{k_{d,j}}{\rightleftarrows}} [\text{Exosome}] + [\text{RNA}_{j}]$$

$$\downarrow k_{c,j}$$

$$[\text{Exosome:RNA}_{j-1}] \underset{k_{a,j-1}}{\overset{k_{d,j-1}}{\rightleftarrows}} [\text{Exosome}] + [\text{RNA}_{j-1}]$$
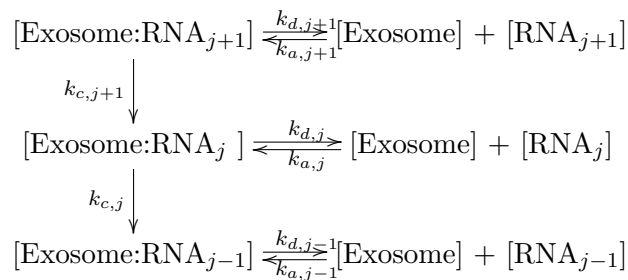
Based on this scheme, the changes in the amounts of RNA of length j, $j = 3, 4, ..., 30$ with regard to time can be modeled by ordinary differential equations (ODEs).

11.2. Ordinary Differential Equations (ODEs)

We denote by $r_j = r_j(t)$ the total amount of RNA of length j, $j \in J^\star = \{3\} \cup J$. Let the corresponding amount of exosome-bound RNA be $x_j = x_j(t)$, the unbound fraction be $y_j = y_j(t)$. The (initial) amount of free exosome is denoted by $e = e(t)$. The system of ordinary differential equations (ODEs) that describes the dynamics in the kinetic system is then parametrized by the set $\Theta = \{k_{a,j}, k_{d,j}, k_{c,j} | j \in J\}$:

$$
\begin{aligned}
r_j(t) &= x_j(t) + y_j(t) \\
\frac{dx_j}{dt}(t) &= k_{c,j+1}x_{j+1}(t) + k_{a,j}y_j(t)e(t) - (k_{c,j} + k_{d,j})x_j(t) \\
\frac{dy_j}{dt}(t) &= k_{d,j}x_j(t) - k_{a,j}y_j(t)e(t)
\end{aligned}
\tag{11.1}
$$

The initial conditions are $r_j(0) = x_j(0) = R_j$, $y_j(0) = 0$, $e(0) = E$, where the initial total amount of RNA, R_j, and the initial amount of free exosome, E, are given. The signs in Eq. 11.1 are chosen such that all parameter values are positive, thus our parameter space is $\Omega = \mathbb{R}_{>0}^J \times \mathbb{R}_{>0}^J \times \mathbb{R}_{>0}^J$. If we want to emphasize the dependence of the results on the parameters Θ, we denote those in the superscript, e.g., we write $r_j^\Theta(t)$ instead of $r_j(t)$ etc. The experimental data consists of a matrix $D = (R_{j,k})$, where $j \in J^\star$ runs through the lengths of the measured RNA populations, and $k \in \{1, ..., K\}$ enumerates the measurements that were taken at times $t \in \{t_1, ..., t_K\}$ respectively. A standard ODE solver (MATLAB® [69], ode15s, default parameters) is used to calculate the predictions $r_j^\Theta(t)$ of a model given by Θ. The comparison of the measurements $R_{j,k}$ with the predictions $r_j^\Theta(t_k)$ is the basis of our MCMC sampling strategy.

11.3. Parametrization revised

It soon turned out that the parameters of the model were partly redundant, hence, that the model was clearly overparameterized. A reparametrization together with a reduction in the number of free parameters could solve the problem.

Overparametrization

After excluding polymerization due to the experimental conditions we still were left with 81 free parameters to be estimated reliably. This is unlikely to be feasible, based on an amount of ~ 1000 individual measurements. In addition, the amount of each RNA population is only measured in total whereas the amount of free or bound RNA is not known. It can therefore not be detected whether either association or cleavage are the bottleneck for the decay process. More precisely, it is not discernible whether an average decay rate is caused by fast association and slow cleavage (most of the RNA is bound to the exosome), slow association and fast cleavage (most of the RNA is free) or by average association and average cleavage (free and

bound RNA are well-balanced). Furthermore, a fast association / cleavage can be compensated by a fast dissociation, leading to a similar overall decay as a slow association / cleavage and a slow dissociation. A reparametrization in combination with a parameter reduction can cure these problems. In search of an identifiable quantity that describes the efficiency of decay appropriately we tested the straightforward guess $\frac{k_a \cdot k_c}{k_d}$, and the *catalytic efficiency* $\frac{k_a \cdot k_c}{k_d + k_c}$.

Catalytic efficiency

The quantity *catalytic efficiency* $e = \frac{k_a \cdot k_c}{k_d + k_c}$ is based on Michaelis-Menten type kinetics and a measure of the velocity of the RNA intermediate's degradation by the exosome. It has been shown to be meaningful when comparing the reaction rates for multiple substrates competing for one enzyme [57], where also the name specificity constant was coined. For a more detailed derivation of the constant see Appendix Section B.1.

Parameter reduction

To determine the catalytic efficiency for each RNA length, not every parameter has to be estimated individually. Taking into account the relations between association, cleavage and dissociation we opt for the following procedure:

- **Dissociation** is fixed to one value for all RNA lengths (one manually fixed parameter). This keeps the range of possible association / cleavage parameters limited. The empirically determined value is high enough to avoid a limiting influence ($k_d = 10$).

- **Cleavage** is sampled once for all RNA lengths (one free parameter). This offers more flexibility for the association parameters.

- **Association** is sampled individually for each RNA length (27 free parameters, restricted by our prior assumptions). It accounts for the length-dependent differences of RNA decay.

The choice of association as the flexible parameter is motivated by the biochemical background of RNA degradation by the exosome: The actual length of the remaining RNA molecule seems to be quit insignificant for the split-off of the first base, when it is already bound to the active center. However, for reaching this bound state, we surely expect some kind of length dependency. Furthermore, since the interest of this approach is to determine the catalytic efficiency of RNA decay, which is mainly driven by association and cleavage but antagonized by dissociation, we excluded the possibility of dissociation being the flexible parameter.

This parameter reduction leaves 28 parameters to be estimated ($\Theta = \{k_{a,j} | j \in J\} \cup k_c$), which still suffice for a good approximation and are identifiable. Both simulated data and observed data justify this approach, as illustrated in Section 13.1.1 and 13.2.1, respectively.

11.4. Probability model

11.4.1. The likelihood distribution

To assess the probability of a given parameter set Θ_n, we solve Eq. 11.1 and compare the predicted values $r_j^{\Theta_n}(t_k)$ with the observed values $R_{j,k}$. We assume that the observations $R_{j,k}$, $j \in J^\star$, $k = 1, ..., K$, are realizations of a Gaussian $\mathcal{N}(r_j^{\Theta}(t_k), \sigma_{j,k}^2)$-distributed random variable respectively, i.e.,

$$P(R_{j,k}|r_j^{\Theta}(t_k), \sigma) = \frac{1}{\sqrt{2\pi}\sigma_{j,k}} \exp\left(-\frac{1}{2} \frac{(R_{j,k} - r_j^{\Theta}(t_k))^2}{\sigma_{j,k}^2} \right) \ , \tag{11.2}$$

where the variances $\sigma = \{\sigma_{j,k}|j \in J^\star, k = 1, ..., K\}$ have to be given *a priori*, or must be determined from the data. This point is addressed in the following section. Assuming independence of the measurement errors, the likelihood of the model becomes

$$L(\Theta) = \prod_{j=3}^{30} \prod_{k=1}^{K} P(R_{j,k}|r_j^{\Theta}(t_k), \sigma) \tag{11.3}$$

11.4.2. The measurement error model

We assume a common error model: the measurement variance is composed of a constant "background" term β and a term which is proportional to the square of the intensity [21, 42],

$$(\sigma_{j,k})^2 = \alpha(r_j^{\Theta_{\text{true}}}(t_k))^2 + \beta \ , \quad \text{for some } \alpha, \beta \geq 0 \tag{11.4}$$

Since the measurement variances are not known here *a priori*, we developed an adaptive strategy for their estimation, which will be introduced in conjunction with the MCMC sampling scheme in Section 12.1. This adaptive strategy is initialized rather conservatively with values inferred from a good fit for the observed data ($\alpha = 0.9420$, $\beta = -0.0036$).

11.4.3. The prior distribution

To avoid overfitting and to reduce the effective number of parameters, we did not choose a uniform (pseudo-) prior on Θ. It is sensible to believe that consecutive catalytic efficiency values e_j, e_{j+1} tend to have similar values, since they reflect length (j) dependent properties of the RNA-exosome interaction. Therefore we introduced a smoothness prior by assuming that each of the consecutive differences $e_j - e_{j+1}$ (independently) follows a Gaussian distribution:

$$\pi(e) = \prod_{j=4}^{29} \frac{1}{\sqrt{2\pi}\lambda} \exp\left(-\frac{1}{2} \frac{(e_j - e_{j+1})^2}{\lambda^2} \right), \tag{11.5}$$

The hyperparameter λ was determined in simulation runs (see Section 13.1.2). The smoothness prior introduces a bias-variance tradeoff by reducing the effective number of model parameters with decreasing λ.

12. Parameter estimation by MCMC sampling

This chapter focuses on the application-specific implementation of Markov Chain Monte Carlo sampling. In particular, this includes the choice of the proposal function and the development of the adaptive likelihood MCMC sampling approach. A general introduction to Markov Chain Monte Carlo sampling is provided in Section 2.4.

12.1. Adaptive likelihood MCMC

The measurement variances are not known *a priori* in our situation. Instead of guessing these values beforehand, we introduce an adaptive strategy for their estimation. Starting with very high initial variances, i.e., a very flat likelihood function, we estimate the variances from the comparison of predicted and measured RNA values after every 100 steps in the Markov chain, and we update the variances by moving their current values into the direction of these estimates. More specifically, let $\hat{\sigma}^{\text{old}} = \{\hat{\sigma}^{\text{old}}_{j,k} | j \in J^\star, k = 1, ..., K\}$ be the current set of variance estimates. Let $r^{\Theta_s}_j(t_k)$, $s = 1...100$, be the predictions that were produced during 100 steps in a Markov chain $(\Theta_s)_{s=1,...,100}$, using $\hat{\sigma}^{\text{old}}$ as variance parameters. Assuming that the parameters Θ_s are close to the true parameters, the measurement error for the RNA population of length j at time point t_k is roughly $R_{j,k} - r^{\Theta_s}_j(t_k)$. A sensible guess for the variance $(\hat{\sigma}^{\text{new}}_{j,k})^2$ is thus

$$(\hat{\sigma}^{\text{est}}_{j,k})^2 = \text{mean}((R_{j,k} - r^{\Theta_s}_j(t_k))^2 | s = 1, ..., 100) \tag{12.1}$$

We used $(\hat{\sigma}^{\text{est}}_{j,k})^2$ in place of $(\sigma_{j,k})^2$ to fit the parameters α^*, β^* of the error model in Eq. 11.4. Plugging α^*, β^*, and $r^{\Theta_s}_j(t_k)$ into Eq. 11.4 again produces smoothed estimates $(\hat{\sigma}^{\text{est}*}_{j,k})^2$ [82]. One could replace the old variances $(\hat{\sigma}^{\text{old}})^2$ by these estimates. Such a procedure has desirable properties: Since we are starting with permissive values for σ, the initial likelihood landscape is flat, which makes the chain fully explore the model space and reduces the danger of getting irreversibly caught in a local maximum. During the adaptation process, the variances tend to drop, which can be understood as a kind of annealing. Individual variances $\sigma_{j,k}$ may remain disproportionally large if the corresponding measurements are faulty/flawed, therefore providing an automatic outlier detection mechanism. On the other hand, a too fast diminishment of the variances may precipitately fix some kinetic parameters to wrong values. We guard against such effects by smoothly adjusting the old variances into the direction of the new ones:

$$(\hat{\sigma}^{\text{new}}_{j,k})^2 = \delta \cdot (\hat{\sigma}^{\text{est}*}_{j,k})^2 + (1 - \delta) \cdot (\hat{\sigma}^{\text{old}}_{j,k})^2 \tag{12.2}$$

$\delta \in [0, 1]$ has been determined empirically (data not shown) and set to $\delta = 0.05$. To restrict the range of possible variances, an empirical boundary has been set to $\sigma_{\min}^2 = 0.5$ and $\sigma_{\max}^2 = 150$.

12.2. Implementation of the sampling procedure

Initialization

The MCMC sampling has been initialized with both random and uniform parameters to assess its robustness with regard to initialization. For the final analysis it has been initialized with "good" parameters derived from preliminary runs on the observed data to produce a fast-converging sample.

Choice of the proposal function

Each individual parameter in $\Theta = \{k_{a,j}|j \in J\} \cup k_c$ is independently sampled on a log scale from a Gaussian distribution, centered at the previous parameter value,

$$
\begin{aligned}
k'_{aj} &\sim \mathcal{LN}(k_{a,j}^n, \tau_a^2) \ , \quad j \in J \\
k'_c &\sim \mathcal{LN}(k_c^n, \tau_c^2)
\end{aligned}
\tag{12.3}
$$

The width of the proposal distribution is given by the variances τ_x^2. Simulation runs have demonstrated (data not shown) that setting $\tau_a^2 = 0.05$ and $\tau_c^2 = 0.005$ yields a sufficient acceptance rate as well as a good mixing behavior of the chain. $k_{a,j}$, $j \in J$ and k_c are sampled alternating.

Chain length and burn-in phase

Each chain consists of $2 \cdot 10^5$ steps. The length of the burn-in phase varies among the datasets and has been set to a safe value of $1.5 \cdot 10^5$.

13. Results & Discussion

13.1. Simulation

Datasets have been simulated as explained in Section B.2.1. For each dataset $M = 5$ Markov chains, each with $S = 2 \cdot 10^5$ steps, have been created, producing the parameter values $\Theta_s^m = \{k_{a,j,s}^m | j \in J\} \cup k_{c,s}^m$, $s = 1, ..., S$, $m = 1, ..., M$.

13.1.1. Assessment of parameter dependencies

In this application, the simulation was especially useful to analyze parameter dependencies and to detect an identifiable parameter. As has been discussed before (Section 11.3), the association, cleavage and dissociation rates are highly redundant, which is illustrated in Fig. 13.1. This problem has been solved by the introduction of the catalytic efficiency as an identifiable parameter (see Fig. 13.2). These figures refer to the predicted parameters for the 30mer RNA, based on a simulated dataset with a noise level of $\alpha = 10\%$. Choosing the association rate rather than the cleavage rate as the flexible parameter (i.e., to be sampled RNA length-dependent) is motivated by biochemical reasons as well as by comparative simulation runs (see Fig. 13.3).

13.1.2. Choice of the prior strength

We performed several MCMC runs to determine the strength of the prior. The goal is to optimize the bias-variance trade-off by restricting the parameter fluctuation from RNA length j to length $j+1$. This is based on the assumption that the catalytic efficiency changes smoothly from step to step rather than making big jumps. The results are depicted in Fig. 13.4. In simulation runs the best trade-off was met at a hyperparameter choice $\lambda = 0.5$.

13.1.3. Assessment of the MCMC sampling behavior

Fig. 13.5 illustrates the convergence speed of a Markov chain based on a simulated dataset. We define by eye a value U after which the variation of the chain does not decrease further. The first steps $s = 1, ..., U$ (the burn-in phase) are excluded from further analysis and only the steps $s = U + 1, ..., 2 \cdot 10^5$ (the stationary phase) are taken into account.
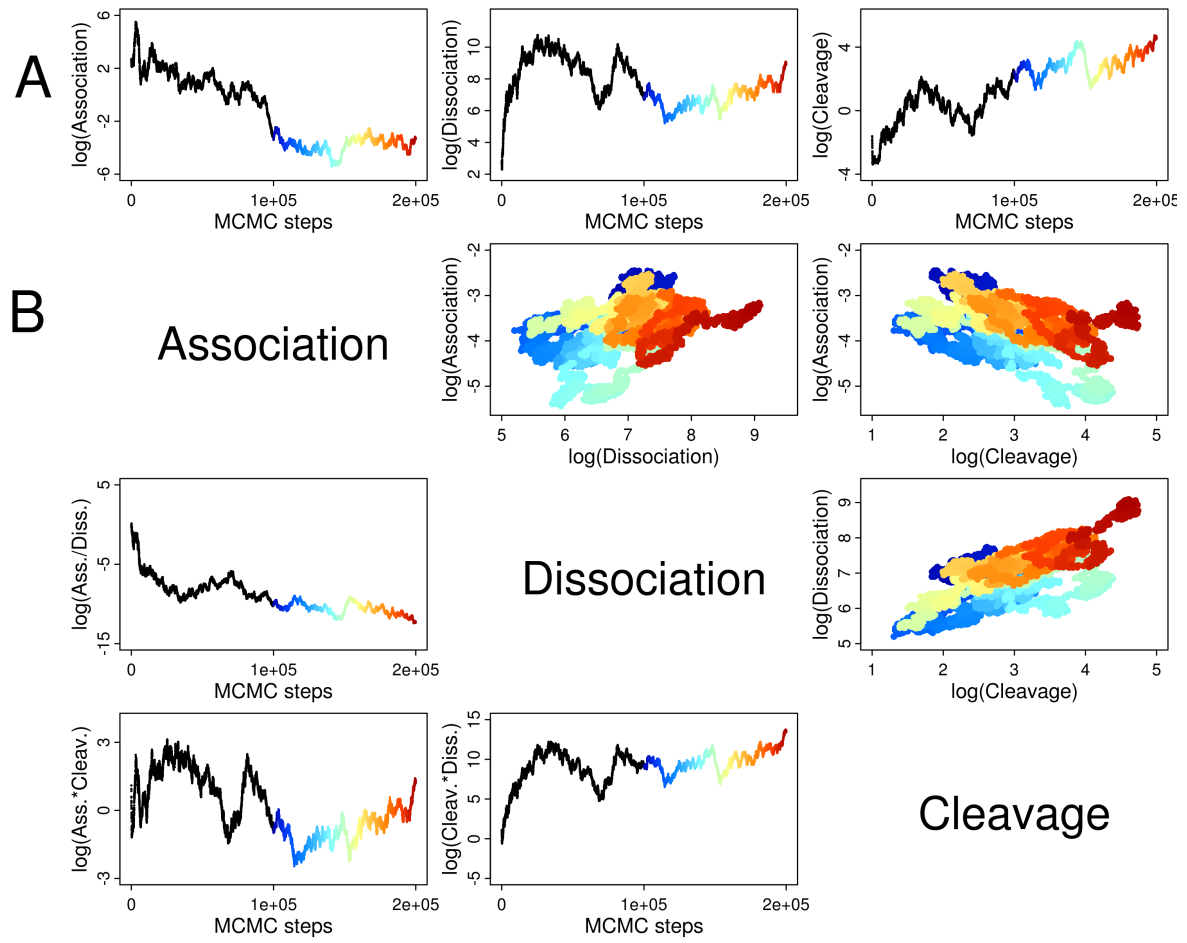
Figure 13.1.: Association, dissociation and cleavage can't be determined on an absolute value, but are related to each other. **(A)** The traceplots of the association, dissociation and cleavage parameters show that neither of them converges in the course of the sampling process. **(B)** In the top right corner, association, cleavage and dissociation parameters are plotted against each other. Here, the colors indicate the development in the course of the sampling process, as shown in the traceplots (bottom left corner). Importantly, note that even the derived parameters $\frac{association}{dissociation}$, $association \cdot cleavage$ and $cleavage \cdot dissociation$ do not converge well, they show a slight drift throughout the second half of the MCMC run.
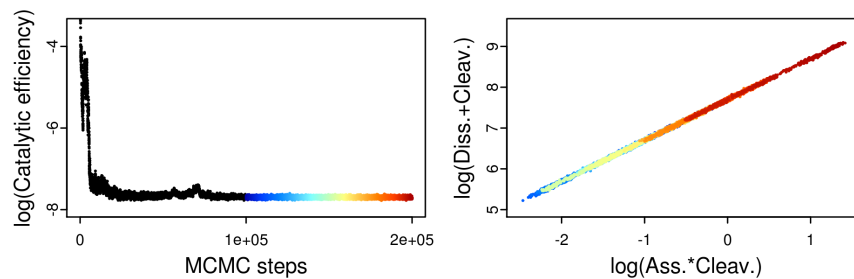
Figure 13.2.: The catalytic efficiency converges and has a narrow posterior distribution, as opposed to those of the individual parameters. The plot on the left-hand side shows the development of the catalytic efficiency $\frac{k_a \cdot k_c}{k_d + k_c}$ during the MCMC procedure, while on the right-hand side nominator and denominator are plotted against each other.
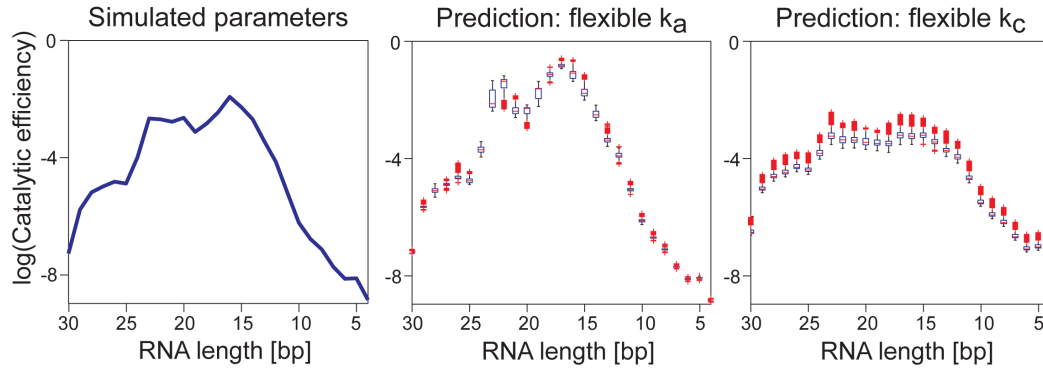
Figure 13.3.: The catalytic efficiency values can be derived by either sampling the association or the cleavage parameter individually for all RNA lengths, while the respective other one stays the same. Here, the two possibilities are compared, based on a simulated dataset with a noise level of $\alpha = 5\%$. The plot in the middle shows the predicted catalytic efficiencies when all association rates but only one cleavage rate is sampled, the one on the right shows the predicted catalytic efficiencies when all cleavage rates but only one association rate is sampled. On the left-hand side, the simulated catalytic efficiencies are depicted. It can be seen that the plot in the middle fits slightly better, although the basic curve stays the same. Combined with the biochemical reasons outlined before, this leads to the choice of association as the flexible parameter.
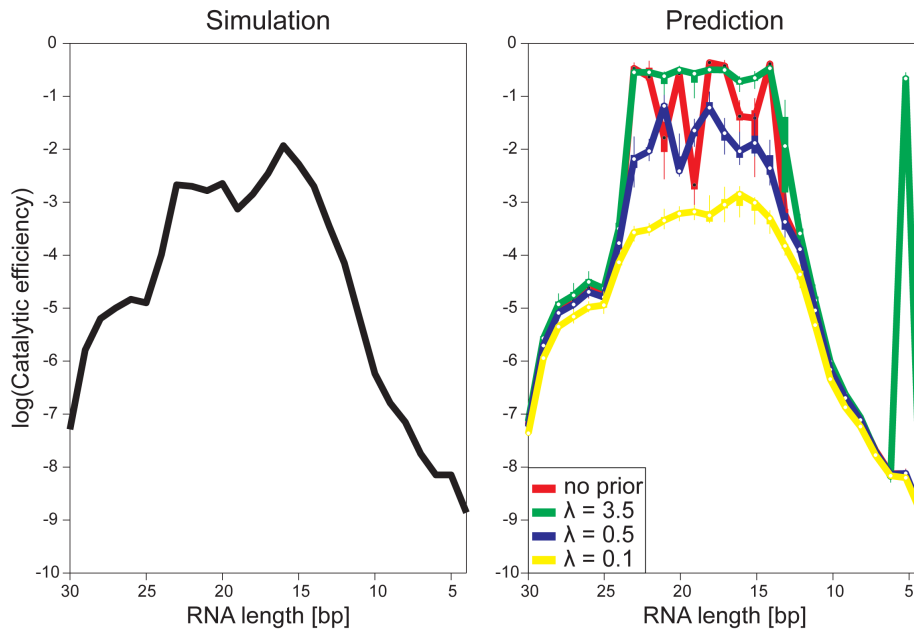
Figure 13.4.: This figure compares the influence of different smoothness priors for the catalytic efficiency parameters. On the left-hand side, the simulated catalytic efficiencies are shown, on the right-hand side the predicted catalytic efficiencies for different prior strengths are depicted. It is obvious that an absent prior (red) or a prior that is not strong enough (negligible prior strength, green: $\lambda = 3.5$) lead to a strong variation and jumps amongst the parameters, while a prior that is too strong (yellow: $\lambda = 0.1$) does not afford the flexibility needed to adapt the parameters correctly. Based on these results, our prior of choice is $\lambda = 0.5$ (blue).
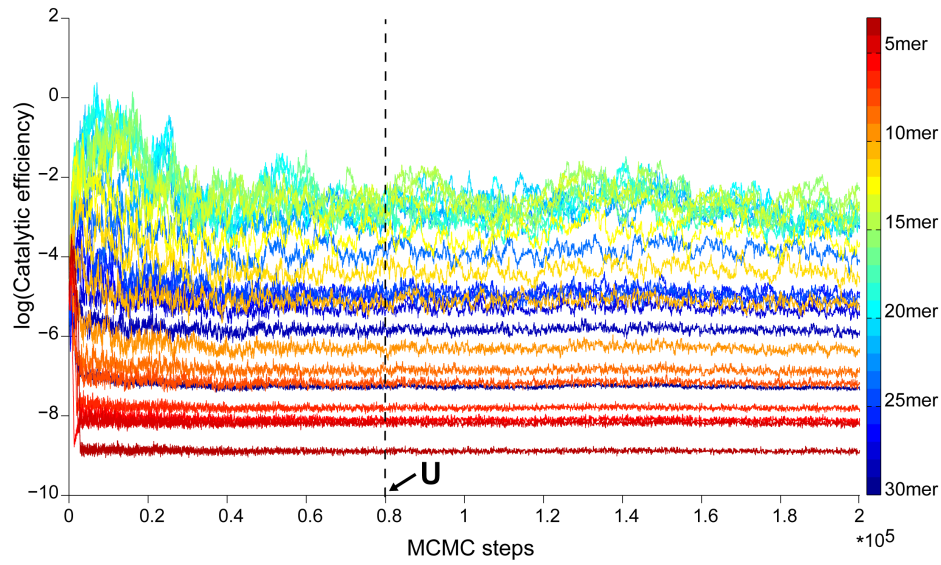
Figure 13.5.: The exemplary burn-in parameter U is chosen based on the catalytic efficiency traceplot, here for a simulation run ($2 \cdot 10^5$ MCMC steps and 27 catalytic efficiency parameters, depicted by different colors; noise level of the dataset: $\alpha = 25\%$). It indicates the boundary between burn-in phase ($s = 1, ..., U$) and stationary phase ($s = U + 1, ..., 2 \cdot 10^5$). Here, the challenge is to find a trade-off between fast converging parameters (the ones with lower values) and slow-converging parameters (the ones with higher values).
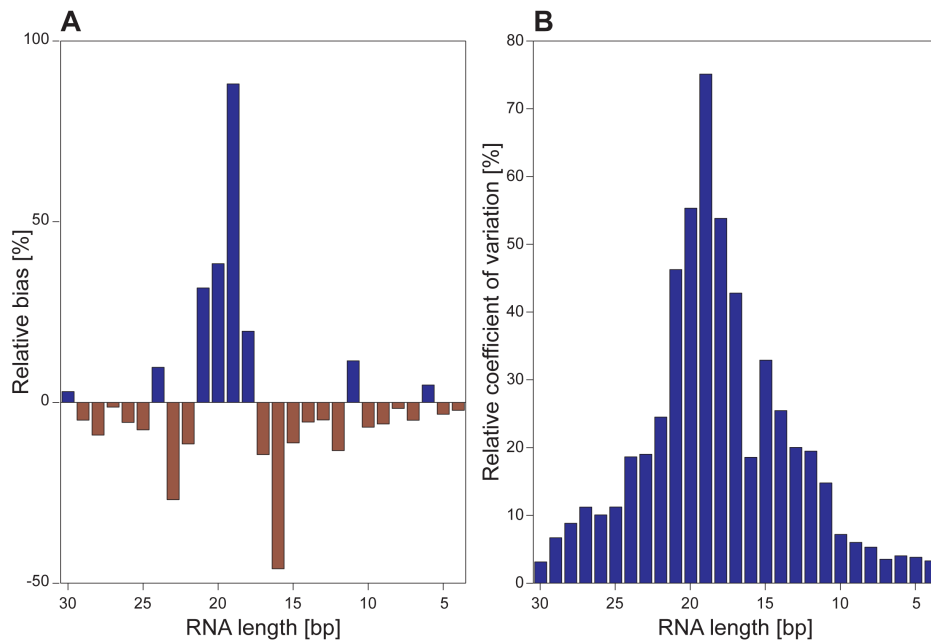
Figure 13.6.: For every catalytic efficiency parameter, the relative bias **(A)** and the coefficient of variation **(B)** are depicted, here for a dataset with a noise level of $\alpha = 25\%$. For the bias, red color indicates that the mean of the predicted parameters is shifted down with regard to the simulated parameter, blue color indicates that it is shifted up.

13.1.4. Assessment of bias and variance

For each catalytic efficiency value e_j, $j \in J$ the relative bias and variance respectively standard deviation of its marginal posterior distribution can be assessed as

$$\text{Bias}_j = \frac{\text{mean}(E_j) - e_j^{\text{true}}}{e_j^{\text{true}}} \ , \qquad \text{std}_j = \frac{\text{std}(E_j)}{e_j^{\text{true}}} \ , \tag{13.1}$$

with $E_j = \{e_{j,s}^m, \ s > U, \ m = 1, ..., M\}$. Bias and variance are visualized in Fig. 13.6.

13.1.5. Prediction quality

To assess the prediction quality of our method we simulated datasets based on the measurements for the interface mutant (see Fig. 10.1). Noise was added using varying error levels ($\alpha = 5\%$, 10%, 25% and $\beta = 0$). Then we applied the MCMC sampling approach to the noisy simulated datasets and drew 1000 parameter sets at random from the stationary phase of the Markov chain. The average of the predictions for these parameter sets has then be used to calculate the relative squared error $\left(\frac{|Data_{\text{real/noisy}} - Data_{\text{predicted}}|}{Data_{\text{real/noisy}}}\right)^2$, both between the predicted and the true simulated data as well as between the predicted and the noisy simulated data. The results are shown in Fig. 13.7 and Fig. 13.8.

13.1.6. Bayesian methods vs. least-squares fitting

We compared our Bayesian MCMC sampling approach to a straightforward optimization method for the minimization of the quadratic loss (using the MATLAB [69] function fminsearch, with the parameter MaxIter and MaxFunEval both set to $50000 \cdot |\Theta|$). Fig. 13.9 shows the relative squared error for the least-squares fitting approach (see Fig. 13.8 for the MCMC sampling's relative squared error). In Fig. 13.10, the quality and the reliability of the parameters estimated by the least-squares approach and by the MCMC sampling approach are compared. Bayesian sampling clearly exhibits less variance and a smaller bias than least-squares fitting.
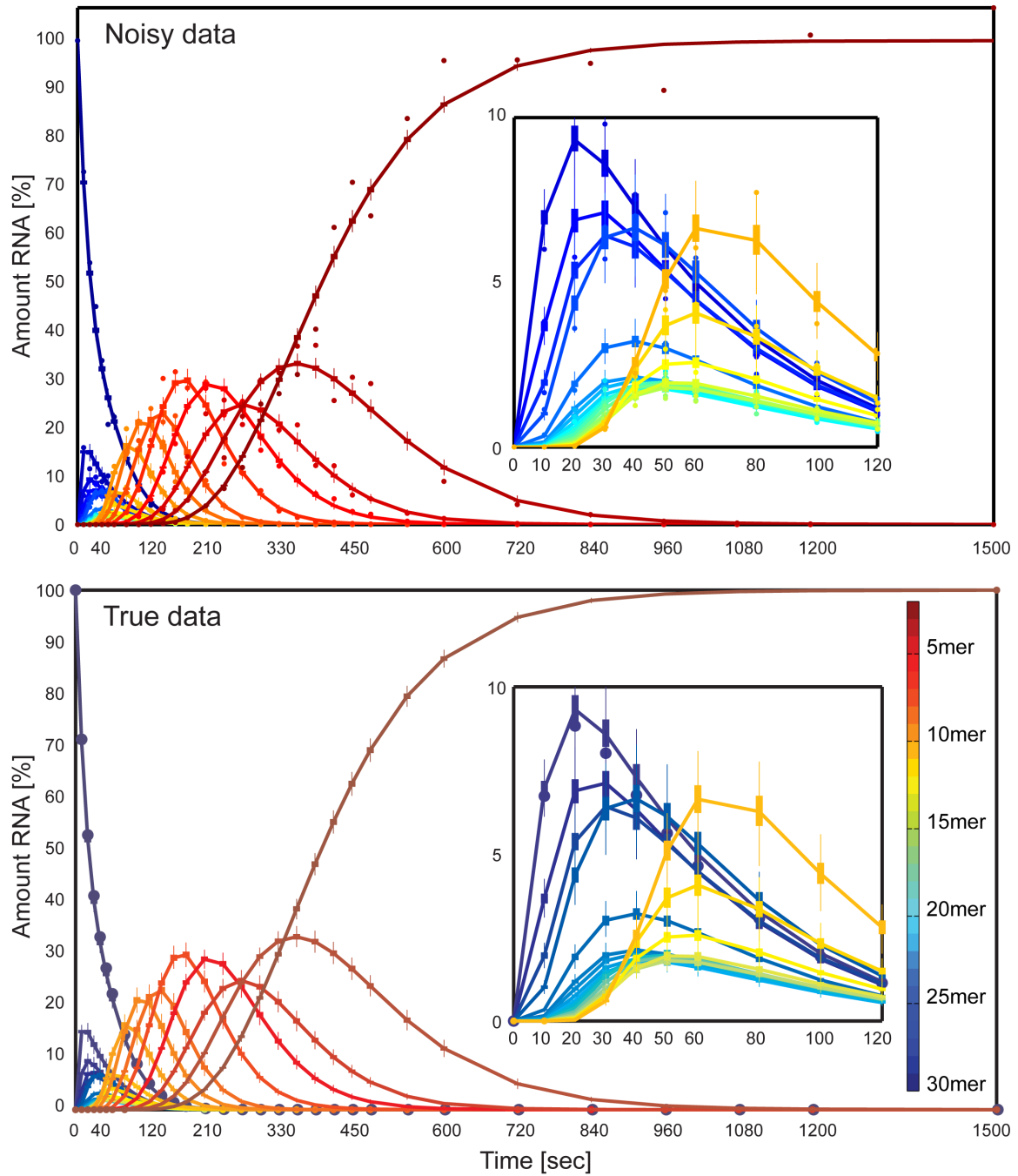
Figure 13.7.: Goodness of fit for a simulation run. Circles in the lower plot are the unperturbed time series measurements as produced by the simulated parameter set, while the circles in the upper plot are the noisy counterparts. The MCMC approach has been applied to the noisy data ($\alpha = 25\%$) and 1000 parameter sets have been randomly drawn from the stationary phase of the Markov chain. The predictions for these parameter sets are depicted by boxplots. It can be seen that our model offers a better description of the true simulated data (lower plot) than it does for the noisy simulated data.
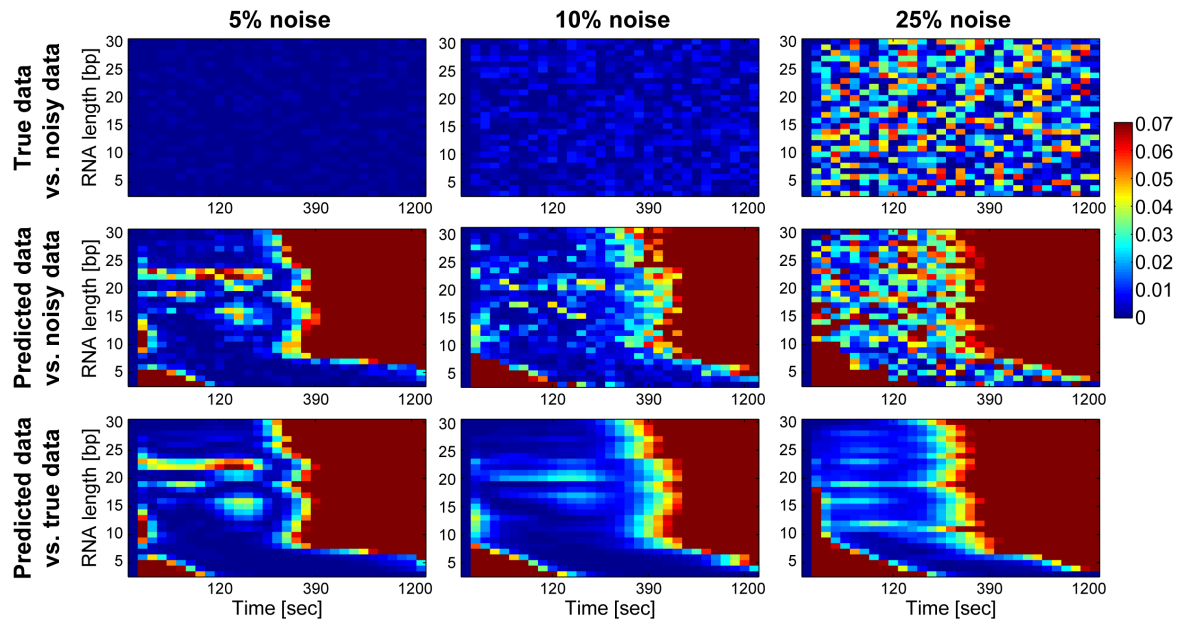
Figure 13.8.: The relative squared error induced by the MCMC sampling approach is calculated based on the averaged predictions of 1000 parameter sets, randomly sampled from the stationary phase of a Markov chain. For a more detailed visualization, the color scheme has been scaled such that all values ≥ 0.07 have the same color. Every column of diagrams corresponds to a given noise (5%, 10% and 25%) used to generate noisy datasets. The first row of diagrams displays how the noisy simulated data deviate from the true (without noise) simulated data, the second row displays the relative squared error of the estimate from the MCMC sampling compared to the noisy simulated data and the third row displays the relative squared error of the estimate from the MCMC sampling compared to the true simulated data. The results allow several conclusions: First, individual RNA measurements with higher values can be fitted very well, while areas with lower amounts of RNA are fitted relatively poorly. Second, our predictions fit the true simulated data better than the noisy simulated data, which has actually been used for fitting. Third, our procedure appears to be robust against considerable amounts of noise, since the increase in α does not visibly reduce the fit to the true simulated data.
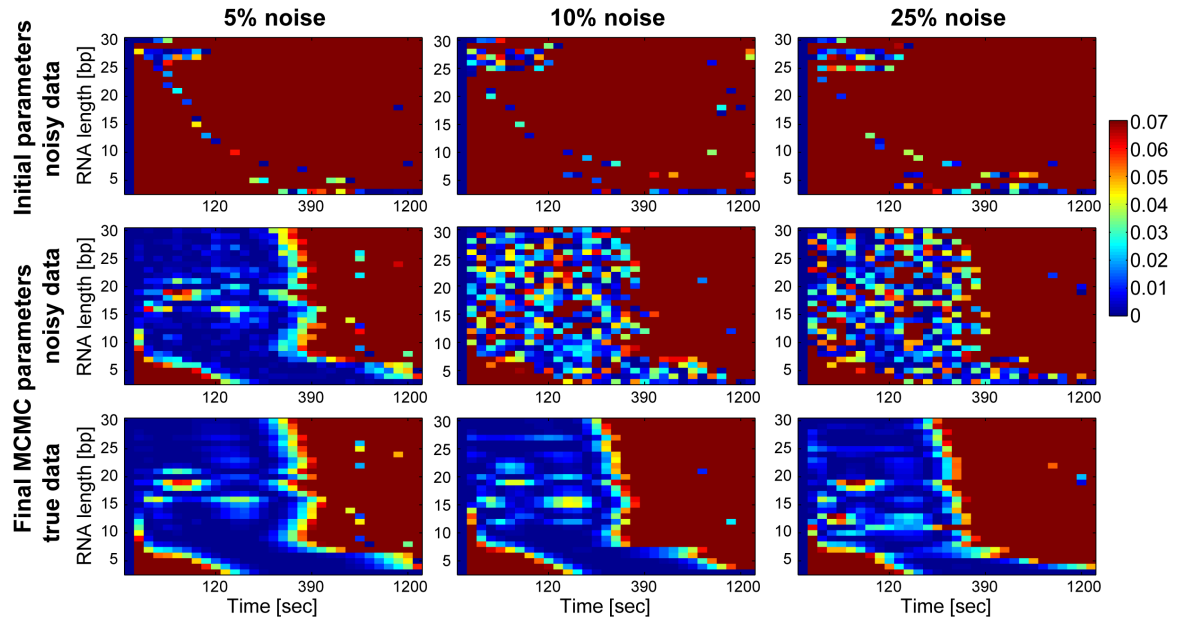
Figure 13.9.: The relative squared error induced by the least-squares fit is calculated based on predictions of the estimated parameters. For a more detailed visualization, the color scheme has been scaled such that all values ≥ 0.07 have the same color. Every column of diagrams corresponds to a given noise (5%, 10% and 25%) used to generate noisy datasets. Just as the MCMC sampling approach, the least-squares fitting has to be initialized with some set of parameters. In the first row of diagrams, the initialization is the same as for the MCMC sampling. The relative squared error is calculated with respect to the noisy simulated data. In the second row, the initialization of the least-squares fitting corresponds to the mean of a random sample from the stationary phase of the Markov chain (i.e., to a good set of parameters, see Fig. 13.8). Again, the relative squared error is calculated with respect to the noisy data. In the third row, the initialization is the same as in the second one, but the relative squared error is calculated with respect to the true simulated data. It is clearly visible, that while the straightforward optimization method yields good results (respectively, results that are similar to those of the MCMC approach) when the initial parameters are already close to the true ones, it performs very poorly when no prior information is available.
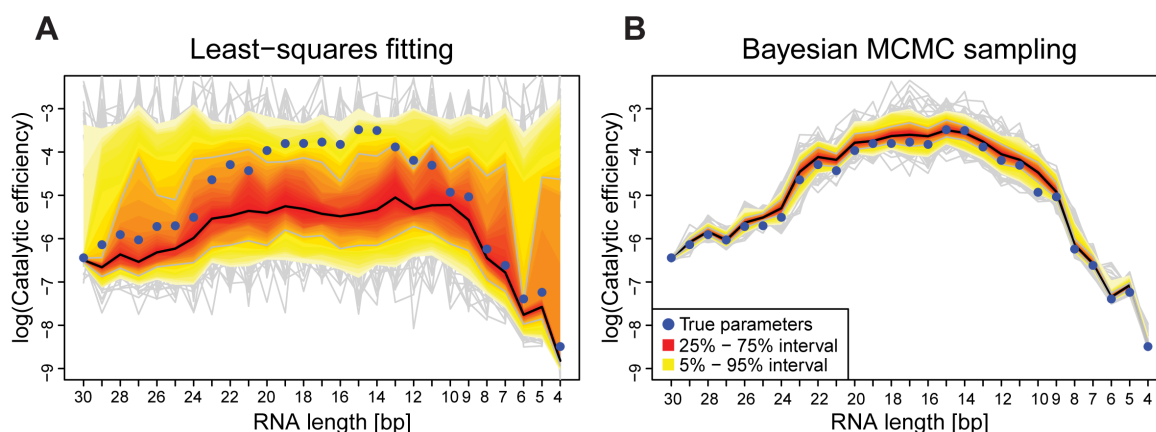
Figure 13.10.: Simulation study comparing the parameter estimation quality for (A) least-squares fitting and (B) Bayesian MCMC sampling. The blue points represent the simulated ("true") kinetic parameters that were fixed throughout the whole simulation. To obtain realistic parameters, they have been randomly sampled from a preceding MCMC run based on the data of the Rrp4-exosome. The corresponding dataset has been generated by applying measurements errors that have been estimated from the exosome data. The quantile profile plots are based on 100 least-squares estimates and 100 parameter sets that have been randomly drawn from different Markov chains, respectively: The red (yellow) band marks the central 90% (50%) intervals for the estimated catalytic efficiencies e_j, for $j = 30, ..., 4\,\mathrm{nt}$. **(A)** shows the central 90% (50%) intervals of the 100 least squares parameter estimates. **(B)** shows Bayesian confidence intervals, averaged over 100 runs. Bayesian sampling clearly exhibits less variance (narrower bands) and a smaller bias (bands are closer to the true parameters) than least-squares fitting.

13.2. Application: RNA degradation by the archaeal exosome

For every exosome variant we used an initial MCMC run to determine the set of parameters that, from a chain of 10^5 steps, best describes the data (least sum of squared errors). Those where used (1) to initialize two Markov chains for each exosome variant that converge very fast due to the the good initial parameters and that are used for further analysis (this is not just a continuation of the original Markov chain, since the error model is reset to the initial one) and (2) to initialize three additional runs for some other exosome variant to analyze the robustness of our approach with regard to the initial parameter values.

13.2.1. Assessment of parameter dependencies

As has been discussed before (Section 11.3), the association, cleavage and dissociation rates are highly redundant, which is illustrated in Fig. 13.11. This problem has been solved by the introduction of the catalytic efficiency as an identifiable parameter (see Fig. 13.12). These figures refer to the predicted parameters for the 30mer RNA for the Rrp4-exosome. See also Section 13.1.1 for results on simulated datasets.
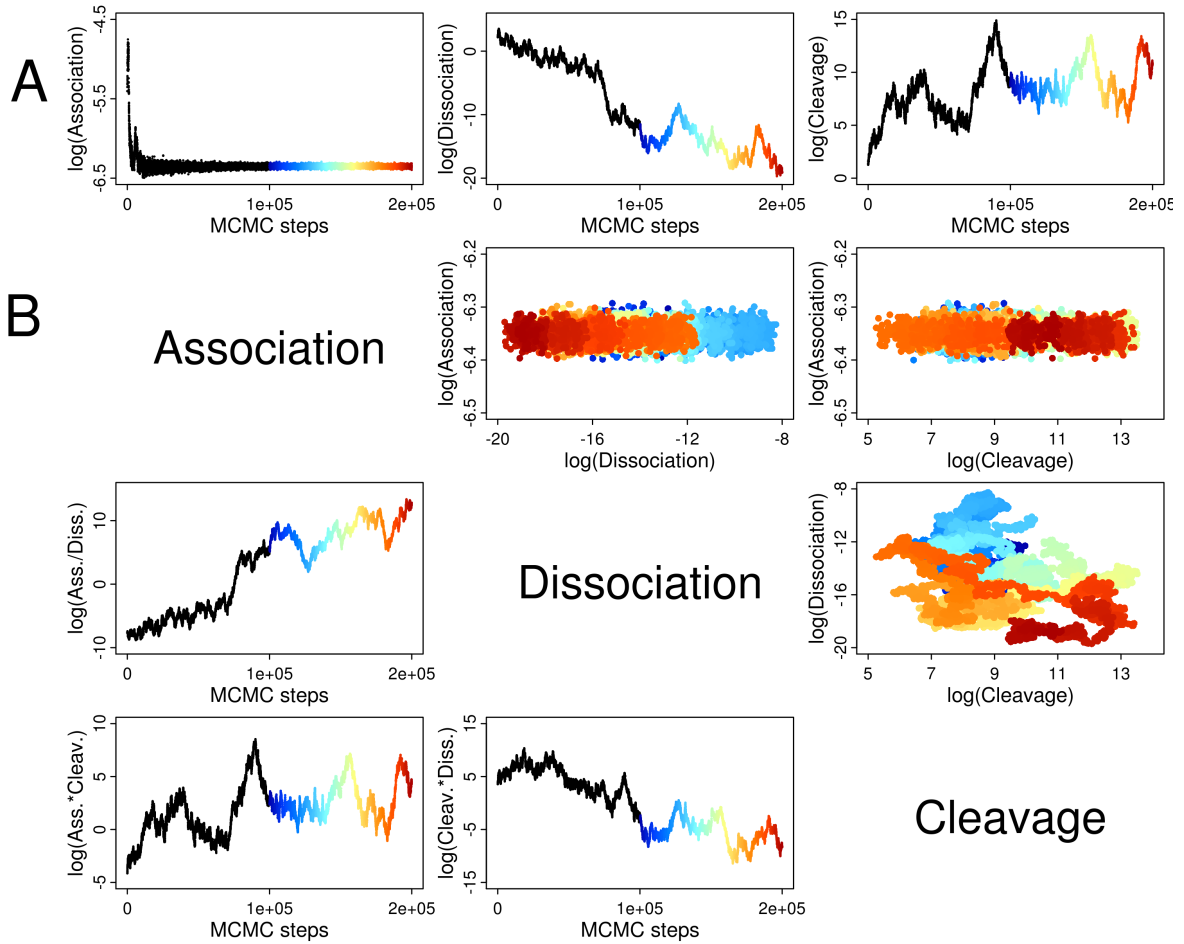
Figure 13.11.: Association, dissociation and cleavage can't be determined on an absolute value, but are related to each other. This figure is based on on the measurements for the Rrp4 exosome. **(A)** The traceplots of the association, dissociation and cleavage parameters show that only the association rate converges in the course of the sampling process. **(B)** In the top right corner, association, cleavage and dissociation parameters are plotted against each other. Here, the colors indicate the development in the course of the sampling process, as shown in the traceplots (bottom left corner). Importantly, note that even the derived parameters $\frac{association}{dissociation}$, $association \cdot cleavage$ and $cleavage \cdot dissociation$ do not converge well.
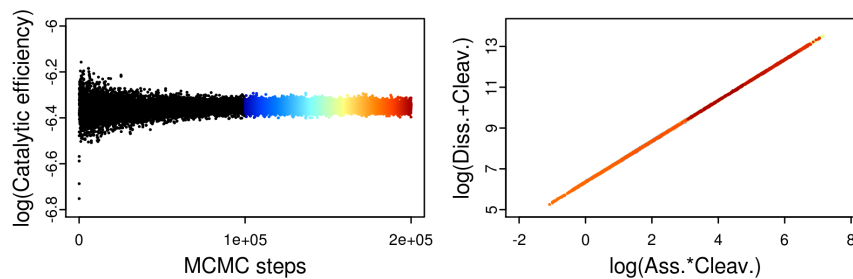
Figure 13.12.: The catalytic efficiency converges and has a narrow posterior distribution, as opposed to those of the individual parameters. The plot on the left-hand side shows the development of the catalytic efficiency $\frac{k_a \cdot k_c}{k_d + k_c}$ during the MCMC procedure, while on the right-hand side nominator and denominator are plotted against each other.
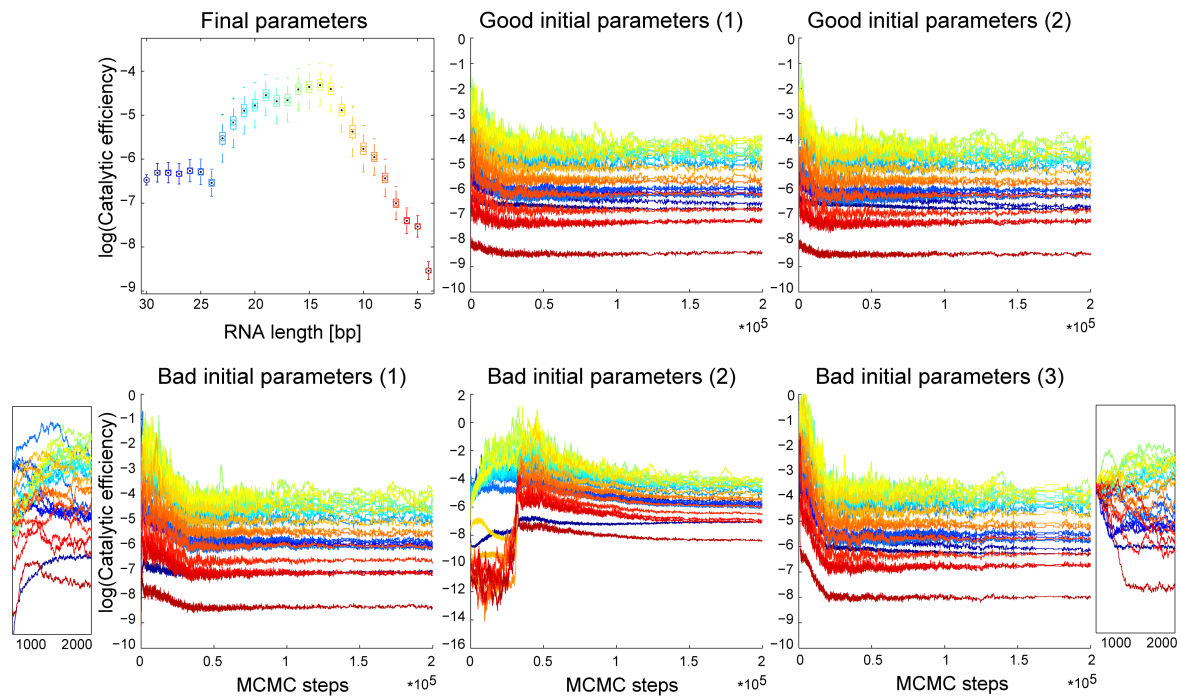
Figure 13.13.: The traceplots of five MCMC runs, based on the Rrp4 measurements and initialized with different sets of parameters, show that the result of the MCMC sampling is independent of initialization. A common catalytic efficiency distribution (first plot) is obtained in all cases, whether the initial parameters are close to the final ones (second and third plot), based on the results obtained for another exosome mutant (fourth and fifth plot) or set to the same value (sixth plot). The beginning of the chain is enlarged for the fourth and the sixth plot to emphasize the differences between the initial and the final parameters.
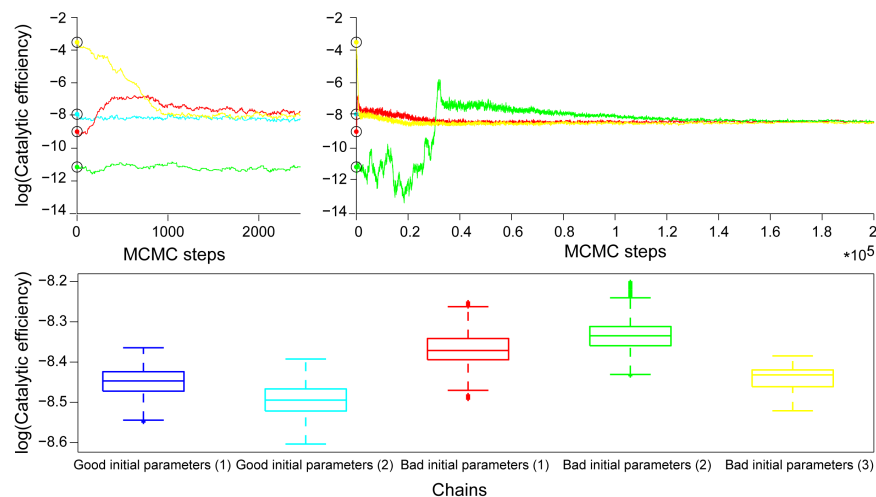
Figure 13.14.: The traceplots of five MCMC runs for the 4mer RNA, based on the Rrp4 measurements and initialized with 4 different sets of parameters (top right corner), show that the initial parameters (black circles) have no influence on the result of the sampling procedure. The first 2500 steps are enlarged in the top left corner. Boxplots of the stationary phases of the 5 Markov chains are depicted in the third plot (please note the changed scaling of the y-axis).
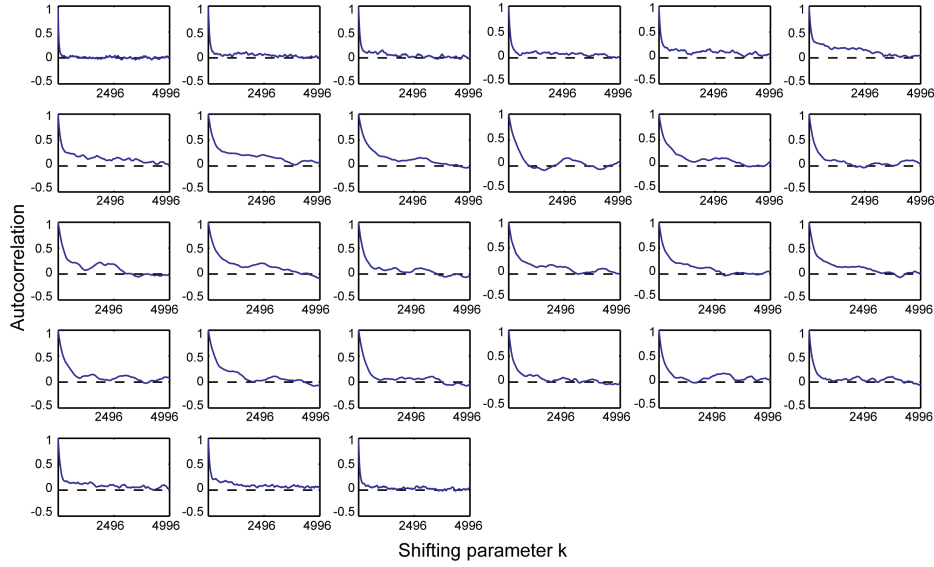
Figure 13.15.: Autocorrelation plots of the catalytic efficiency values for an MCMC chain in stationary phase, based on the Rrp4 data. Each plot shows the autocorrelation of the parameters for one RNA population, ranging from 30mers (top left corner) to 4mers (bottom right corner). The x-axes represent the values of k, the y-axes represent the autocorrelation values. Further information on the calculation of the autocorrelation values is provided in the main text.

13.2.2. Assessment of the MCMC sampling behavior

Robustness w.r.t. initial parameter values

Fig. 13.13 displays the traceplots for five different MCMC runs, based on the Rrp4 measurements and initialized with different sets of parameters. Even though the convergence time depends on the choice of these initial parameters, a common invariant distribution is always reached in reasonable time ($\sim 50000 - 100000$ steps). The chain for the 4mer catalytic efficiency is extracted in Fig. 13.14 for a more detailed analysis.

Mixing

An important criteria for the quality of the sample is the "mixing". Mixing describes the efficiency with which we sample from the whole distribution, i.e., the speed with which the empirical distribution of n consecutive individual parameter samples $\Theta_1, ..., \Theta_n$ from a Markov chain converges against the true posterior distribution. The autocorrelation function [20] is an indicator for the mixing behavior. It can be estimated by

$$a(k) = \frac{1}{(n-k)\sigma^2} \sum_{t=1}^{n-k} (\Theta_t - \bar{\Theta})(\Theta_{t+k} - \bar{\Theta}) , \qquad (13.2)$$

with sample mean $\bar{\Theta}$ and sample variance σ^2. The quantity $a(k)$ tells how a sequence of numbers correlates with a copy of itself which has been shifted by k entries. If the entries of the sequence were totally uncorrelated, $a(k)$ should be approximately zero for all k. In

a sequence that has been generated by a MCMC run, consecutive entries are correlated by construction, but it is a desirable property of a MCMC chain that this correlation decreases rapidly for entries that are $k > 1$ places apart from each other. That is, the sooner $a(k)$ decreases to zero, the better the mixing behavior of the chain. A plot of the autocorrelation functions for all parameter samples $K_{x,j}$ is shown in Fig. 13.15.

13.2.3. Results

The traceplots as well as the boxplots for the final posterior distributions (the burn-in has been set to 150000) for all exosome variants (see Section 10.2 for details) are shown in the Appendix, Fig. B.1. The convergence speed as well as the variance in the stationary phase (i.e., the identifiability of the parameters) vary among the exosome variants. Appendix Table B.1 and Appendix Table B.2 specify median, 1^{st}quartile and 3^{rd}quartile of the catalytic efficiency posterior distributions for all RNA lengths. Appendix Fig. B.2 and Appendix Fig. B.3 depict the relative squared error $\left(\frac{|Data_{real} - Data_{predicted}|}{Data_{real}} \right)^2$ of the predictions derived by our MCMC approach and by the straightforward least-squares optimization method (see Section 13.1.6) compared to the original measurements. The results for the R65E mutant of the Csl4-exosome have turned out to be highly sensitive to small changes of the model specification, hence hardly reliable (data not shown). This variant has thus been excluded from further analysis. In the following, I will summarize the main results of the MCMC sampling approach applied to the RNA degradation by the archaeal exosome. For a more detailed explanation of the results and their biochemical foundations and implications please refer to [36], since this is not within the scope of this thesis.

The efficiency of RNA degradation depends on the length of the substrate

The results of the MCMC sampling confirmed what a first glance at the data (see Fig. 10.2) had already suggested: The efficiency of RNA degradation depends on the length of the substrate (see Fig. 13.16 and Appendix, Fig. B.1). On the whole, three main phases can be identified. First, the initial degradation steps are very slow, presumably restricted by the initial threading of the substrate in the central chamber. Second, the degradation of medium length RNA molecules that are already bound is fast and relatively constant. Third, the degradation efficiency decreases rapidly for short RNAs. This comparatively slow decay of short RNAs may be explained by the exosome structure. While longer RNAs might still reach the neck structure while being degraded, which would have a stabilizing influence, short RNAs might lose this contact and therefore easily diffuse from the active site. This hypothesis is addressed in the next paragraph. The exact meaning of "long", "medium" and "slow" depends on the exosome variant, in general the decay efficiency decreases at around 13 nucleotides.
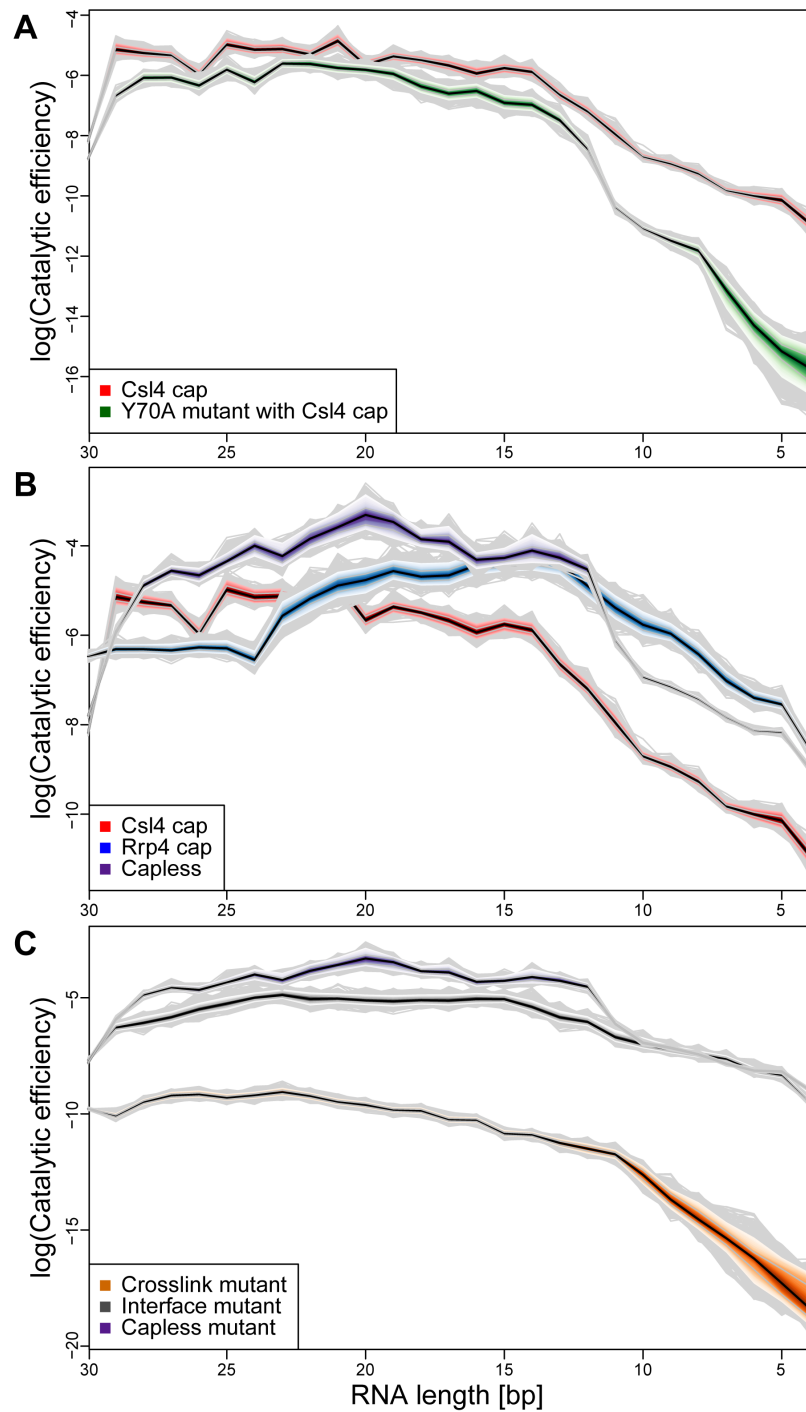
Figure 13.16.: Comparison of the exosome variants. The quantile profile plot shows the 50% Bayesian confidence intervals (dark color) and the 100% intervals (light color) of the predicted catalytic efficiencies. If two bands do not overlap at some length i, this means that under the assumptions of the model, the catalytic efficiencies e_i differ with an estimated probability of 99.99%. For all variants, the catalytic efficiency depends on the length of the RNA substrate. **(A)** Tyr70$^{\text{Rrp42}}$ is supposed to be important for RNA binding. The Y70A mutant of the Csl4-exosome (green) thus supports the hypothesis that long RNAs are additionally stabilized by the neck. **(B)** The degradation efficiency depends on the cap structure (red: Csl4-exosome, blue: Rrp4-exosome, purple: capless exosome). **(C)** Comparing the crosslink mutant (orange) and the interface mutant (black) suggests that conformational flexibility of the ring is important for efficient RNA degradation.

Tyr70^{Rrp42} is important for RNA binding

Tyr70^{Rrp42} close to the active site is supposed to be important for RNA binding [35]. To test the hypothesis that longer RNAs are additionally stabilized by the neck structure we analyzed the Y70A mutant of the Csl4-exosome, where the tyrosine is mutated to an alanine. The results are depicted in Fig. 13.16A. Even though the overall shape is the same, the catalytic efficiency is lower for the mutant than for the wild-type exosome for all RNA lengths. However, the difference is increasingly pronounced, the shorter the RNA molecules are: While for long RNAs (> 13nt) the difference is only $\sim 2-$ to $3-$fold (about 1 log unit), it is $\sim 20-$ to $150-$fold (about 3 to 5 log units) for shorter RNAs. This is consistent with the hypothesis that long RNAs are additionally stabilized by the neck structure, while shorter RNAs lose this contact and fully rely on the active site. That the additional stability is actually provided by the neck rather than by the cap structure is illustrated by the degradation profile of the capless exosome (see for example Fig. 13.16B): The capless exosome shows the same decrease in degradation efficiency for short RNAs as the Csl4- and the Rrp4-exosomes.

The cap-structure influences the degradation efficiency

Next we analyzed the influence of the cap structure on the degradation efficiency by comparing the Csl4-exosome, the Rrp4-exosome and the capless exosome (see Fig. 13.16B). The initial degradation step is considerably faster for the Rrp4 exosome as compared to the two other variants ($\sim 7-$fold, about 2 log units). This indicates that Rrp4 recruits the RNA molecules more efficiently, possibly due to a more specific binding site. However, the same binding site seems to prevent the RNA molecules from proceeding further towards the active center since, in the next steps, RNA degradation by the Rrp4-exosome is significantly slower as compared to the Csl4-exosome ($\sim 3-$ to $7-$fold , about 1 to 2 log units). Surprisingly, exosomes with different cap structures also differ in their efficiency to degrade small RNAs which are too short to be in contact with the active site and the cap proteins at the same time. From the crystal structures for both exosome variants, we know that the Rrp4 protein interacts more intimately with the ring of the processing chamber than Csl4 [14]. It is not unlikely that this interaction can influence the flexibility of the core ring and hence the degradation dynamics. Furthermore, compared to the Csl4-exosome, the catalytic efficiency of the capless exosome is significantly higher for all intermediates, indicating that the Csl4-cap has no stimulating influence on the degradation of the poly(rA) substrate.

The state of the ring structure influences the degradation efficiency

To test the influence of the ring structure on the degradation efficiency, we compared the crosslink mutant (with a rigid ring structure, but with the same size and shape as the capless exosome) and the interface mutant (which can't form the ring structure any more and results in three stable Rrp41:Rrp42 heterodimers) to the corresponding wild-type, the capless exosome (see Fig. 13.16C). While the degradation efficiency of the crosslink mutant

decreases considerably ($\sim 500-$ to $2000-$ fold) across the whole range, the decrease induced by the interface mutant is, if at all, only marginal. This observation suggests that the ring architecture needs to breathe or display some conformational dynamics to increase the size of the neck. Only after the RNA molecule is bound in the neck, the protein ring might tightly close around its substrate. This flexibility is not provided in the rigidified crosslink mutant. Furthermore, the interface mutant seems to compensate the higher dissociation rate induced by the open conformation with higher association and cleavage rates: while in the wild type exosome only one RNA molecule can be degraded at a time, three active centers are accessible here simultaneously.

13.2.4. Summary & Outlook

Although the analysis described here has been tailored to the kinetics of the RNA degradation by the archaeal exosome, it is by no means limited to this system. Needless to say, the Bayesian sampling method is well suited to address the RNA degradation by the eukaryotic exosome, for instance to reveal the interplay between endo- and exonuclease activities [61, 85, 87] and biochemical differences between different Dis3/Dis3L isoforms as well as Rrp6 of the human exosome complex [71, 94, 100]. Related systems of interest are the degradation of 5' ends by XRN1, RNA degradation by bacterial degradosomes, or in general any system that involves the (semi)processive synthesis or degradation of biopolymers.

In the future, it would be interesting to try a prior based proposal approach where the association parameters are sampled correlated. In this way, the smoothness prior is directly incorporated in the proposal which might yield faster convergence. In addition, an extension of the experimental setup with respect to different substrates would be interesting to analyze the influence of the different cap structures.

Part IV.

Statistical analysis of a cellular decision process: Differentiation of hematopoietic stem cells

14. Introduction

Cytokines play a major role during the generation of blood cells (hematopoiesis). Yet, the precise nature of their contribution is still unclear: In the selective (also: permissive) scenario they allow the survival and proliferation only of specific cell types. In this case, the commitment of a cell is determined purely intrinsically by transcription factors, independent of external signals. In the instructive scenario, cytokines already influence the differentiation process itself via specific signaling pathways [81]. Both scenarios lead to the same result: In the end, there exists one predominant lineage - whether this is due to different cell death (selective scenario) or differentiation rates (instructive scenario). As a consequence, it is not obvious which scenario to prefer. This question is addressed by several publications [32, 74, 80, 84].

Stem cell genealogical trees (in the following shortly genealogies) can provide an answer. They trace the development of a stem cell including cell division events, cell death events, and, ideally, the differentiation in various lineages. An example of a genealogy is provided in Fig. 15.1A. So far, the amount and quality of the available data have been limited due to technical restrictions. Yet, a bioimaging approach has been presented recently that enables long-term observations on the single-cell level [80]. It produces a series of hematopoietic stem cell genealogies including information on "stemness" and lineage commitment of the individual cells. Such technical advances and the consequential increase in (new kinds of) data raise the need for new analysis methods.

In this context, we developed a factor graph model for stem cell genealogies in combination with a reversible-jump Markov Chain Monte Carlo sampling algorithm to infer the predominant scenario as well as the corresponding lineage specific differentiation and cell death rates. The factor graph model has been developed within the scope of a bachelor thesis I supervised (see [103]), I will briefly summarize the main points here. The reversible-jump MCMC sampling is work in progress, the manuscript for a publication is currently prepared. Again, the implementation is done by a student assistant I supervise, Diana Uskat. She also prepared the figures included in this part. For these reasons, I will only provide a short outline on this project focusing on the development of the reversible-jump algorithm. Furthermore, I will present some preliminary results on simulated datasets. They show that our method is able to reliably infer the predominant scenario as well as the corresponding differentiation and cell death rates. At present, we apply the approach to simulation data produced by a more elaborate model [63, 83] and to a recently published dataset [80]. These results will be included in the publication.

15. A factor graph model for hematopoietic stem cell differentiation

We developed a simple model to describe the differentiation of hematopoietic stem cells including two possible lineages (see Fig. 15.1B, right-hand side). The genealogies based on this model are represented as factor graphs as explained in the following.

15.1. Factor graphs

Factor graphs are bipartite graphical models that consist of two different types of nodes (variable nodes and factor nodes) and edges that are only allowed between nodes of different types. Variable nodes represent the variables, here the cells and their properties at predefined time points. In this model, we consider three possible cell types: Stem cells, cells differentiated to lineage 1 and cells differentiated to lineage 2. Factor nodes establish relationships among the variable nodes they are connected to, i.e., they provide probabilities for potential transitions between cells. The factor nodes included in this model are depicted in Fig. 15.1B. The high-dimensional joint posterior distribution for a given genealogy can thus be easily factorized into a multitude of local, low-dimensional distributions. These refer only to neighboring variable nodes (where neighboring means being connected by the same factor node) and hence are easier to derive. In particular, the local probability distributions are based on only three possible transitions and do not increase the number of parameters to be estimated (see also Section 1.2). The main merit of factor graphs is the existence of very fast algorithms for the calculation of the highest probability and the corresponding variable setting (max-sum algorithm), or for the calculation of marginal probability distributions (sum-product algorithm) [8, Chapter 8]. The basic factor graph model visualizes exactly the genealogy derived from the cell tracking process. The variable nodes correspond to the cells, and the factor nodes correspond to the observed transitions. A second set of factor nodes, one for each variable node, incorporates additional information with regard to the properties of the cells, i.e., whether a cell still has its stem cell properties or whether it already differentiated (respectively, to which degree). This scheme is depicted in Fig. 15.1C. The observed data thus provide the topology of the factor graph, as well as the probability distributions for the properties of the individual cells. We account for measurement errors by defining these distributions accordingly, e.g., probabilities of 1 or 0 are avoided.

A detailed introduction to factor graphs and the corresponding algorithms can be found

in [8, Chapter 8]. Their adaption to the differentiation of hematopoietic stem cells has already been discussed in the bachelor thesis of Diana Uskat (see [103]) and is not part of this thesis.

15.2. General parametrization

The following parameters determine the general differentiation model (see also Fig. 15.1B):

$$\Theta = \left\{ p_{\text{diff}_1}, p_{\text{diff}_2}, p_{\text{death}_1}, p_{\text{death}_2}, p_{\text{div}_0}, p_{\text{div}_1}, p_{\text{div}_2} \right\},$$

where p_{diff} is the probability that a stem cell differentiates (in lineage 1 or 2, respectively), p_{death} is the probability that a cell dies (possible for lineage 1 or 2, but not for stem cells) and p_{div} is the probability that a cell divides. The following conditions have to be met:

$$
\begin{aligned}
1 &= p_{\text{diff}_1} + p_{\text{diff}_2} + p_{\text{div}_0} + p_{\text{cont}_0} \text{ (Stem cells)} \\
1 &= p_{\text{death}_1} + p_{\text{div}_1} + p_{\text{cont}_1} \text{ (Lineage 1)} \\
1 &= p_{\text{death}_2} + p_{\text{div}_2} + p_{\text{cont}_2} \text{ (Lineage 2)}
\end{aligned}
\tag{15.1}
$$

i.e., the probabilities for all possible events for one cell type have to add up to one. p_{cont} constitutes the probability that the cell's properties won't change from one time point to the next one (i.e., it will not die, differentiate or divide). Since its value is uniquely determined by the values of the other probabilities, it is not part of the model's parametrization.

15.3. Model selection

The aim of this work is not to infer all parameters individually, but to decide whether differentiation is driven by a selective or an instructive mechanism. The selective scenario states that even though different lineages evolve equally, selection pressure favors one of them. In the mathematical model, this scenario is represented by identical differentiation probabilities $p_{\text{diff}_1} = p_{\text{diff}_2} = p_{\text{diff}}$, but different probabilities for cell death p_{death_1} and p_{death_2}. The instructive scenario states that all lineages are equally probable to die, but that one of them evolves with a higher probability. In the mathematical model, the instructive scenario is thus represented by identical cell death probabilities $p_{\text{death}_1} = p_{\text{death}_2} = p_{\text{death}}$, but different differentiation probabilities p_{diff_1} and p_{diff_2}.

Obviously, the two scenarios have to be described by two different model classes with different parametrizations: The model class for the selective scenario M_{sel} includes the parameters $\Theta_{\text{sel}} = \{p_{\text{diff}}, p_{\text{death}_1}, p_{\text{death}_2}, p_{\text{div}_0}, p_{\text{div}_1}, p_{\text{div}_2}\}$, while the model class for the instructive scenario M_{instr} includes the parameters $\Theta_{\text{instr}} = \{p_{\text{diff}_1}, p_{\text{diff}_2}, p_{\text{death}}, p_{\text{div}_0}, p_{\text{div}_1}, p_{\text{div}_2}\}$. In this situation, it is not possible to apply the standard Metropolis-Hastings MCMC approach since the model classes have different dimensions and are not directly comparable. This problem is solved by reversible-jump MCMC sampling which integrates both model selection and parameter estimation into one sampling process.
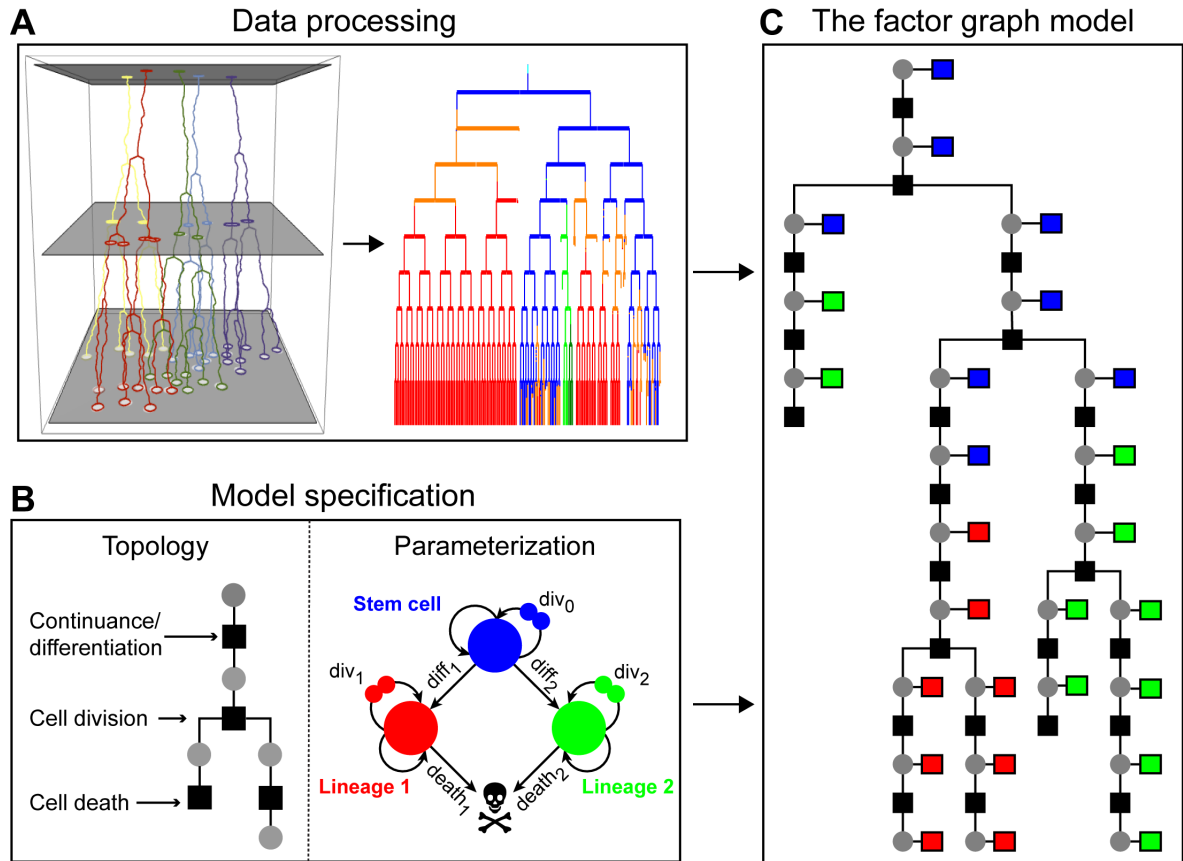
Figure 15.1.: **(A)** shows the basic data processing: Cell tracking produces a genealogy providing all information that can be derived by the experimental setup. The schematic representation of the cell tracking has been provided by the group of Ingo Röder. **(B)** shows the model specification. Factor nodes are depicted as squares, variable nodes are depicted as circles (left-hand side). Three types of factor nodes (i.e., cell division, cell death, and continuance/differentiation) can be formed by a simple model including probabilities for cell division, differentiation and cell death (right-hand side). Note that the topology alone provides no information on whether a stem cell persists (continuance) or differentiates. Colors indicate the cell type: stem cells are colored in blue, cells differentiated to lineage 1 or 2 are colored in red and green, respectively. Note that our model excludes cell death events for stem cells. **(C)** shows the factor graph model (variable nodes: cells, black factor nodes: transitions) for a given observation (genealogy). Additional factor nodes (colored), each connected to exactly one variable node, provide probability distributions for the characteristics of the cells (i.e., whether it is a stem cell or already differentiated to a given cell type), according to what can be measured during cell tracking (see Chapter 14).

16. Parameter estimation using reversible-jump MCMC sampling

For numerical stability, all calculations were carried out in log-space. An introduction to MCMC sampling in general can be found in Section 2.4. The reversible jump algorithm includes steps suggesting jumps between model classes as well as pure parameter sampling steps within the same model class. Here, this is realized by drawing $u \sim \mathcal{U}_{[0,1]}$ at every step to decide whether a model jump is suggested ($u \leq \alpha$), or whether the model class is maintained and a new set of parameters is suggested ($u > \alpha$). The percentage of model jumps is set to $\alpha = 0.05$. In the following, both situations are explained in detail.

16.1. Jumps between model classes

Changing the model class involves comparing one common cell death and two lineage-specific differentiation probabilities in the instructive scenario with one common differentiation and two lineage-specific cell death probabilities in the selective scenario. This is not possible and so the parameter spaces of the two scenarios have to be adjusted such that each scenario contains two differentiation related and two cell death related parameters. In the following, the application-specific implementation of the reversible-jump algorithm is explained in detail. For the general case, please refer to Section C.1.

16.1.1. Jumping from the selective to the instructive scenario

Jumping from the selective (one common differentiation probability, two lineage-specific cell death probabilities) to the instructive scenario (one common cell death probability, two lineage-specific differentiation probabilities) involves suggesting a parameter set $\Theta'_{\text{instr}} = \{p'_{\text{diff}_1}, p'_{\text{diff}_2}, p'_{\text{death}}, p'_{\text{div}_0}, p'_{\text{div}_1}, p'_{\text{div}_2}\}$ based on the current parameter set $\Theta^n_{\text{sel}} = \{p^n_{\text{diff}}, p^n_{\text{death}_1}, p^n_{\text{death}_2}, p^n_{\text{div}_0}, p^n_{\text{div}_1}, p^n_{\text{div}_2}\}$.

Extension of the parameter sets

To ensure a common measure, the parameter sets have to be extended with the parameters Δ_{diff} and Δ_{death} as follows: $\bar{\Theta}^n_{\text{sel}} = \{p^n_{\text{diff}}, \Delta_{\text{diff}}, p^n_{\text{death}_1}, p^n_{\text{death}_2}, p^n_{\text{div}_0}, p^n_{\text{div}_1}, p^n_{\text{div}_2}\}$ and $\bar{\Theta}'_{\text{instr}} = \{p'_{\text{diff}_1}, p'_{\text{diff}_2}, p'_{\text{death}}, \Delta_{\text{death}}, p'_{\text{div}_0}, p'_{\text{div}_1}, p'_{\text{div}_2}\}$.

The model jump then involves the following assignments:

$$
\begin{aligned}
p'_{\text{diff}_1} &= p^n_{\text{diff}} + \Delta_{\text{diff}} \\
p'_{\text{diff}_2} &= p^n_{\text{diff}} - \Delta_{\text{diff}} \\
p'_{\text{death}} &= \frac{p^n_{\text{death}_1} + p^n_{\text{death}_2}}{2} \\
\Delta_{\text{death}} &= \frac{p^n_{\text{death}_1} - p^n_{\text{death}_2}}{2}
\end{aligned}
$$

where $\Delta_{\text{diff}} \sim \mathcal{U}_{[-\frac{p^n_{\text{diff}}}{2}, \frac{p^n_{\text{diff}}}{2}]}$. The Jacobian for this transformation is 1 (see Appendix Section C.1 for a derivation). The division probabilities remain unchanged. The conditions $p'_{\text{death}} + p'_{\text{div}_1} \leq 1$ and $p'_{\text{death}} + p'_{\text{div}_2} \leq 1$ are met by definition, see Section 16.2.1.

Calculation of the acceptance probability

The general reversible-jump acceptance probability is explained in the Appendix, Section C.1. Here, the model jump is accepted with probability $\min(A_{\text{sel}\rightarrow\text{instr}}, 1)$, with

$$
A_{\text{sel}\rightarrow\text{instr}} = \frac{\text{sum-product}(\bar{\Theta}'_{\text{instr}})}{\text{sum-product}(\bar{\Theta}^n{}_{\text{sel}})} \cdot 1 \cdot \frac{\frac{1}{p'_{\text{death}}}}{\frac{1}{p^n_{\text{diff}}}} \cdot 1
$$

and sum-product(Θ) being the output of the sum-product algorithm for a given parameter set Θ, i.e., the probability for this set of parameters. The model jump is unambiguous since only two different model classes are available and the probability for suggesting a model jump is the same for both scenarios, hence $q(M_{\text{sel}}|M_{\text{instr}}) = q(M_{\text{instr}}|M_{\text{sel}}) = \alpha$. Yet, $q_{\text{sel}\rightarrow\text{instr}}(\Delta_{\text{diff}}|\Theta^n_{\text{sel}})$ and $q_{\text{instr}\rightarrow\text{sel}}(\Delta_{\text{death}}|\Theta'_{\text{instr}})$ have to be taken into account, since the ranges from which Δ_{diff} and Δ_{death} are sampled differ.

16.1.2. Jumping from the instructive to the selective scenario

Jumping from the instructive (one common cell death probability, two lineage-specific differentiation probabilities) to the selective scenario (one common differentiation probability, two lineage-specific cell death probabilities) involves suggesting a parameter set $\Theta'_{\text{sel}} = \{p'_{\text{diff}}, p'_{\text{death}_1}, p'_{\text{death}_2}, p'_{\text{div}_0}, p'_{\text{div}_1}, p'_{\text{div}_2}\}$ based on the current parameter set $\Theta^n_{\text{instr}} = \{p^n_{\text{diff}_1}, p^n_{\text{diff}_2}, p^n_{\text{death}}, p^n_{\text{div}_0}, p^n_{\text{div}_1}, p^n_{\text{div}_2}\}$.

Extension of the parameter sets

To ensure a common measure, the parameter sets have to be extended with the parameters Δ_{death} and Δ_{diff} as follows: $\bar{\Theta}^n_{\text{instr}} = \{p^n_{\text{diff}_1}, p^n_{\text{diff}_2} p^n_{\text{death}}, \Delta_{\text{death}}, p^n_{\text{div}_0}, p^n_{\text{div}_1}, p^n_{\text{div}_2}\}$ and $\bar{\Theta}'_{\text{sel}} = \{p'_{\text{diff}}, \Delta_{\text{diff}}, p'_{\text{death}_1}, p'_{\text{death}_2}, p'_{\text{div}_0}, p'_{\text{div}_1}, p'_{\text{div}_2}\}$.

The model jump then involves the following assignments:

$$p'_{\text{diff}} = \frac{p^n_{\text{diff}_1} + p^n_{\text{diff}_2}}{2}$$

$$\Delta_{\text{diff}} = \frac{p^n_{\text{diff}_1} - p^n_{\text{diff}_2}}{2}$$

$$p'_{\text{death}_1} = p^n_{\text{death}} + \Delta_{\text{death}}$$

$$p'_{\text{death}_2} = p^n_{\text{death}} - \Delta_{\text{death}}$$

where $\Delta_{\text{death}} \sim \mathcal{U}_{[-\frac{p^n_{\text{death}}}{2}, \frac{p^n_{\text{death}}}{2}]}$. The Jacobian for this transformation is 1 (see Appendix Section C.1 for a derivation). The division probabilities remain unchanged. The step is repeated if $p'_{\text{death}_1} + p'_{\text{div}_1} > 1$ or $p'_{\text{death}_2} + p'_{\text{div}_2} > 1$ since this proposal is invalid. However, due to small values for cell death and division probabilities this is usually not the case.

Calculation of the acceptance probability

The general reversible-jump acceptance probability is explained in the Appendix, Section C.1. Here, the model jump is accepted with probability $\min(A_{\text{instr}\rightarrow\text{sel}}, 1)$, with

$$A_{\text{instr}\rightarrow\text{sel}} = \frac{\text{sum-product}(\bar{\Theta}'_{\text{sel}})}{\text{sum-product}(\bar{\Theta}^n_{\text{instr}})} \cdot 1 \cdot \frac{\frac{1}{p'_{\text{diff}}}}{\frac{1}{p^n_{\text{death}}}} \cdot 1$$

and sum-product(Θ) being the output of the sum-product algorithm for a given parameter set Θ, i.e., the probability for this set of parameters. The model jump is unambiguous since only two different model classes are available and the probability for suggesting a model jump is the same for both scenarios, so $q(M_{\text{instr}}|M_{\text{sel}}) = q(M_{\text{sel}}|M_{\text{instr}}) = \alpha$. Yet, $q_{\text{instr}\rightarrow\text{sel}}(\Delta_{\text{death}}|\Theta^n_{\text{instr}})$ and $q_{\text{sel}\rightarrow\text{instr}}(\Delta_{\text{diff}}|\Theta'_{\text{sel}})$ have to be taken into account, since the ranges from which Δ_{death} and Δ_{diff} are sampled differ.

16.2. Sampling new parameters

New parameters are sampled according to the general Metropolis-Hastings algorithm. The set of parameters that have to be sampled depends on the current model class (see Chapter 15).

16.2.1. Selective scenario

In the selective scenario, a new set of parameters $\Theta'_{\text{sel}} = \{p'_{\text{diff}}, p'_{\text{death}_1}, p'_{\text{death}_2}, p'_{\text{div}_0}, p'_{\text{div}_1}, p'_{\text{div}_2}\}$ (including two cell death probabilities and one differentiation probability) has to be sampled based on the previous one ($\Theta^n_{\text{sel}} = \{p^n_{\text{diff}}, p^n_{\text{death}_1}, p^n_{\text{death}_2}, p^n_{\text{div}_0}, p^n_{\text{div}_1}, p^n_{\text{div}_2}\}$).

This is done individually for each cell type:

$$
\begin{aligned}
(2 \cdot p'_{\mathrm{diff}}, p'_{\mathrm{div}_0}, p'_{\mathrm{cont}_0}) &\sim \mathcal{D}(c \cdot (2 \cdot p^n_{\mathrm{diff}}, p^n_{\mathrm{div}_0}, p^n_{\mathrm{cont}_0})) \\
(p'_{\mathrm{death}_1}, p'_{\mathrm{div}_1}, p'_{\mathrm{cont}_1}) &\sim \mathcal{D}(c \cdot (p^n_{\mathrm{death}_1}, p^n_{\mathrm{div}_1}, p^n_{\mathrm{cont}_1})) \\
(p'_{\mathrm{death}_2}, p'_{\mathrm{div}_2}, p'_{\mathrm{cont}_2}) &\sim \mathcal{D}(c \cdot (p^n_{\mathrm{death}_2}, p^n_{\mathrm{div}_2}, p^n_{\mathrm{cont}_2}))
\end{aligned}
\tag{16.1}
$$

As outlined in Eq. 15.1, p_{cont} is the complementary event to the transitions that are possible for a cell type, and is thus uniquely determined and not part of the parametrization of a model. Nevertheless it has to be included in the sampling step. The proposal function (here, the Dirichlet distributions with high parameter values, $c = 500$) suggests a new set of parameters that are centered at the old ones, and add up to one for each cell type (for more details see Appendix Section C.2). The probabilities for a stem cell to differentiate into lineage 1 or lineage 2, respectively, are the same. To nevertheless maintain the proportions among the possible stem cell transitions, p_{diff} is counted twice in the proposal function. In addition, jumping from the selective to the instructive scenario has to be possible anytime. Therefore, the following conditions have to be met: $\frac{p'_{\mathrm{death}_1} + p'_{\mathrm{death}_2}}{2} + p'_{\mathrm{div}_1} \le 1$ and $\frac{p'_{\mathrm{death}_1} + p'_{\mathrm{death}_2}}{2} + p'_{\mathrm{div}_2} \le 1$. If this is not the case, the proposal step is repeated since the parameter combination is invalid. Θ'_{sel} is accepted with probability $\min(A_{\mathrm{sel}}, 1)$, with

$$
A_{\mathrm{sel}} = \frac{\text{sum-product}(\Theta'_{\mathrm{sel}}) \cdot q(\Theta^n_{\mathrm{sel}} | c \cdot \Theta'_{\mathrm{sel}})}{\text{sum-product}(\Theta^n_{\mathrm{sel}}) \cdot q(\Theta'_{\mathrm{sel}} | c \cdot \Theta^n_{\mathrm{sel}})}
$$

and sum-product(Θ) being the output of the sum-product algorithm for a given parameter set Θ, i.e., the probability for this set of parameters. $q(\Theta^n_{\mathrm{sel}} | c \cdot \Theta'_{\mathrm{sel}})$ and $q(\Theta'_{\mathrm{sel}} | c \cdot \Theta^n_{\mathrm{sel}})$ are the products of the Dirichlet density functions for the respective parameter subsets (one for each cell type, see Eq. 16.1).

16.2.2. Instructive scenario

In the instructive scenario, a new set of parameters $\Theta'_{\mathrm{instr}} = \{p'_{\mathrm{diff}_1}, p'_{\mathrm{diff}_2}, p'_{\mathrm{death}}, p'_{\mathrm{div}_0}, p'_{\mathrm{div}_1}, p'_{\mathrm{div}_2}\}$ (including two differentiation probabilities and one cell death probability) has to be sampled based on the previous one ($\Theta^n_{\mathrm{instr}} = \{p^n_{\mathrm{diff}_1}, p^n_{\mathrm{diff}_2}, p^n_{\mathrm{death}}, p^n_{\mathrm{div}_0}, p^n_{\mathrm{div}_1}, p^n_{\mathrm{div}_2}\}$). Here, four proposal functions have to be defined:

$$
\begin{aligned}
(p'_{\mathrm{diff}_1}, p'_{\mathrm{diff}_2}, p'_{\mathrm{div}_0}, p'_{\mathrm{cont}_0}) &\sim \mathcal{D}(c \cdot (p^n{}_{\mathrm{diff}_1}, p^n{}_{\mathrm{diff}_2}, p^n_{\mathrm{div}_0}, p^n_{\mathrm{cont}_0})) \\
p'_{\mathrm{death}} &\sim \mathcal{B}(c \cdot (p^n_{\mathrm{death}}, (1 - p^n_{\mathrm{death}}))) \\
p'_{\mathrm{div}_1} &\sim \mathcal{B}(c \cdot (p^n_{\mathrm{div}_1}, (1 - p^n_{\mathrm{div}_1}))) \\
p'_{\mathrm{div}_2} &\sim \mathcal{B}(c \cdot (p^n_{\mathrm{div}_2}, (1 - p^n_{\mathrm{div}_2})))
\end{aligned}
\tag{16.2}
$$

The proposal function (here, the Dirichlet distributions with high parameter values, $c = 500$) suggests a new set of parameters that are centered at the old ones, and add up to one for the

stem cell probabilities (for more details see Appendix Section C.2). To meet the condition that the cell death probabilities in lineage 1 and 2 are the same, it is not possible to use the same lineage specific proposal functions as in the selective scenario. Thus, transition specific proposal functions based on the beta distribution (a special case of the multivariate Dirichlet distribution) have been defined (see Appendix Section C.3 for details). If $p'_{\text{death}} + p'_{\text{div}_1} > 1$ or $p'_{\text{death}} + p'_{\text{div}_2} > 1$, the proposal step is repeated because this parameter combination is not included in the defined parameter space. However, small values for cell death and division probabilities in combination with narrow proposal functions generally ensure that the cell type specific transition probabilities will add up to one in spite of the different proposal functions. Θ'_{instr} is accepted with probability $\min(A_{\text{instr}}, 1)$, with

$$A_{\text{instr}} = \frac{\text{sum-product}(\Theta'_{\text{instr}}) \cdot q(\Theta^n_{\text{instr}} | c \cdot \Theta'_{\text{instr}})}{\text{sum-product}(\Theta^n_{\text{instr}}) \cdot q(\Theta'_{\text{instr}} | c \cdot \Theta^n_{\text{instr}})}$$

and sum-product(Θ) being the output of the sum-product algorithm for a given parameter set Θ, i.e., the probability for this set of parameters. $q(\Theta^n_{\text{instr}} | c \cdot \Theta'_{\text{instr}})$ and $q(\Theta'_{\text{instr}} | c \cdot \Theta^n_{\text{instr}})$ are the products of the Dirichlet and beta density functions for the respective parameter subsets (see Eq. 16.2).

17. Simulation results

We generated 50 trees for each scenario (selective and instructive) based on the model depicted in Fig. 15.1B, and predefined transition probabilities. Fig. 17.1 summarizes the results outlined in the following.

Assessment of the MCMC sampling behavior

We verified independence of initialization (data not shown). Fig. 17.1C shows a high acceptance rate (with regard to both model jumps and newly suggested parameters) and fast convergence.

Prediction quality

The reversible-jump approach is able to infer the true scenario as well as the true parameters. Applying it to datasets with increasing size reveals that a larger amount of data increases the quality of the prediction significantly (see Fig. 17.1A).
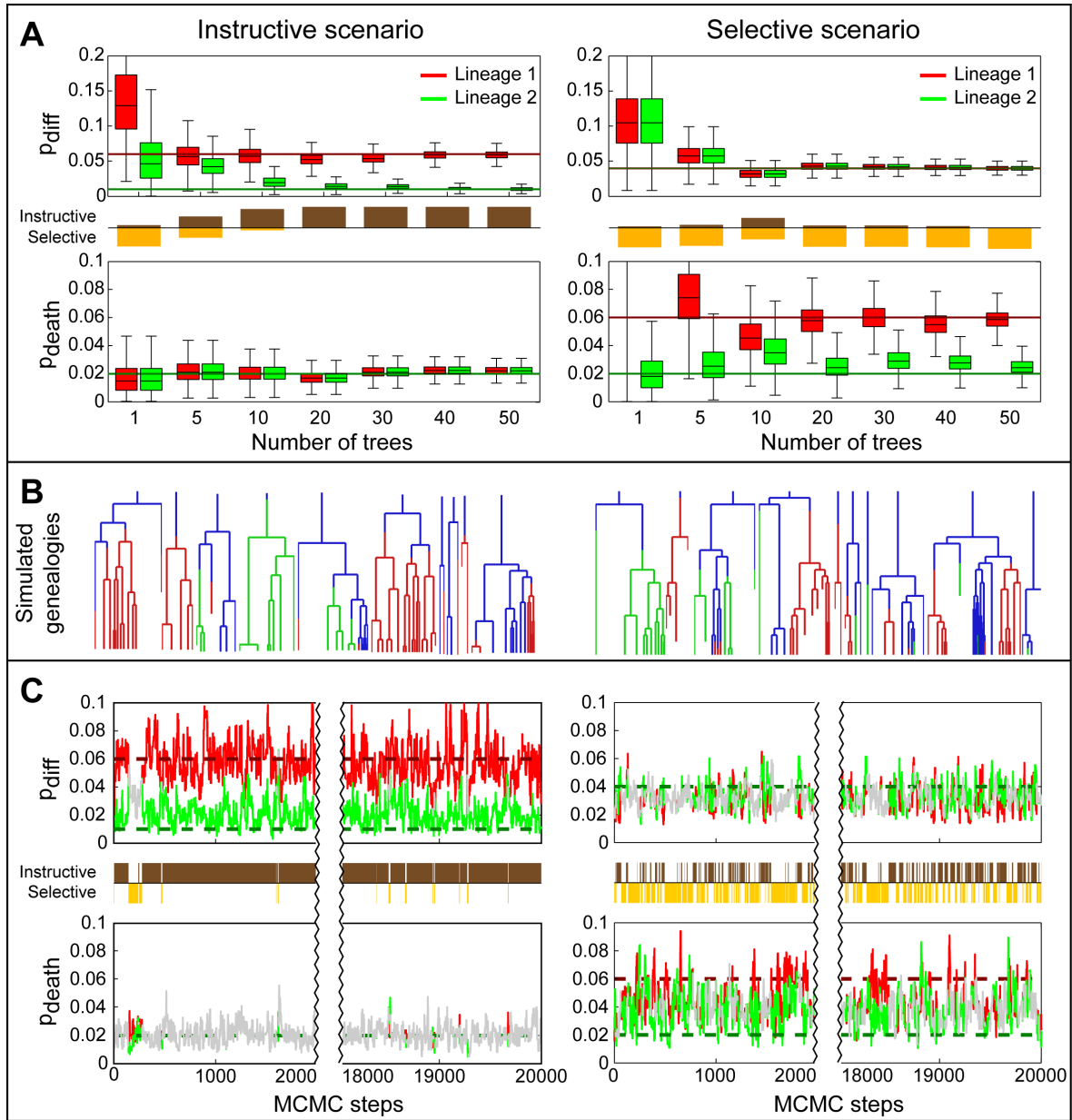
Figure 17.1.: Prediction quality and MCMC sampling behavior (simulation) for an instructive (left-hand side) and a selective (right-hand side) scenario. **(A)** Boxplots for the predicted differentiation (top) and cell death (bottom) probabilities for both lineages are shown, as well as the proportions of the predicted scenarios (middle). The boxplots include only parameters that have been sampled in the respective scenario. The horizontal green and red lines depict the simulated ("true") parameter values. On the x-axis the numbers of trees used for the predictions are depicted in increasing order. Each dataset is a subset of the next one in size. Note that in the selective scenario the predictions for the cell death probabilities based on only one tree have dropped out of the depicted range. **(B)** The simulated trees used for the prediction (here: $|Trees| = 10$) are shown. Stem cells are colored in blue, cells differentiated in lineage 1 and 2 are colored in red and green, respectively. **(C)** The traceplots (for the first and the final 2000 steps of the Markov chains) for all parameters as well as for the jumps between model classes are depicted. Grey color indicates that the corresponding parameter is the same for both lineages in the currently selected model class. The dashed green and red lines depict the simulated ("true") parameter values.

Part V.

Conclusion

The inherent complexity of systems biology models requires advanced methods for the identification of their parameters. While point estimates may be arbitrary, instable and not reproducible, Markov Chain Monte Carlo methods embed estimates in confidence intervals. These provide valuable information on the uniqueness and the variability of the parameter estimates, making the results more reliable and meaningful. They enable comparisons between different experiments as has been demonstrated in Part III (RNA degradation by the archaeal exosome). Without the information on the parameter distributions, statements such as "Short RNAs are degraded slower than long RNAs" or "The Tyr70$^{\mathrm{Rrp42}}$ mutation has a negative influence on the degradation of short RNAs" would not be possible (or at least not credible).

As for the success of a parameter estimation method, the number of parameters turned out to be not necessarily the decisive factor. What seemed to be more important was the appropriateness of the model and the quality of the data. For example, the estimation of the 28 parameters in the RNA degradation model (Part III) was highly successful since the model describes the biochemistry well and the available data is of high quality. In contrast, estimating the 7 parameters in the stem cell factor graph model in Part IV is more difficult. It is thus essential to carefully carry out the modeling process and to figure out which parameters are the ones of interest and which might lead to overparametrization. It is important to take into account the quality of the data as well as the amount and kind of data that is required to describe the parameters of interest. This can be achieved by an appropriate error model and, where necessary, a reparametrization of the model. In particular, it is important to avoid redundancy as shown in Part III (RNA degradation by the exosome). There, it was impossible to distinguish between the accumulation of bound or of free RNAs, i.e., between fast association and slow cleavage or slow association and fast cleavage. Constructing smooth probability landscapes for the parameters simplifies the estimation process, however, this is easier for continuous parameter spaces (e.g., Part III) than for discrete ones. The rugged probability landscape of the Nested Effects Models in Part II, for example, necessitated the incorporation of an Expectation Maximization algorithm into the MCMC sampling scheme for the exploration of the model space. To assess the usefulness and identifiability of parameters when the model class is complex, extensive simulations are indispensable.

An important criterion for the success and reliability of a MCMC sampling method is the convergence of the Markov chain. Again, obtaining good mixing and fast converging chains is often easier in continuous than in discrete parameter spaces as can be seen by comparing Part III (RNA degradation by the exosome) with Part II (yeast Mediator and NEMs). It is thus essential to define the parameter neighborhood and construct the transition kernel (i.e., the proposal function) carefully, and to take the time to verify MCMC convergence by extensive simulations.

It has also turned out to be important to effectively narrow down the space of suitable parameters, for example by choosing a suitable prior. While this was of great value in Part II and Part III, it turned out to be the key breakthrough in Part IV (the differentiation model

for hematopoietic stem cells). The intuitive model consisting of two differentiation and two cell death probabilities was not identifiable, but splitting it into two simpler model classes where either the differentiation probabilities (selective scenario) or the cell death probabilities (instructive scenario) are *a priori* defined to be equal, turned out to be successful.

In conclusion, I have demonstrated in this thesis that Markov Chain Monte Carlo sampling methods, if implemented carefully, can add great value to the interpretation of experimental data.

Part VI.

Appendix

A. Supplementary material for Part II - MC EMiNEM and yeast Mediator

A.1. EM algorithm

This section is an extended version of the EM Section in the main text. We report the results that arise from the EM algorithm when applied to our situation. The calculations involve only elementary algebra but are sometimes tedious. According to [102], the likelihood function ("structure likelihood") of the signals graph in a NEM is

$$L(\Theta, H) = P(D|\Theta, H) \quad = \quad \prod_{j \in \mathcal{S}} \prod_{k \in \mathcal{E}} P(D_{jk}|(\Theta H)_{jk}) = \prod_{j,k} p_{jk} \tag{A.1}$$

The log likelihood can be written as

$$
\begin{aligned}
\log L(\Theta, H) \quad &= \quad \log P(D|H, \Theta) = \prod_{j,k} p_{jk} \\
&= \quad \log \prod_{j,k} [P(D_{jk}|(\Theta H)_{jk})/q_{jk}] + \log \prod_{j,k} q_{jk} \\
&= \quad \sum_{j,k} \log \begin{cases} p_{jk}/q_{jk} & \text{if } (\Theta H)_{jk} > 0 \\ 1 & \text{if } (\Theta H)_{jk} = 0 \end{cases} + \text{const} \\
&= \quad \sum_{j,k} \begin{cases} R_{jk} & \text{if } (\Theta H)_{jk} > 0 \\ 0 & \text{if } (\Theta H)_{jk} = 0 \end{cases} + \text{const} \\
&= \quad \sum_{j,k} (\Theta H)_{jk} R_{jk} + \text{const} \\
&= \quad \sum_{j} \left[\sum_{k} (\Theta H)_{jk} (R^T)_{kj} \right] + \text{const} \\
&= \quad \sum_{j} (\Theta H R^T)_{jj} + \text{const} \tag{A.2} \\
&= \quad \text{trace}(\Theta H R^T) + \text{const} \tag{A.3}
\end{aligned}
$$

The (full) posterior is then given by

$$\log P(\Theta, H|D) = \log P(D|H, \Theta) + \log \pi(\Theta, H) + \text{const}$$

We assume edge-wise independent priors, $\pi(\Theta, H) = \pi^{\mathcal{S}}(\Theta) \cdot \pi^{\mathcal{E}}(H)$, and $\pi^{\mathcal{S}}(\Theta) = \prod_{i,j} \pi^{\mathcal{S}}(\Theta_{ij})$, $\pi^{\mathcal{E}}(H_{\bullet k}) = \prod_k \pi^{\mathcal{E}}(H_{\bullet k})$.

A.1.1. The general EM algorithm

Throughout this section, the data D, respectively the matrix R, is considered given and fixed. We want to find the maximum *a posteriori* estimate $\hat{\Theta}$ for the signals graph,

$$\hat{\Theta} = \arg\max_{\Theta} P(\Theta|D) = \arg\max_{\Theta} \sum_{H \in \mathcal{M}_{\mathcal{E}}} P(\Theta, H|D) \tag{A.4}$$

The Expectation-Maximization algorithm was developed exactly for this purpose, to perform a maximization task in the presence of hidden variables [19]. The EM proceeds by iteratively constructing a sequence of parameter estimates Θ^t, $t = 1, 2, \dots$ such that the sequence $(P(\Theta^t|D))_{t=1,2,\dots}$ is monotonically increasing, and converges (under mild additional assumptions that are met in our case) to a local maximum of $P(\Theta|D)$.

The expectation (E-)step of the EM algorithm involves calculating the expectation value $Q(\Theta; \Theta^t)$,

$$Q(\Theta; \Theta^t) = \mathbb{E}_{P(H|D,\Theta^t)}\left[\log P(D, H|\Theta)\right] = \sum_{H \in \mathcal{M}_{\mathcal{E}}} \log P(D, H|\Theta) \cdot P(H|D, \Theta^t) . \tag{A.5}$$

The maximization (M-)step of the EM algorithm then consists of finding

$$\Theta^{t+1} = \arg\max_{\Theta} \left[Q(\Theta; \Theta^t) + \log \pi^{\mathcal{S}}(\Theta) \right] , \tag{A.6}$$

which is usually a much easier task than solving (A.4) directly. In the following, both steps are described in detail.

A.1.2. The E-step

Let us assume that the priors for Θ and H are independent, $\pi(\Theta, H) = \pi^{\mathcal{S}}(\Theta)\pi^{\mathcal{E}}(H)$. Then, the terms in $Q(\Theta; \Theta^t)$ can be rearranged

$$
\begin{aligned}
Q(\Theta; \Theta^t) &= \mathbb{E}_{P(H|D,\Theta^t)}\left[\log P(D, H|\Theta)\right] \tag{A.7}\\
&= \sum_H P(H|D, \Theta^t) \log P(D, H|\Theta)\\
&= \sum_H \frac{P(D|H, \Theta^t)P(H|\Theta^t)}{P(D|\Theta^t)} \log(P(D|H, \Theta)P(H|\Theta))\\
&\overset{\pi(H,\Theta)=\pi^{\mathcal{E}}(H)\pi^{\mathcal{S}}(\Theta)}{=} \frac{1}{P(D|\Theta^t)} \sum_H P(D|H, \Theta^t)\pi^{\mathcal{E}}(H)[\log P(D|H, \Theta) + \log \pi^{\mathcal{E}}(H)]\\
&= c^{-1} \sum_H P(D|H, \Theta^t)\pi^{\mathcal{E}}(H) \log P(D|H, \Theta) + \text{const}
\end{aligned}
$$

with a normalizing factor $c = P(D|\Theta^t) = \sum_H P(D|H, \Theta^t)\pi^{\mathcal{E}}(H)$ and a constant that does not depend on Θ. The problem of maximizing $Q(\Theta; \Theta^t)$ is therefore equivalent to maximizing $\tilde{Q}(\Theta; \Theta^t)$, where

$$\tilde{Q}(\Theta; \Theta^t) = c^{-1} \sum_H P(D|H, \Theta^t)\pi^{\mathcal{E}}(H) \log P(D|H, \Theta) \tag{A.8}$$

We seek for an expression for (A.6) which is amenable to analytic maximization strategies. Let $V = \mathbb{R}^{\mathcal{E}}$ be an m-dimensional vector space, which is spanned by the unit column vectors $e_k \in V$, $k \in \mathcal{E}$, and let $e_0 = 0 \in V$. We assume further that the prior for H factorizes into priors for each effect,

$$\pi^{\mathcal{E}}(H) = \prod_{k \in \mathcal{E}} \pi_k^{\mathcal{E}}(He_k) . \tag{A.9}$$

Let d_j be the j-th unit column vector of dimension n, and d_0 the n-dimensional null vector. The NEM model assumes that each effect assigns to at most one signal, so $\pi_k^{\mathcal{E}}(v) = 0$ for each vector $v \notin \{d_j, j = 0, ..., n\}$, $k \in \mathcal{E}$, and

$$\pi_k^{\mathcal{E}}(d_j) = \pi_{jk} , \ j = 0, 1, ..., n, \text{ and } \sum_{j=0}^{n} \pi_{jk}^{\mathcal{E}} = 1 . \tag{A.10}$$

The $m \times m$ unit matrix is denoted by E. Be aware of the identity $E = \sum_{k \in \mathcal{E}} e_k e_k^T$. We take advantage of the fact that the trace of a quadratic matrix is a linear function, and that $tr(AB) = tr(BA)$ for arbitrary (compatible) matrices A, B.

$$\begin{aligned} tr(\Theta H R^T) &= tr(R^T \Theta H) = tr(\sum_{k \in \mathcal{E}} e_k e_k^T R^T \Theta H) \tag{A.11} \\ &= \sum_{k \in \mathcal{E}} tr(e_k e_k^T R^T \Theta H) = \sum_{k \in \mathcal{E}} e_k^T R^T \Theta(He_k) \end{aligned}$$

Thus by (A.3), letting $g_k(v, \Theta) = e_k^T R^T \Theta v$,

$$\log P(D|H, \Theta) = \sum_{k \in \mathcal{E}} g_k(He_k, \Theta) + \text{const.} \tag{A.12}$$

Analogously,

$$P(D|H, \Theta^t) \propto \exp(tr(\Theta^t H R^T)) = \prod_{k \in \mathcal{E}} f_k(He_k, \Theta^t) , \tag{A.13}$$

with $f_k(v, \Theta^t) = \exp(g_k(v, \Theta^t))$. For convenience we suppress the dependence of g_k on Θ (and write $g_k(v)$ instead of $g_k(v, \Theta)$) and of f_k on Θ^t (and write $f_k(v)$ instead of $f_k(v, \Theta^t)$). Let $W = \{0, 1\}^n$. The evaluation of $\tilde{Q}(\Theta; \Theta^t)$ can be simplified considerably. For $r = 1, ..., m$, let

$$F_r(\Theta) = \sum_{v_r \in W} \sum_{v_{r+1} \in W} ... \sum_{v_m \in W} \left(\prod_{l \geq r}^{m} \pi_l^{\mathcal{E}}(v_l) f_l(v_l) \right) \cdot \left(\sum_{k \geq r}^{m} g_k(v_k) \right) \tag{A.14}$$

Note that

$$\tilde{Q}(\Theta;\Theta^t) \quad = \quad c^{-1} \sum_H P(D|H,\Theta^t)\pi^{\mathcal{E}}(H) \log P(D|H,\Theta) \tag{A.15}$$

$$\stackrel{(A.12,A.13)}{=} \quad c^{-1} \sum_H \left(\prod_{l=0}^m \pi_l^{\mathcal{E}}(He_l)f_l(He_l) \right) \cdot \left(\sum_{k=0}^m g_k(He_k) \right)$$

$$= \quad c^{-1} F_1(\Theta)$$

We introduce two more terms,

$$A_k \quad = \quad \sum_{v \in W} \pi_k^{\mathcal{E}}(v)f_k(v) = \sum_{j=0}^n \pi_{jk}^{\mathcal{E}} f_k(d_j)) \tag{A.16}$$

$$B_k(\Theta) \quad = \quad \sum_{v \in W} \pi_k^{\mathcal{E}}(v)f_k(v)g_k(v) = \sum_{j=0}^n \pi_{jk}^{\mathcal{E}} f_k(d_j)g_k(d_j) \ . \tag{A.17}$$

F_r can be calculated from F_{r+1} via the recursive formula (A.18):

$$F_r(\Theta) \quad = \quad \sum_{v_{r+1},\dots,v_m \in W} \sum_{v_r \in W} \left(\pi_r^{\mathcal{E}}(v_r)f_r(v_r) \cdot \prod_{l>r}^m \pi_l^{\mathcal{E}}(v_l)f_l(v_l) \right) \cdot \left(\sum_{k \geq r}^m g_k(v_k) \right) \tag{A.18}$$

$$= \quad \sum_{v_{r+1},\dots,v_m \in W} \left(\prod_{l>r}^m \pi_l^{\mathcal{E}}(v_l)f_l(v_l) \right) \cdot \sum_{v_r \in W} \pi_r^{\mathcal{E}}(v_r)f_r(v_r) \cdot \left(\sum_{k \geq r}^m g_k(v_k) \right)$$

$$= \quad \sum_{v_{r+1},\dots,v_m \in W} \left(\prod_{l>r}^m \pi_l^{\mathcal{E}}(v_l)f_l(v_l) \right) \cdot \sum_{v_r \in W} \pi_r^{\mathcal{E}}(v_r)f_r(v_r) \cdot \left(g_r(v_r) + \sum_{k>r}^m g_k(v_k) \right)$$

$$= \quad \sum_{v_{r+1},\dots,v_m \in W} \left(\prod_{l>r}^m \pi_l^{\mathcal{E}}(v_l)f_l(v_l) \right) \cdot \left(B_r(\Theta) + A_r \sum_{k>r}^m g_k(v_k) \right)$$

$$= \quad B_r(\Theta) \left(\prod_{l>r}^m \sum_{v_l \in W} \pi_l^{\mathcal{E}}(v_l)f_l(v_l) \right) + \sum_{v_{r+1},\dots,v_m \in W} \left(\prod_{l \geq r+1}^m \pi_l^{\mathcal{E}}(v_l)f_l(v_l) \right) \left(A_r \sum_{k \geq r+1}^m g_k(v_k) \right)$$

$$= \quad B_r(\Theta) \prod_{l>r}^m A_l \ + \ A_r \cdot F_{r+1}(\Theta)$$

By reverse induction we prove the formula

$$F_r = \left(\prod_{l \geq r} A_l \right) \left(\sum_{k \geq r} \frac{B_k(\Theta)}{A_k} \right) \ , \tag{A.19}$$

the initial case $r = m$ is $F_m(\Theta) = \sum_{v \in W} \pi_k^{\mathcal{E}}(v)f_k(v)g_k(v) = B_m(\Theta) = A_m \cdot \frac{B_m(\Theta)}{A_m}$. The

induction step is completed by

$$F_r \overset{(A.18)}{=} B_r(\Theta) \prod_{l>r}^{m} A_l + A_r \cdot F_{r+1}(\Theta) \tag{A.20}$$

$$= B_r(\Theta) \prod_{l>r}^{m} A_l + A_r \left(\prod_{l>r} A_l \right) \left(\sum_{k>r} \frac{B_k(\Theta)}{A_k} \right)$$

$$= \frac{B_r(\Theta)}{A_r} \prod_{l\geq r}^{m} A_l + \left(\prod_{l\geq r} A_l \right) \left(\sum_{k>r} \frac{B_k(\Theta)}{A_k} \right)$$

$$= \left(\prod_{l\geq r} A_l \right) \left(\sum_{k\geq r} \frac{B_k(\Theta)}{A_k} \right)$$

We realize that

$$c = \sum_H P(D|H, \Theta^t) \pi^{\mathcal{E}}(H) \overset{(A.13),\, \pi^{\mathcal{E}}(H)=\prod_{k\in\mathcal{E}} \pi_k^{\mathcal{E}}(He_k)}{=} \sum_{v_1,...,v_m \in W} \left(\prod_{k=1}^{m} \pi_k^{\mathcal{E}}(v_k) f_k(v_k) \right)$$

$$= \prod_{k=1}^{m} \sum_{v_k \in W} \pi_k^{\mathcal{E}}(v_k) f_k(v_k) = \prod_{k=1}^{m} A_k \tag{A.21}$$

(note that $g_k(d_j, \Theta) = e_k^T R^T \Theta d_j = (R^T \Theta)_{kj}$). Note that for a deterministic prior, fixing an effects gene assignment $H \in \{0,1\}^{n \times m}$,

$$\log c = \sum_{k=1}^{m} \log A_k$$

$$= \sum_{k=1}^{m} \log \left(\sum_{j=0}^{n} \pi_{jk}^{\mathcal{E}} f_k(d_j) \right) = \sum_{k=1}^{m} \log \left(\sum_{j=0}^{n} \pi_{jk}^{\mathcal{E}} \exp g_k(d_j, \Theta^t) \right)$$

$$= \sum_{k=1}^{m} \log \left(\sum_{j=0}^{n} \pi_{jk}^{\mathcal{E}} \exp (R^T \Theta^t)_{kj} \right)$$

$$= \sum_{k=1}^{m} \log \left(H_{\bullet k} \exp (R^T \Theta^t)_{k\bullet} \right)$$

$$\overset{H\in\{0,1\}^{n \times m}}{=} \sum_{k=1}^{m} H_{\bullet k} (R^T \Theta^t)_{k\bullet}$$

$$= tr \left(H R^T \theta^t \right) \tag{A.22}$$

Finally, we obtain

$$
\tilde{Q}(\Theta;\Theta^t) \stackrel{(A.15)}{=} c^{-1} \cdot F(\Theta) \tag{A.23}
$$
$$
\stackrel{(A.20),(A.21)}{=} \left(\prod_{k=1}^m A_k\right)^{-1} \cdot \left(\prod_{l=1}^m A_l\right)\left(\sum_{k=1}^m \frac{B_k(\Theta)}{A_k}\right)
$$
$$
= \sum_{k=1}^m \frac{B_k(\Theta)}{A_k}
$$

A.1.3. The M-step

According to (7.2) and (A.23) we have to maximize

$$
\tilde{Q}(\Theta;\Theta^t) + \log \pi^S(\Theta) = \sum_{k\in\mathcal{E}} \frac{B_k(\Theta)}{A_k} + \log \pi^S(\Theta) \ . \tag{A.24}
$$

As a further simplification we assume edgewise independent priors:

$$
\pi^S(\Theta) = \prod_{i,j}(\pi_{ij}^S)^{\Theta_{ij}}(1 - \pi_{ij}^S)^{1-\Theta_{ij}} \ , \ \text{with } 0 \le \pi_{ij}^S \le 1 \tag{A.25}
$$

(we may disregard the cases in which $\pi_{ij}^S \in \{0,1\}$, because in this case the corresponding edge Θ_{ij} is fixed as absent or present and is therefore not subject to optimization). The log of the prior is then a linear function in each Θ_{ab}:

$$
\log \pi^S(\Theta) = \log \prod_{i,j} \pi_{ij}^{\Theta_{ij}}(1 - \pi_{ij})^{1-\Theta_{ij}} \tag{A.26}
$$
$$
= \sum_{i,j}[\,\Theta_{ij}\log\pi_{ij} + (1-\Theta_{ij})\log(1-\pi_{ij})\,]
$$
$$
= \sum_{i,j}[\,\Theta_{ij}(\log\pi_{ij} - \log(1-\pi_{ij}) + \log(1-\pi_{ij})\,]
$$
$$
= \sum_{i,j}\Theta_{ij}\log\frac{\pi_{ij}}{1-\pi_{ij}} + \text{const} =: \sum_{i,j}\Theta_{ij}\tau_{ij} + \text{const}
$$

This implies that the objective function (A.24) $\tilde{Q}(\Theta;\Theta^t) + \log \pi^S(\Theta)$ is a polynomial in the variables $\{\Theta_{ab}|a = 1,...,n; b = 1,...,n\}$ of total degree 1. The partial derivatives of the objective function with respect to Θ_{ab} are therefore constant, i.e., independent of Θ (Note that Θd_j equals the j-th column of Θ, so $g_k(d_j,\Theta) = e_k^T R^T \Theta d_j$ is linear in the entries of Θ):

$$
\frac{\partial g_k(d_j,\Theta)}{\partial \Theta_{ab}} = \frac{\partial}{\partial \Theta_{ab}}[(e_k^T R^T)(\Theta d_j)] = \frac{\partial}{\partial \Theta_{ab}}\sum_{i=1}^n R_{ik}\Theta_{ij} \tag{A.27}
$$
$$
= \sum_{i=1}^n R_{ik}\frac{\partial}{\partial \Theta_{ab}}\Theta_{ij} = \delta_{j=b}R_{ak} \ .
$$

Hence

$$
\begin{aligned}
\frac{\partial B_k(\Theta)}{\partial \Theta_{ab}} &= \sum_{i=0}^{n} \pi_{ik}^{\mathcal{E}} f_k(d_i, \Theta^t) \frac{\partial}{\partial \Theta_{ab}} g_k(d_s, \Theta) \overset{(A.27)}{=} \sum_{i=0}^{n} \pi_{ik}^{\mathcal{E}} f_k(d_i, \Theta^t) \delta_{i=b} R_{ak} \quad \text{(A.28)} \\
&= \pi_{bk}^{\mathcal{E}} f_k(d_b, \Theta^t) R_{ak}
\end{aligned}
$$

Consequently,

$$
\frac{\partial \tilde{Q}(\Theta; \Theta^t)}{\partial \Theta_{ab}} = \frac{\partial}{\partial \Theta_{ab}} \sum_{k=1}^{m} \frac{B_k(\Theta)}{A_k} = \sum_{k=1}^{m} \frac{\pi_{bk}^{\mathcal{E}} f_k(d_b, \Theta^t) R_{ak}}{A_k} , \quad \text{(A.29)}
$$

which together with (A.26) implies

$$
\frac{\partial(\tilde{Q}(\Theta; \Theta^t) + \log \pi^S(\Theta))}{\partial \Theta_{ab}} = \sum_{k=1}^{m} \pi_{bk}^{\mathcal{E}} R_{ak} \exp(g_k(d_b, \Theta^t))(A_k)^{-1} + \tau_{ab} , \quad \text{(A.30)}
$$

Using the step function $\text{step}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$, the updated values in Θ^{t+1} can be stated in closed form:

$$
\Theta_{ab}^{t+1} = \text{step} \left\{ \sum_{k=1}^{m} R_{ak} \pi_{bk}^{\mathcal{E}} \exp((R^T \Theta^t)_{kb})(A_k)^{-1} + \tau_{ab} \right\} . \quad \text{(A.31)}
$$

In the general case of an arbitrary prior $\pi^S(\Theta)$, it can be difficult to find a global optimum of the objective function $\tilde{Q}(\Theta; \Theta^t) + \log \pi^S(\Theta)$. However it is not necessary to find a global optimum, it is sufficient to find a Θ^{t+1} that increases the value of the objective function over the current value $\tilde{Q}(\Theta; \Theta^t) + \log \pi^S(\Theta^t)$. It has been shown in [72] that such a "stepwise" EM still converges to a local maximum of $P(\Theta \mid D)$. Therefore, we start with $\Theta = \Theta^t$ and go through all edges Θ_{ab} in a random order and check whether alteration of Θ_{ab} improves the objective function. If yes, we perform this change in Θ and continue until all edges were checked. The resulting Θ is our new Θ^{t+1}.

A.2. MCMC sampling

A.2.1. A theoretical motivation for the sampling of local maxima

EMiNEM is viewed as a function $EM : \Theta \mapsto \hat{\Theta} = EM(\Theta)$, which maps the signals graph space $\mathcal{M}_S$ onto the space $\mathcal{N} = EM(\mathcal{M}_S)$ of local maxima of the posterior. The current section is devoted to constructing a sequence in $\mathcal{N}$ that provides a representative sample of $P|_{\mathcal{N}}$, the restriction of the posterior probability P to $\mathcal{N}$. Our task is complicated by the fact that we cannot construct functions that sample from $\mathcal{N}$ directly, because the calculation of each member requires the application of EMiNEM. Instead, we use Metropolis-Hastings Markov Chain Monte Carlo (MCMC) sampling (see Section 2.4 for a general introduction)

to construct a sequence in $\mathcal{M}_\mathcal{S}$, and lift it to $\mathcal{N}$. Let $(\Theta_i)_{i=1,2,\ldots}$ be a sequence of signals graphs in $\mathcal{M}_\mathcal{S}$ obtained by MCMC sampling from the distribution P. The corresponding sequence $(EM(\Theta_i))_{i=1,2,\ldots}$ is then an approximate empirical sample from the distribution $\hat{P}(\hat{\Theta}) = \sum\{P(\Theta|D); EM(\Theta) = \hat{\Theta}\} \approx P|_\mathcal{N}(\hat{\Theta})$ on $\mathcal{N}$. This approximation is valid under the assumption that the probability of $P(\hat{\Theta} \mid D)$ is substantially larger than $P(\Theta \mid D)$ for all other $\Theta \in EM^{-1}(\hat{\Theta})$, which is presumably the case. However, the convergence speed of this Markov chain is very slow, the reason being implicit in the assumption: In order to move from one local maximum to a different one, the underlying Markov chain in $\mathcal{M}_\mathcal{S}$ needs to traverse regions of substantially lower probability. We remove this obstacle by sampling $(\Theta_i)_{i=1,2,\ldots}$ from the distribution $Q(\Theta) \propto P(EM(\Theta)|D)$ instead of sampling from P. The corresponding sequence $(EM(\Theta_i))_{i=1,2,\ldots}$ is then an approximate empirical sample from the distribution

$$\hat{P}(\hat{\Theta}) \propto \sum\{P(EM(\Theta)|D); EM(\Theta) = \hat{\Theta}\} = P(\hat{\Theta}|D) \cdot |\{\Theta; EM(\Theta) = \hat{\Theta}\}|$$
$$\approx P(\hat{\Theta} \mid D) \cdot c \tag{A.32}$$

The last approximation assumes that the pre-image of $\hat{\Theta}$ under EM has a similar size c for all $\hat{\Theta} \in \mathcal{N}$. In any case, we expect the relative probability $\frac{\hat{P}(\hat{\Theta}_1)}{\hat{P}(\hat{\Theta}_2)}$ to be dominated by the quotient $\frac{P(\hat{\Theta}_1|D)}{P(\hat{\Theta}_2|D)}$, which justifies our approximation in Eq. A.32 for the purpose of finding high-scoring graphs $\hat{\Theta}$.

A.2.2. Empirical Bayes estimation of the effects graph prior

The attachment probability H_{jk}^i of effect node k to signal node j, based on one signals graph Θ^i, is:

$$H_{jk}^i = P(H_{\bullet k}^i = e_j|\Theta^i, R, H^{old}) = \frac{\exp f_k^i(j)}{\sum_j \exp f_k^i(j)} \ , \ \text{with}$$

$$f_k^i(j) = \begin{cases} \log \pi(H_{jk}^{old}) + R_{\bullet k}\Theta_{\bullet j}^i & \text{for } j \epsilon S \\ \log \pi(H_{jk}^{old}) & \text{for } j \text{ the null node} \end{cases}$$

The new attachment probability, based on the preceding $N = \frac{|chain|}{12}$ steps of the Markov chain, is then $H^{new} = \frac{\sum_{i=1}^N H^i}{N}$.

In our approach, we do not sample from $\mathcal{M}_\mathcal{S}$ directly, but we sample from a set of local maxima $\mathcal{N}$. This set is much smaller and develops slower than $\mathcal{M}_\mathcal{S}$, as can be seen in the traceplots. Note that this set changes every epoch, since the prior is updated empirically.
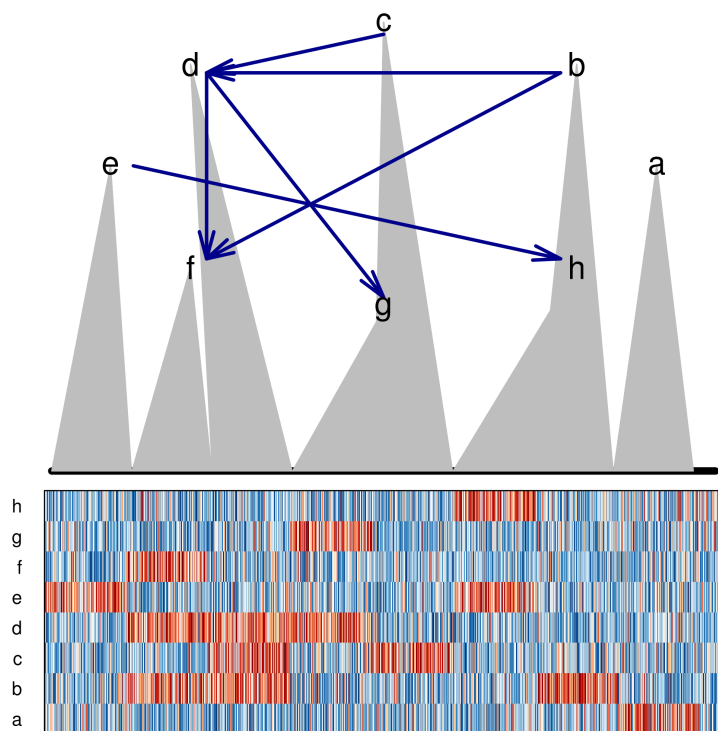
Figure A.1: A simulated NEM. Above, the signals graph is shown, below the corresponding R matrix, clustered according to the gene attachment (rows: perturbations, columns: effects on measured genes). Red color indicates a positive log-ratio value, blue color indicates a negative log-ratio value. The stronger the color of a field R_{kj}, $k \in \mathcal{E}$, $j \in \mathcal{S}$, the higher the probability that the measured data is due to the fact that there actually is an effect of signal j on gene k, or, that there is no effect, respectively.
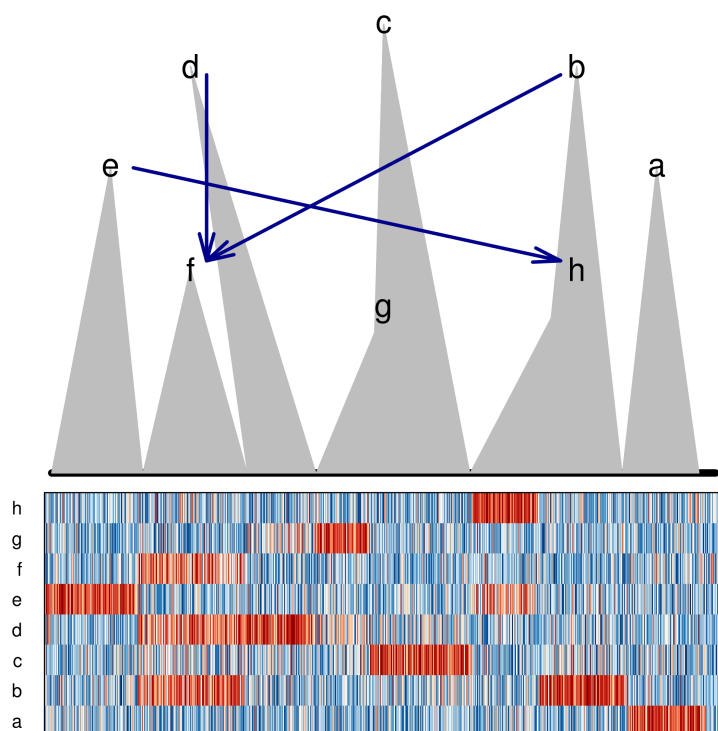
Figure A.2: Prediction for Fig. A.1. Above, the signals graph is shown, below the corresponding R matrix, clustered according to the gene attachment (rows: perturbations, columns: effects on measured genes). Red color indicates a positive log-ratio value, blue color indicates a negative log-ratio value. The stronger the color of a field R_{kj}, $k \in \mathcal{E}$, $j \in \mathcal{S}$, the higher the probability that the measured data is due to the fact that there actually is an effect of signal j on gene k, or, that there is no effect, respectively. The matrix here is the same as in Fig. A.1, but the ordering of genes (columns) is different, since it depends on the gene attachment derived by the MCMC sampling.

A.3. Simulation

A.3.1. Data generation

Simulated datasets have been generated using the method createNEM, provided by Nessy [101]. It takes as input the number of signals $|\mathcal{S}|_{\text{true}}$ and effect genes $|\mathcal{E}|_{\text{true}}$, as well as the two noise parameters μ and δ. Θ_{true} and H_{true} are randomly sampled according to $|\mathcal{S}|_{\text{true}}$, $|\mathcal{E}|_{\text{true}}$ and a predefined edge frequency for the signals graph $0.7 \cdot |\mathcal{S}|$ that corresponds to the expected number of edges in the observed data. The simulated ("true") data matrix ($|\mathcal{S}|_{\text{true}} \times |\mathcal{E}|_{\text{true}}$) is calculated according to these graphs. A noisy log-odds ratio matrix is then calculated based on the "true" effects by sampling its values from two normal distributions with ($mean = -\frac{\mu}{2}$, $sd = \delta$) and ($mean = \frac{\mu}{2}$, $sd = \delta$), respectively. μ and δ have been chosen such that for an optimal test with a type-I error ($\alpha - \text{level}$) of 5%, a type II error ($\beta - \text{level}$) of $0.04\%, 20\%, 49\%$, and 66% would be achieved, respectively. A simulated NEM and the corresponding prediction of MC EMiNEM, for $|\mathcal{S}| = 8$ and $\beta - \text{level} = 49\%$ are shown in Fig. A.1 and Fig. A.2.

A.3.2. Prediction quality

To assess the prediction quality, MC EMiNEM has been compared to four other methods. In the following, the results of this comparison (as depicted in Fig. 8.5A) are discussed and a detailed explanation of the four methods is provided. In all cases, a uniform prior and the data-driven prior have been chosen for the signals graph and the effects graph, respectively, to ensure a fair basis for comparison. Randomly sampled sparse graphs with $p_{\text{edge}} = \frac{1}{|S|}$ have been used for initialization.

Random

For each NEM, 5000 random signals graphs have been sampled, according to the parameters described above. Every (unique) graph has than been weighted by its posterior and a consensus signals graph has been built including all edges with a (weighted) value of ≥ 0.5. This is the most trivial method for parameter estimation.
As expected, this method yields quit good results for small numbers of signal nodes, where the probability of randomly drawing reasonable graphs is higher. However, for larger number of signal nodes, independent of the noise level, this method is not able to detect the correct edges at all.

EMiNEM

This method is based on the random sampling approach, except that not the sampled signals graphs but their corresponding local maxima have been weighted and combined to a consensus signals graph.
EMiNEM is slightly better than random sampling, because by only taking into account

local maxima unlikely graphs are excluded from the consensus. However, it still relies on random drawing of signals graphs and only yields good results for small numbers of nodes. By comparing it to the considerably better results of the more elaborate MC EMiNEM it is clearly visible that the more complex and time-consuming Markov Chain Monte Carlo approach, which leads to a reasonably "guided" exploration of the model space, is justified.

Nessy

Nessy is a publicly available NEM implementation [102]. Unlike (MC) EMiNEM it's a maximum full likelihood / posteriori approach, where not only the maximum for the signals graph but also for the effects graph should be identified. Since no prior knowledge regarding the signals graph is available, but a sparse graph is assumed, Nessy is initialized with the empty graph.
MC EMiNEM is a maximum marginal posteriori approach, it only calculates the maximum for the signals graph and marginalizes over the effects graph. For good data, with low amount of noise, the effects graph is clearly identifiable and MC EMiNEM and Nessy perform comparably. However, for higher noise the calculation of the maximum effects graph is error-prone and the risk of getting stuck in the wrong model is high, so Nessy is clearly outperformed by MC EMiNEM there.

nem

nem is the original NEM implementation, publicly available through Bioconductor [67]. Recently, [27] published a review of all currently available NEM algorithms, where they recommend the Bayesian greedy hill climbing approach for small networks as the method of choice. It calculates the original NEM score by integrating over all effects graphs. According to these findings, we applied nem on the log-odds ratios with the following parameters: inference="nem.greedy"and type="CONTmLLBayes".

A.4. Application: The yeast Mediator signaling network

A.4.1. Data processing

Data processing has been done using R [79]. The arrays were read in and transformed to expression values one by one, using expresso() from the **R**/*Bioconductor* package affy [29] with the following parameters for background correction and summarization: bgcorrect.method="rma", pmcorrect.method="pmonly", summary.method="avgdiff". Some arrays included *S.pombe* probes, they were filtered to *S.cerevisiae*. The median expression values were centered to zero (on the log-scale) for each array (this step has only the purpose of generating a sensible average expression distribution in the subsequent quantile normalization step). The expression values were log2 transformed and quantile normalization was performed afterwards using quantile.normalization() from the affy-package.

The **R**/*Bioconductor* package limma [92] was used for further the assessment of differential gene expression. A design matrix was constructed that takes into account batch-specific effects as well as subunit-specific effects. The linear regression model was fitted using lmFit(). Finally, the log-odds ratios corresponding to subunit specific effects were extracted using ebayes().

To accommodate the different experiments that have been combined, contrasts with respect to batch effects have been created and fitted. Genes showing differential expression (here with a fold change ≥ 2.5) with respect to these contrasts were removed from the subsequent analysis. Additionally, genes that do not react to any perturbation (here with a log-odds ratio < 0 in all cases) were removed.

A.4.2. Comparison with cluster analysis

Recently, "structure-function" analyses have been suggested and conducted [56, 104]. In a clustering approach, they use expression profile similarity as a proxy for physical interaction, respectively for common module membership. Their method was strikingly successful in identifying physical interactions between Mediator subunits. However, it did not exploit the fact that their data originated from active interventions into the cellular system.

In Fig. A.6, the clustering of Mediator subunits and genes based on fold changes and log-odds ratios is depicted. Both approaches lead to an almost identical (isomorphic) dendrogram, which also agrees well with the MC EMiNEM's signals graph (if edge directions are ignored). This means that the coarse grouping of Mediator subunits can already be read off the expression profiles. However, MC EMiNEM provides more detailed information on the hierarchical structure of the Mediator organization, as well as on the attachment of effects.

A.4.3. Gene set enrichment analysis for transcription factor targets

The gene set enrichment analysis was done according to [6], using the **R**/*Bioconductor* package mgsa (version 1.2.0) [7], with the following parameters: p=seq(0.02,0.2,by=0.004), alpha=seq(0.02,0.98,by=0.02), beta=seq(0.02,0.98,by=0.02), steps=$(5 \cdot 10^6)$, restarts=10. Restraining p to small values ensures a sparse solution. For each gene set a mgsa run was performed, taking into account only TFs being mapped to at least one gene of the study set. The total population has been set to all effect genes being part of the corresponding Nested Effects Model. Only TFs being enriched with a probability of $\geqq 50\%$ were valued as significant and further analyzed.

Two examples of enriched TFs are explained in more detail in the main text (see Fig. 8.11). For similar figures for all Mediator subunit - transcription factor pairs, please refer to Supplementary file 2 of [77], which is provided in digital form along with this thesis.
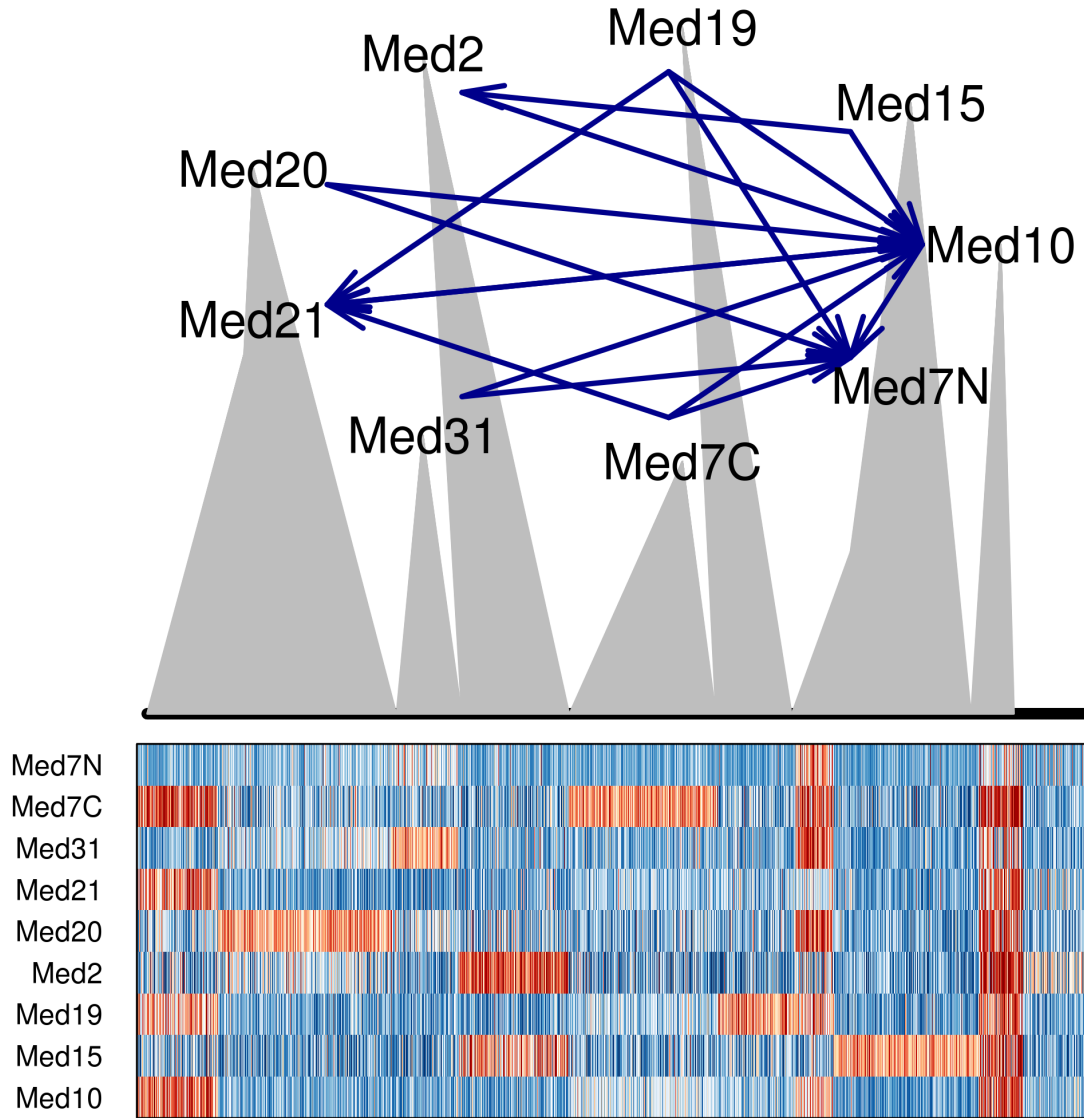
Figure A.3.: The Mediator-NEM treating all subunits as individual nodes, version 1 (result of nine runs out of ten). Above, the resulting signals graph is shown, below the underlying R matrix, clustered according to the final gene attachment (rows: perturbations, columns: effects on measured genes). Red color indicates a positive log-ratio value, blue color indicates a negative log-ratio value. The stronger the color of a field R_{kj}, $k \in \mathcal{E}$, $j \in \mathcal{S}$, the higher the probability that the measured data is due to the fact that there actually is an effect of signal j on gene k, or, that there is no effect, respectively. There exists an edge Med10 → Med21 as well as Med21 → Med10. The similarity between the two perturbations is also clearly visible in the perturbation profile. Thus, in the following, the two Mediator subunits are treated as one node in the NEM.

Figure A.4.: The Mediator-NEM treating all subunits as individual nodes, version 2 (result of one runs out of ten). Above, the resulting signals graph is shown, below the underlying R matrix, clustered according to the final gene attachment (rows: perturbations, columns: effects on measured genes). Red color indicates a positive log-ratio value, blue color indicates a negative log-ratio value. The stronger the color of a field R_{kj}, $k \in \mathcal{E}$, $j \in \mathcal{S}$, the higher the probability that the measured data is due to the fact that there actually is an effect of signal j on gene k, or, that there is no effect, respectively. There exists an edge Med10 $\rightarrow$ Med21 as well as Med21 $\rightarrow$ Med10.The similarity between the two perturbations is also clearly visible in the perturbation profile. Thus, in the following, the two Mediator subunits are treated as one node in the NEM.

Figure A.5.: The final Mediator-NEM, where Med10 and Med21 are combined to one single node. Above, the resulting signals graph is shown, below the underlying R matrix, clustered according to the final gene attachment (rows: perturbations, columns: effects on measured genes). Red color indicates a positive log-ratio value, blue color indicates a negative log-ratio value. The stronger the color of a field R_{kj}, $k \in \mathcal{E}$, $j \in \mathcal{S}$, the higher the probability that the measured data is due to the fact that there actually is an effect of signal j on gene k, or, that there is no effect, respectively. A detailed discussion of the results can be found in the main text.
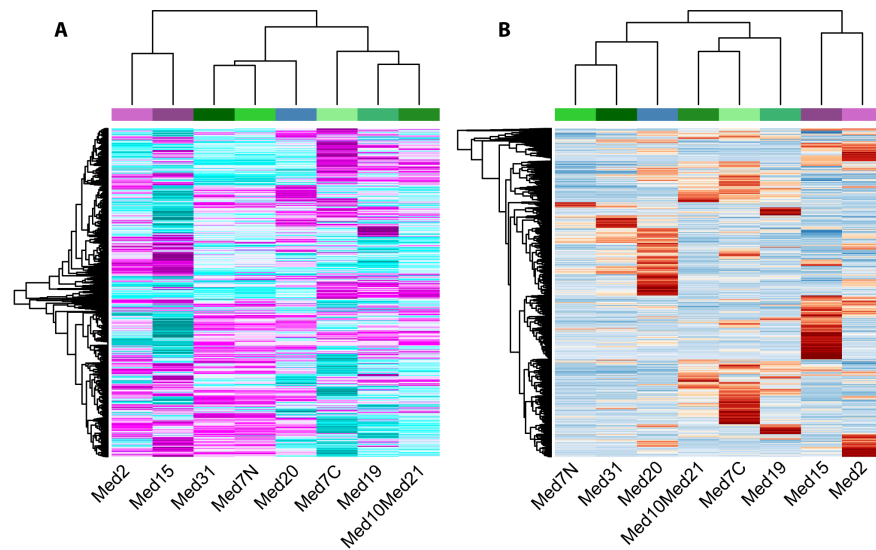
Figure A.6.: Clustering of Mediator subunits and genes based on **(A)** fold changes and **(B)** log-odds ratios. Mediator subunits are colored according to Fig. 8.9 and Fig. 8.10.

Study set	TF	in population	in study set	Estimate
Med2 - downregulated	TEC1	42	14	0.999
	YAP6	73	15	0.963
	GTS1	10	3	0.808
	SUM1	32	5	0.792
	YAP1	25	7	0.780
	SWI4	80	11	0.562
	ASH1	20	4	0.513
Med7C - downregulated	MBP1	77	22	0.991
Med7C - upregulated	RPN4	44	11	0.907
Med7N - downregulated	SWI5	51	7	0.910
	FKH2	65	11	0.901
	GLN3	63	8	0.664
	YOX1	3	1	0.510
Med10Med21 - downregulated	INO4	12	6	0.901
	STB5	14	3	0.557
Med10Med21 - upregulated	UME6	71	13	0.999
	HSF1	29	9	0.994
	HAP4	27	9	0.980
	SKN7	93	17	0.904
	SKO1	15	4	0.842
	HAP3	13	3	0.519

Table A.1.: This table provides the results of the gene set enrichment analysis conducted as outlined in Section A.4.3. First column: the studied gene set (i.e., the Mediator subunit and the direction of expression change); Second column: The transcription factor (TF) whose targets are enriched; Third column: the number of genes in the whole population annotated to this TF; Fourth column: the number of genes in the study set; Fifth column: the estimate for the enrichment (cutoff for this study: 0.5).

B. Supplementary material for Part III - RNA degradation by the exosome

B.1. A Michaelis-Menten based derivation of the catalytic efficiency

Michaelis-Menten type kinetics apply to enzymatic reactions of the type

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \overset{k_2}{\longrightarrow} E + P$$
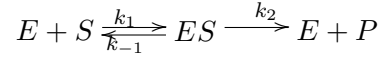
In this context, a specificity constant $\frac{k_{\text{cat}}}{K_M}$ has been defined, where $k_{\text{cat}} = k_2$ and $K_M = \frac{k_{-1}+k_2}{k_1}$ is the Michaelis-Menten constant [57]. We defined the RNA length-dependent *catalytic efficiency* according to this specificity constant, with the association rate corresponding to k_1, the dissociation rate corresponding to k_{-1} and the cleavage rate corresponding to k_2. Hence, the *catalytic efficiency* for a RNA substrate of length j results in $e_j = \frac{k_{c,j} \cdot k_{a,j}}{k_{d,j}+k_{c,j}}$.

B.2. Simulation

B.2.1. Data generation

The simulations were carried out as follows: Starting with a realistic set of parameters $\Theta_{\text{true}} = \{k_{a,j}^{\text{true}}|j \in J\} \cup k_c^{\text{true}}$ (which usually was obtained by applying a maximum likelihood parameter estimation method to our data), the "true" concentrations $r_j^{\Theta_{\text{true}}}(t_k)$ are computed by the above mentioned ODE solver (ode15s). Then, an artificial dataset $D = (R_{j,k})$ is generated by drawing

$$R_{j,k} \sim \mathcal{N}(r_j^{\Theta_{\text{true}}}(t_k), (\sigma_{j,k}^{\text{true}})^2) \,, \quad j \in J^\star, \ k = 1, ..., K \tag{B.1}$$

We performed MCMC runs with $(\sigma_{j,k}^{\text{true}})^2$ calculated based on Eq. 11.4, $r_j^{\Theta_{\text{true}}}(t_k)$ and error levels set to $\beta = 0$ and $\alpha = 5\%, 10\%$ and 25%.

B.3. Application: RNA degradation by the archaeal exosome

The results of the MCMC sampling approach applied to all exosome variants are presented in Fig. B.1, Table B.1, Table B.2, Fig. B.2 and Fig. B.3.
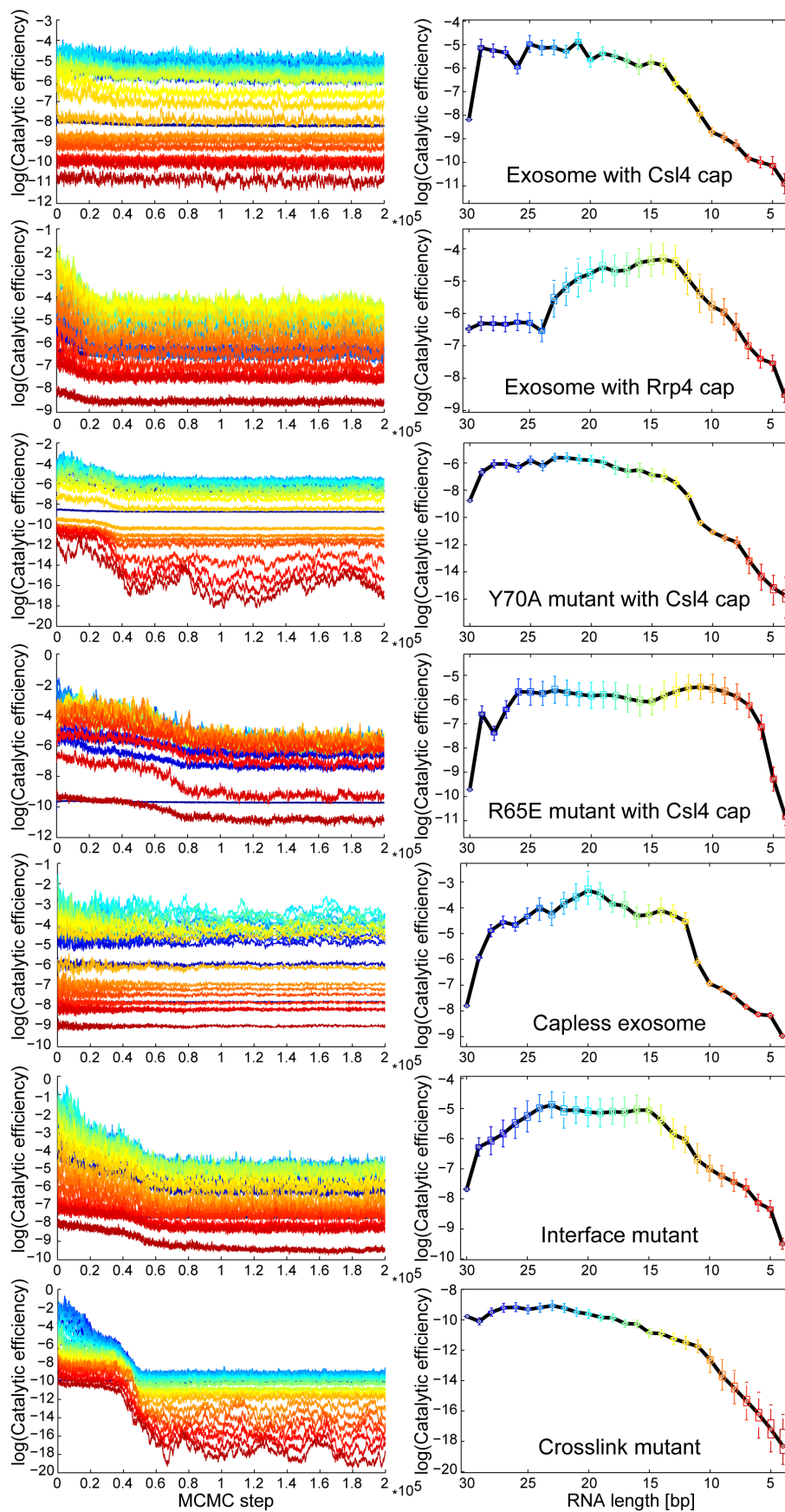
Figure B.1.: On the left-hand side the traceplots for all exosome variants are depicted, the right-hand side shows the boxplots for the corresponding posterior distributions (burn-in = 150000). The Markov chains vary with regard to both convergence speed and variability.

	Csl4 exosome	Rrp4 exosome	Csl4-Y70A exosome	Csl4-R65E exosome
30	2.795E-04 [2.750E-04,2.839E-04]	1.539E-03 [1.497E-03,1.585E-03]	1.560E-04 [1.543E-04,1.578E-04]	6.016E-05 [5.965E-05,6.069E-05]
29	5.875E-03 [5.373E-03,6.454E-03]	1.814E-03 [1.721E-03,1.917E-03]	1.258E-03 [1.178E-03,1.339E-03]	1.362E-03 [1.254E-03,1.485E-03]
28	5.223E-03 [4.875E-03,5.606E-03]	1.808E-03 [1.712E-03,1.918E-03]	2.276E-03 [2.120E-03,2.461E-03]	6.174E-04 [5.727E-04,6.633E-04]
27	4.842E-03 [4.558E-03,5.146E-03]	1.768E-03 [1.673E-03,1.882E-03]	2.297E-03 [2.138E-03,2.493E-03]	1.643E-03 [1.511E-03,1.807E-03]
26	2.553E-03 [2.398E-03,2.749E-03]	1.885E-03 [1.771E-03,2.018E-03]	1.781E-03 [1.631E-03,1.938E-03]	3.421E-03 [2.968E-03,3.892E-03]
25	6.895E-03 [6.350E-03,7.578E-03]	1.849E-03 [1.719E-03,2.010E-03]	2.974E-03 [2.762E-03,3.236E-03]	3.364E-03 [2.945E-03,3.863E-03]
24	5.862E-03 [5.443E-03,6.366E-03]	1.440E-03 [1.331E-03,1.575E-03]	1.998E-03 [1.823E-03,2.191E-03]	3.170E-03 [2.819E-03,3.611E-03]
23	5.954E-03 [5.534E-03,6.471E-03]	3.824E-03 [3.333E-03,4.454E-03]	3.654E-03 [3.413E-03,3.946E-03]	3.655E-03 [3.237E-03,4.283E-03]
22	4.999E-03 [4.675E-03,5.339E-03]	5.574E-03 [4.767E-03,6.429E-03]	3.606E-03 [3.334E-03,3.987E-03]	3.304E-03 [2.897E-03,3.753E-03]
21	7.714E-03 [7.037E-03,8.450E-03]	7.352E-03 [6.372E-03,8.604E-03]	3.174E-03 [2.935E-03,3.447E-03]	3.090E-03 [2.745E-03,3.468E-03]
20	3.480E-03 [3.249E-03,3.796E-03]	8.446E-03 [7.308E-03,9.706E-03]	2.988E-03 [2.722E-03,3.250E-03]	2.874E-03 [2.539E-03,3.354E-03]
19	4.663E-03 [4.354E-03,4.994E-03]	1.041E-02 [9.314E-03,1.192E-02]	2.621E-03 [2.364E-03,2.874E-03]	3.022E-03 [2.641E-03,3.457E-03]
18	4.116E-03 [3.866E-03,4.390E-03]	9.138E-03 [8.162E-03,1.040E-02]	1.730E-03 [1.543E-03,1.966E-03]	2.905E-03 [2.536E-03,3.340E-03]
17	3.408E-03 [3.181E-03,3.685E-03]	9.433E-03 [8.392E-03,1.086E-02]	1.350E-03 [1.207E-03,1.514E-03]	2.574E-03 [2.237E-03,2.970E-03]
16	2.640E-03 [2.445E-03,2.880E-03]	1.186E-02 [1.061E-02,1.323E-02]	1.474E-03 [1.305E-03,1.671E-03]	2.311E-03 [1.986E-03,2.747E-03]
15	3.172E-03 [2.942E-03,3.405E-03]	1.267E-02 [1.121E-02,1.437E-02]	1.005E-03 [8.912E-04,1.130E-03]	2.254E-03 [1.987E-03,2.611E-03]
14	2.782E-03 [2.589E-03,2.990E-03]	1.324E-02 [1.177E-02,1.505E-02]	9.318E-04 [8.347E-04,1.052E-03]	2.908E-03 [2.557E-03,3.429E-03]
13	1.295E-03 [1.209E-03,1.396E-03]	1.203E-02 [1.061E-02,1.370E-02]	5.594E-04 [5.096E-04,6.235E-04]	3.481E-03 [2.917E-03,4.199E-03]
12	7.501E-04 [6.997E-04,8.093E-04]	7.407E-03 [6.578E-03,8.517E-03]	2.138E-04 [2.009E-04,2.288E-04]	4.000E-03 [3.526E-03,4.594E-03]
11	3.516E-04 [3.239E-04,3.829E-04]	4.556E-03 [3.939E-03,5.289E-03]	3.070E-05 [2.949E-05,3.199E-05]	4.172E-03 [3.695E-03,4.743E-03]
10	1.649E-04 [1.577E-04,1.731E-04]	3.109E-03 [2.751E-03,3.567E-03]	1.524E-05 [1.461E-05,1.588E-05]	3.902E-03 [3.382E-03,4.536E-03]
9	1.304E-04 [1.245E-04,1.373E-04]	2.572E-03 [2.331E-03,2.866E-03]	1.023E-05 [9.582E-06,1.091E-05]	3.403E-03 [2.975E-03,3.982E-03]
8	9.420E-05 [8.927E-05,9.982E-05]	1.621E-03 [1.467E-03,1.809E-03]	7.271E-06 [6.480E-06,7.954E-06]	2.767E-03 [2.446E-03,3.183E-03]
7	5.395E-05 [5.168E-05,5.629E-05]	9.026E-04 [8.251E-04,1.005E-03]	1.921E-06 [1.474E-06,2.351E-06]	1.933E-03 [1.715E-03,2.189E-03]
6	4.559E-05 [4.285E-05,4.825E-05]	6.090E-04 [5.677E-04,6.550E-04]	6.026E-07 [4.588E-07,7.562E-07]	8.088E-04 [7.044E-04,9.015E-04]
5	3.937E-05 [3.588E-05,4.347E-05]	5.328E-04 [5.014E-04,5.669E-04]	2.553E-07 [1.795E-07,3.244E-07]	9.304E-05 [8.226E-05,1.054E-04]
4	1.841E-05 [1.651E-05,2.039E-05]	1.952E-04 [1.856E-04,2.062E-04]	1.452E-07 [9.204E-08,2.228E-07]	1.946E-05 [1.771E-05,2.108E-05]

Table B.1.: Median [1st quartile, 3rd quartile] of the catalytic efficiency posterior distributions.

	Capless exosome	Interface mutant	Crosslink mutant
30	4.102E-04 [4.065E-04,4.139E-04]	4.616E-04 [4.528E-04,4.701E-04]	5.638E-05 [5.560E-05,5.717E-05]
29	2.649E-03 [2.566E-03,2.735E-03]	1.871E-03 [1.743E-03,2.027E-03]	4.112E-05 [3.881E-05,4.363E-05]
28	7.568E-03 [7.086E-03,8.075E-03]	2.295E-03 [2.055E-03,2.594E-03]	7.489E-05 [7.026E-05,8.029E-05]
27	1.048E-02 [9.778E-03,1.111E-02]	2.903E-03 [2.620E-03,3.278E-03]	1.004E-04 [9.299E-05,1.079E-04]
26	9.411E-03 [8.635E-03,1.024E-02]	4.090E-03 [3.627E-03,4.662E-03]	1.035E-04 [9.679E-05,1.123E-04]
25	1.302E-02 [1.217E-02,1.430E-02]	5.226E-03 [4.556E-03,5.903E-03]	8.996E-05 [8.405E-05,9.675E-05]
24	1.832E-02 [1.602E-02,2.018E-02]	6.806E-03 [6.107E-03,7.643E-03]	1.013E-04 [9.344E-05,1.088E-04]
23	1.444E-02 [1.240E-02,1.566E-02]	7.590E-03 [6.845E-03,8.514E-03]	1.157E-04 [1.072E-04,1.261E-04]
22	2.146E-02 [1.855E-02,2.557E-02]	6.305E-03 [5.449E-03,7.340E-03]	9.800E-05 [9.069E-05,1.056E-04]
21	2.747E-02 [2.395E-02,3.181E-02]	6.426E-03 [5.737E-03,7.165E-03]	7.569E-05 [7.070E-05,8.114E-05]
20	3.581E-02 [2.889E-02,4.279E-02]	5.992E-03 [5.351E-03,6.855E-03]	6.651E-05 [6.193E-05,7.145E-05]
19	3.121E-02 [2.567E-02,3.663E-02]	5.783E-03 [5.171E-03,6.542E-03]	5.327E-05 [5.025E-05,5.642E-05]
18	2.125E-02 [1.925E-02,2.306E-02]	6.058E-03 [5.460E-03,6.797E-03]	5.150E-05 [4.827E-05,5.524E-05]
17	1.965E-02 [1.786E-02,2.309E-02]	5.925E-03 [5.318E-03,6.782E-03]	3.556E-05 [3.331E-05,3.776E-05]
16	1.339E-02 [1.185E-02,1.462E-02]	6.396E-03 [5.737E-03,7.296E-03]	3.429E-05 [3.226E-05,3.669E-05]
15	1.394E-02 [1.277E-02,1.542E-02]	6.364E-03 [5.782E-03,7.048E-03]	1.918E-05 [1.795E-05,2.063E-05]
14	1.651E-02 [1.423E-02,1.825E-02]	4.508E-03 [4.023E-03,5.168E-03]	1.832E-05 [1.710E-05,1.955E-05]
13	1.394E-02 [1.282E-02,1.612E-02]	2.882E-03 [2.563E-03,3.289E-03]	1.298E-05 [1.187E-05,1.421E-05]
12	1.076E-02 [9.805E-03,1.170E-02]	2.378E-03 [2.132E-03,2.659E-03]	1.012E-05 [9.149E-06,1.128E-05]
11	2.205E-03 [2.126E-03,2.293E-03]	1.232E-03 [1.092E-03,1.415E-03]	7.864E-06 [7.083E-06,8.713E-06]
10	9.687E-04 [9.409E-04,9.954E-04]	9.045E-04 [8.203E-04,1.023E-03]	3.299E-06 [2.618E-06,4.042E-06]
9	7.784E-04 [7.556E-04,8.012E-04]	7.142E-04 [6.573E-04,7.831E-04]	1.112E-06 [8.742E-07,1.510E-06]
8	5.888E-04 [5.750E-04,6.086E-04]	5.883E-04 [5.429E-04,6.366E-04]	5.029E-07 [3.878E-07,7.019E-07]
7	3.935E-04 [3.848E-04,4.033E-04]	4.795E-04 [4.462E-04,5.175E-04]	2.157E-07 [1.576E-07,2.931E-07]
6	2.924E-04 [2.858E-04,2.989E-04]	2.929E-04 [2.740E-04,3.150E-04]	9.133E-08 [5.832E-08,1.239E-07]
5	2.806E-04 [2.744E-04,2.886E-04]	2.392E-04 [2.229E-04,2.567E-04]	3.399E-08 [1.893E-08,6.257E-08]
4	1.263E-04 [1.240E-04,1.290E-04]	7.528E-05 [7.206E-05,7.893E-05]	1.104E-08 [6.889E-09,3.346E-08]

Table B.2.: Median [1$^{\text{st}}$ quartile, 3$^{\text{rd}}$ quartile] of the catalytic efficiency posterior distributions.
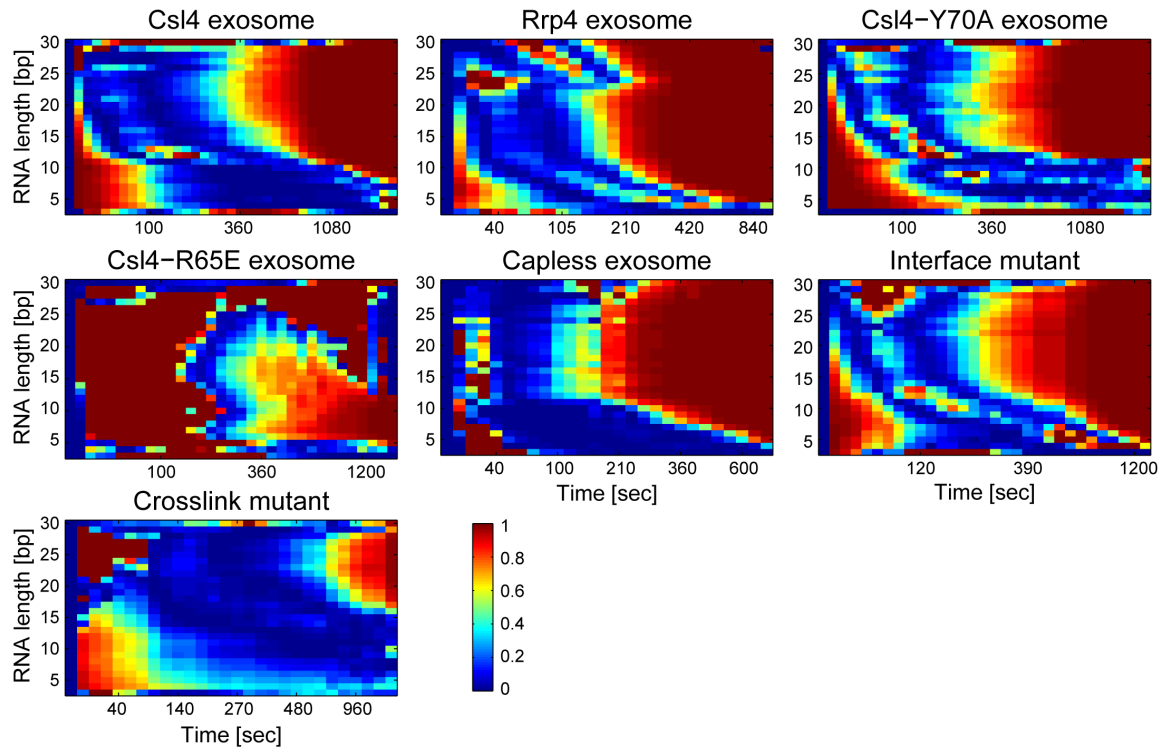
Figure B.2.: The relative squared error induced by the MCMC sampling approach is calculated for each exosome variant based on the averaged predictions of 1000 parameter sets, randomly drawn from the stationary phase of the final Markov chain. For a more detailed visualization, the color scheme has been scaled such that all values ≥ 1 have the same color. Obviously, individual RNA measurements with higher values can be fitted very well, while areas with lower amounts of RNA are fitted relatively poorly, which is in accordance with the results derived from simulation runs (see Fig. 13.8).
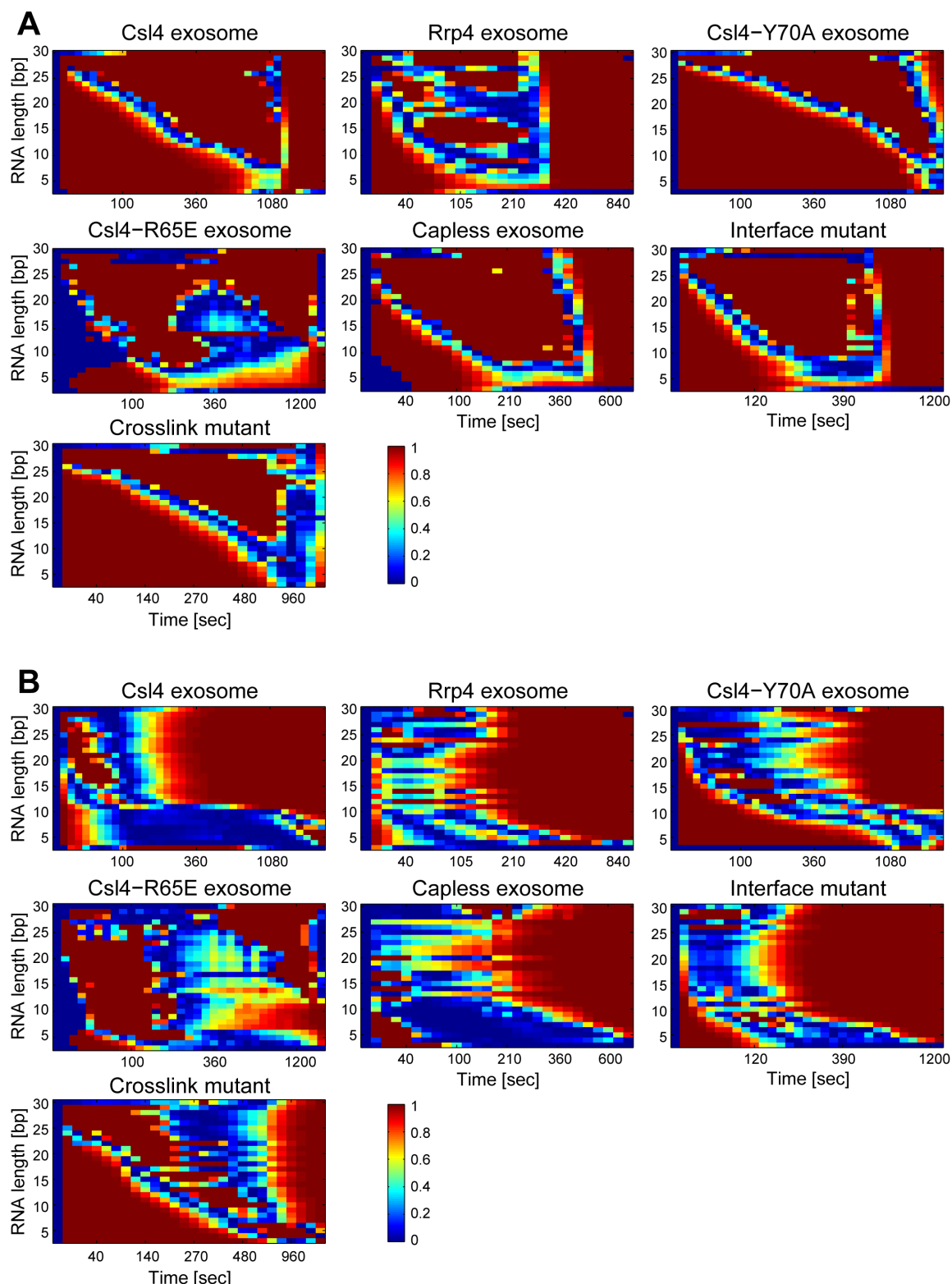
Figure B.3.: The relative squared error induced by the least-squares fitting is calculated for each exosome variant based on predictions of the estimated parameter sets. For a more detailed visualization, the color scheme has been scaled such that all values ≥ 1 have the same color. **(A)** The least-squares fitting has been initialized with the same parameters used for the initialization of the MCMC sampling. **(B)** The least-squares fitting has been initialized with a random sample from the stationary phase of the Markov chain (the same that has been used to generate Fig. B.2). Again, the results are in accordance with what has been seen in simulation runs (see Fig. 13.9): While the straightforward optimization method yields good results (respectively, results that are similar to those of the MCMC approach) when the initial parameters are already close to the true ones, it performs very poorly when no prior information is available.

C. Supplementary material for Part IV - Hematopoietic stem cell differentiation

C.1. Reversible-jump MCMC sampling

Let $m, n \in M$, be part of a family of models M, and Θ_m and Θ_n be the corresponding parameter sets. To ensure a common measure for different model classes, the parameter sets have to be extended ($\bar{\Theta}_m = (\Theta_m, u_{m,n})$ and $\bar{\Theta}_n = (\Theta_n, u_{n,m})$), and a deterministic, differentiable, invertible dimension matching function f has to be defined such that $f_{m \to n}(f_{n \to m}(\bar{\Theta}_n)) = (\bar{\Theta}_n)$. To move from state (n, Θ_n) to state (m, Θ_m), $u_{n,m} \sim q(\bullet | n, \Theta_n)$ is generated. The move is accepted with probability $\min(A_{n \to m}, 1)$, with

$$A_{n \to m} = \frac{p(m, \Theta'_m)}{p(n, \Theta_n)} \cdot \frac{q(n|m)}{q(m|n)} \cdot \frac{q_{m \to n}(u_{m,n}|m, \Theta'_m)}{q_{n \to m}(u_{n,m}|n, \Theta_n)} \cdot \mathcal{J}_{f_{n \to m}}$$

and with $\Theta'_m = f_{n \to m}(\Theta_n, u_{n,m})$ and $\mathcal{J}_{f_{n \to m}}$ being the Jacobian of the transformation $f_{n \to m}$. Since the implementation of a reversible-jump approach is strongly problem dependent, no further details are provided here [2, Chapter 3]. An application for reversible-jump MCMC sampling is presented in Part IV.

Derivation of the Jacobian

The Jacobian matrix includes the first-order partial derivatives of all functions included in the transformation with regard to all current parameters. Here, this results in a $|\bar{\Theta}_n| \times |\bar{\Theta}_m|$ matrix ($|\bar{\Theta}_n| = |\bar{\Theta}_m|$ per definition) where the lines correspond to the functions and the columns correspond to the parameters. The determinant of this square matrix is called the Jacobian [2]:

$$\mathcal{J}_{f_{n \to m}} = |det \frac{\partial f_{n \to m}(\Theta_n, u_{n,m})}{\partial (\Theta_n, u_{n,m})}|$$

Jumping from the selective to the instructive scenario

Here, the Jacobian matrix is

	p_{diff}^n	Δ_{diff}	$p_{\text{death}_1}^n$	$p_{\text{death}_2}^n$
$p_{\text{diff}_1}' = p_{\text{diff}}^n + \Delta_{\text{diff}}$	1	1	0	0
$p_{\text{diff}_2}' = p_{\text{diff}}^n - \Delta_{\text{diff}}$	1	-1	0	0
$p_{\text{death}}' = \frac{p_{\text{death}_1}^n + p_{\text{death}_2}^n}{2}$	0	0	$\frac{1}{2}$	$\frac{1}{2}$
$\Delta_{\text{death}} = \frac{p_{\text{death}_1}^n - p_{\text{death}_2}^n}{2}$	0	0	$\frac{1}{2}$	$-\frac{1}{2}$

Hence, $\mathcal{J}_{\text{sel}\to\text{instr}} = 1$.

Jumping from the instructive to the selective scenario

Here, the Jacobian matrix is

	$p_{\text{diff}_1}^n$	$p_{\text{diff}_2}^n$	p_{death}^n	Δ_{death}
$p_{\text{diff}}' = \frac{p_{\text{diff}_1}^n + p_{\text{diff}_2}^n}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0
$\Delta_{\text{diff}} = \frac{p_{\text{diff}_1}^n - p_{\text{diff}_2}^n}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	0	0
$p_{\text{death}_1}' = p_{\text{death}}^n + \Delta_{\text{death}}$	0	0	1	1
$p_{\text{death}_2}' = p_{\text{death}}^n - \Delta_{\text{death}}$	0	0	1	-1

Hence, $\mathcal{J}_{\text{instr}\to\text{sel}} = 1$.

C.2. The Dirichlet distribution

The density function of the Dirichlet distribution is given by

$$\mathcal{D}(\mu|\alpha) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1)\cdot ... \cdot \Gamma(\alpha_K)} \cdot \prod_{k=1}^{K} \mu_k^{\alpha_k - 1}$$

with $\mu = (\mu_1, ..., \mu_K)^T$, such that $0 \leq \mu_k \leq 1, \forall k \in 1, ..., K$ and $\sum_{k=1}^{K} \mu_k = 1$, and $\alpha = (\alpha_1, ..., \alpha_K)^T$ being the parameters of the distribution, with $a_k > 0, \forall k \in 1, ..., K$ and $\alpha_0 = \sum_{k=1}^{K} \alpha_k$. $\Gamma(\alpha_x)$ denotes the gamma distribution. The expectation value and variance are then: $\text{E}[\mu_k] = \frac{\alpha_k}{\alpha_0}$ and $\text{Var}[\mu_k] = \frac{\alpha_0 \cdot (\alpha_0 - \alpha_k)}{\alpha_0^2 \cdot (\alpha_0 + 1)}$ [8, Chapter 2].

The Dirichlet distribution as proposal function

Here, we set $\alpha = c \cdot \theta_n$, with $\theta_n \subset \Theta_n$ being a subset of the model parameters and $c = 500$ ensuring high values for α. This means that if α is interpreted as the counts for the corresponding transitions, $\theta' = \mu$ is sampled centered around their relative frequencies, i.e., around θ_n. The width of the proposal function is determined by α (i.e., higher values of α result in a narrower proposal function). The suggested model parameters θ' add up to one by definition.

C.3. The beta distribution

The density function of the beta distribution (which is a special case of the multivariate Dirichlet distribution) is given by

$$B(\mu|a,b) = \frac{\Gamma(a+b)}{\Gamma(a) \cdot \Gamma(b)} \cdot \mu^{a-1} \cdot (1-\mu)^{b-1}$$

with $\mu \in [0,1]$, and $a > 0$, $b > 0$ being the parameters of the distribution. $\Gamma(x)$ denotes the gamma distribution. The expectation value and variance are then: $\mathrm{E}[\mu] = \frac{a}{a+b}$ and $\mathrm{Var}[\mu] = \frac{ab}{(a+b)^2 \cdot (\alpha+b+1)}$ [8, Chapter 2].

The beta distribution as proposal function

Here, we set $a = c \cdot p^n_{\text{death/div}_{1/2}}$ and $b = c \cdot (1 - p^n_{\text{death/div}_{1/2}})$, with $c = 500$ ensuring high values for a and b. This means that if a is interpreted as the count for the corresponding transition, $p'_{\text{death/div}_{1/2}} = \mu$ is sampled centered around its relative frequency, i.e., around $p^n_{\text{death/div}_{1/2}}$. The width of the proposal function is determined by a and b (i.e., higher values of a and b result in a narrower proposal function).

Bibliography

[1] Anchang, B., Sadeh, M. J., Jacob, J., Tresch, A., Vlad, M. O., Oefner, P. J., and Spang, R. Modeling the temporal interplay of molecular signaling and gene expression by using dynamic nested effects models. *Proc Natl Acad Sci U S A 106*, 16 (Apr 2009), 6447–6452.

[2] Andrieu, C., de Freitas, N., Doucet, A., and Jordan, M. I. An Introduction to MCMC for Machine Learning. *Machine Learning 50* (2003), 5–43.

[3] Ansari, S. A., Ganapathi, M., Benschop, J. J., Holstege, F. C. P., Wade, J. T., and Morse, R. H. Distinct role of Mediator tail module in regulation of SAGA-dependent, TATA-containing genes in yeast. *EMBO J 31*, 1 (Jan 2012), 44–57.

[4] Baidoobonso, S. M., Guidi, B. W., and Myers, L. C. Med19(Rox3) regulates Intermodule interactions in the Saccharomyces cerevisiae mediator complex. *J Biol Chem 282*, 8 (Feb 2007), 5551–5559.

[5] Basso, K., Margolin, A. A., Stolovitzky, G., Klein, U., Dalla-Favera, R., and Califano, A. Reverse engineering of regulatory networks in human B cells. *Nat Genet 37*, 4 (Apr 2005), 382–390.

[6] Bauer, S., Gagneur, J., and Robinson, P. N. GOing Bayesian: model-based gene set analysis of genome-scale data. *Nucleic Acids Res 38*, 11 (Jun 2010), 3523–3532.

[7] Bauer, S., Robinson, P. N., and Gagneur, J. Model-based gene set analysis for Bioconductor. *Bioinformatics 27*, 13 (Jul 2011), 1882–1883.

[8] Bishop, C. M. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006.

[9] Bolstad, W. M. *Understanding Computational Bayesian Statistics*. Series in Computational Statistics. Wiley, 2010.

[10] Borggrefe, T., and Yue, X. Interactions between subunits of the Mediator complex with gene-specific transcription factors. *Semin Cell Dev Biol 22*, 7 (Sep 2011), 759–768.

[11] Bourbon, H.-M. Comparative genomics supports a deep evolutionary origin for the large, four-module transcriptional mediator complex. *Nucleic Acids Res 36*, 12 (Jul 2008), 3993–4008.

[12] BOURBON, H.-M., AGUILERA, A., ANSARI, A. Z., ASTURIAS, F. J., BERK, A. J., BJORKLUND, S., BLACKWELL, T. K., BORGGREFE, T., CAREY, M., CARLSON, M., CONAWAY, J. W., CONAWAY, R. C., EMMONS, S. W., FONDELL, J. D., FREEDMAN, L. P., FUKASAWA, T., GUSTAFSSON, C. M., HAN, M., HE, X., HERMAN, P. K., HINNEBUSCH, A. G., HOLMBERG, S., HOLSTEGE, F. C., JAEHNING, J. A., KIM, Y.-J., KURAS, L., LEUTZ, A., LIS, J. T., MEISTERERNEST, M., NAAR, A. M., NASMYTH, K., PARVIN, J. D., PTASHNE, M., REINBERG, D., RONNE, H., SADOWSKI, I., SAKURAI, H., SIPICZKI, M., STERNBERG, P. W., STILLMAN, D. J., STRICH, R., STRUHL, K., SVEJSTRUP, J. Q., TUCK, S., WINSTON, F., ROEDER, R. G., AND KORNBERG, R. D. A unified nomenclature for protein subunits of mediator complexes linking transcriptional regulators to RNA polymerase II. *Mol Cell 14*, 5 (Jun 2004), 553–557.

[13] BOX, G. E. P., AND DRAPER, N. R. *Empirical Model-Building and Response Surfaces.* Series in Probability and Statistics. Wiley, 1987.

[14] BÜTTNER, K., WENIG, K., AND HOPFNER, K.-P. Structural framework for the mechanism of archaeal exosomes in RNA processing. *Mol Cell 20*, 3 (Nov 2005), 461–471.

[15] CARR, J. M., AND WALES, D. J. Global optimization and folding pathways of selected alpha-helical proteins. *J Chem Phys 123*, 23 (Dec 2005), 234901.

[16] CHERRY, J. M., HONG, E. L., AMUNDSEN, C., BALAKRISHNAN, R., BINKLEY, G., CHAN, E. T., CHRISTIE, K. R., COSTANZO, M. C., DWIGHT, S. S., ENGEL, S. R., FISK, D. G., HIRSCHMAN, J. E., HITZ, B. C., KARRA, K., KRIEGER, C. J., MIYASATO, S. R., NASH, R. S., PARK, J., SKRZYPEK, M. S., SIMISON, M., WENG, S., AND WONG, E. D. Saccharomyces Genome Database: the genomics resource of budding yeast. *Nucleic Acids Res 40*, Database issue (Jan 2012), D700–D705.

[17] CONAWAY, R. C., AND CONAWAY, J. W. Origins and activity of the Mediator complex. *Semin Cell Dev Biol 22*, 7 (Sep 2011), 729–734.

[18] DELLAERT, F. The Expectation Maximization Algorithm. Tech. rep., College of Computing, Georgia Institute of Technology, February 2002.

[19] DEMPSTER, A. P., LAIRD, N. M., AND RUBIN, D. B. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological) 39*, 1 (1977), pp. 1–38.

[20] DUNN, P. F. *Measurement and data analysis for engineering and science.* McGraw-Hill series in mechanical engineering. McGraw-Hill Higher Education, New York, 2005.

[21] DURBIN, B. P., HARDIN, J. S., HAWKINS, D. M., AND ROCKE, D. M. A variance-stabilizing transformation for gene-expression microarray data. *Bioinformatics 18 Suppl 1* (2002), S105–S110.

[22] ELMLUND, H., BARAZNENOK, V., LINDAHL, M., SAMUELSEN, C. O., KOECK, P. J. B., HOLMBERG, S., HEBERT, H., AND GUSTAFSSON, C. M. The cyclin-dependent kinase 8 module sterically blocks Mediator interactions with RNA polymerase II. *Proc Natl Acad Sci U S A 103*, 43 (Oct 2006), 15788–15793.

[23] EVGUENIEVA-HACKENBERG, E., WALTER, P., HOCHLEITNER, E., LOTTSPEICH, F., AND KLUG, G. An exosome-like complex in Sulfolobus solfataricus. *EMBO Rep 4*, 9 (Sep 2003), 889–893.

[24] FROEHLICH, H., FELLMANN, M., SUELTMANN, H., POUSTKA, A., AND BEISSBARTH, T. Large scale statistical inference of signaling pathways from RNAi and microarray data. *BMC Bioinformatics 8* (2007), 386.

[25] FROEHLICH, H., MARKOWETZ, F., TRESCH, A., NIEDERBERGER, T., BENDER, C., MANECK, M., LOTTAZ, C., AND BEISSBARTH, T. *nem: Nested Effects Models to reconstruct phenotypic hierarchies*. R package version 2.31.1.

[26] FRÖHLICH, H., FELLMANN, M., SÜLTMANN, H., POUSTKA, A., AND BEISSBARTH, T. Estimating large-scale signaling networks through nested effect models with intervention effects from microarray data. *Bioinformatics 24*, 22 (Nov 2008), 2650–2656.

[27] FRÖHLICH, H., TRESCH, A., AND BEISSBARTH, T. Nested effects models for learning signaling networks from perturbation data. *Biom J 51*, 2 (Apr 2009), 304–323.

[28] GAO, F., FOAT, B. C., AND BUSSEMAKER, H. J. Defining transcriptional networks through integrative modeling of mRNA expression and transcription factor binding data. *BMC Bioinformatics 5* (Mar 2004), 31.

[29] GAUTIER, L., COPE, L., BOLSTAD, B. M., AND IRIZARRY, R. A. affy–analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics 20*, 3 (Feb 2004), 307–315.

[30] GENTLEMAN, R. C., CAREY, V. J., BATES, D. M., BOLSTAD, B., DETTLING, M., DUDOIT, S., ELLIS, B., GAUTIER, L., GE, Y., GENTRY, J., HORNIK, K., HOTHORN, T., HUBER, W., IACUS, S., IRIZARRY, R., LEISCH, F., LI, C., MAECHLER, M., ROSSINI, A. J., SAWITZKI, G., SMITH, C., SMYTH, G., TIERNEY, L., YANG, J. Y. H., AND ZHANG, J. Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol 5*, 10 (2004), R80.

[31] GIAEVER, G., CHU, A. M., NI, L., CONNELLY, C., RILES, L., VÉRONNEAU, S., DOW, S., LUCAU-DANILA, A., ANDERSON, K., ANDRÉ, B., ARKIN, A. P., ASTROMOFF, A., EL-BAKKOURY, M., BANGHAM, R., BENITO, R., BRACHAT, S., CAMPANARO,

S., Curtiss, M., Davis, K., Deutschbauer, A., Entian, K.-D., Flaherty, P., Foury, F., Garfinkel, D. J., Gerstein, M., Gotte, D., Güldener, U., Hegemann, J. H., Hempel, S., Herman, Z., Jaramillo, D. F., Kelly, D. E., Kelly, S. L., Kötter, P., LaBonte, D., Lamb, D. C., Lan, N., Liang, H., Liao, H., Liu, L., Luo, C., Lussier, M., Mao, R., Menard, P., Ooi, S. L., Revuelta, J. L., Roberts, C. J., Rose, M., Ross-Macdonald, P., Scherens, B., Schimmack, G., Shafer, B., Shoemaker, D. D., Sookhai-Mahadeo, S., Storms, R. K., Strathern, J. N., Valle, G., Voet, M., Volckaert, G., yun Wang, C., Ward, T. R., Wilhelmy, J., Winzeler, E. A., Yang, Y., Yen, G., Youngman, E., Yu, K., Bussey, H., Boeke, J. D., Snyder, M., Philippsen, P., Davis, R. W., and Johnston, M. Functional profiling of the Saccharomyces cerevisiae genome. *Nature 418*, 6896 (Jul 2002), 387–391.

[32] Glauche, I., Lorenz, R., Hasenclever, D., and Roeder, I. A novel view on stem cell development: analysing the shape of cellular genealogies. *Cell Prolif 42*, 2 (Apr 2009), 248–263.

[33] Green, P. J. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika 82* (1995), 711–732.

[34] Gromöller, A., and Lehming, N. Srb7p is a physical and physiological target of Tup1p. *EMBO J 19*, 24 (Dec 2000), 6845–6852.

[35] Hartung, S. *Analyzing Protein - Nucleic Acid Complexes using Hybrid Methods I. The DNA Damage Checkpoint Protein DisA II. Structural Biochemistry of RNA Turnover by the Exosome.* PhD thesis, Ludwig-Maximilians-University of Munich (LMU), 2008.

[36] Hartung, S., Niederberger, T., Hartung, M., Tresch, A., and Hopfner, K.-P. Quantitative analysis of processive RNA degradation by the archaeal RNA exosome. *Nucleic Acids Res 38*, 15 (Aug 2010), 5166–5176.

[37] Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Series in Statistics. Springer, 2001.

[38] Hastings, W. K. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika 57*, 1 (1970), 97–109.

[39] Holstege, F. C., Jennings, E. G., Wyrick, J. J., Lee, T. I., Hengartner, C. J., Green, M. R., Golub, T. R., Lander, E. S., and Young, R. A. Dissecting the regulatory circuitry of a eukaryotic genome. *Cell 95*, 5 (Nov 1998), 717–728.

[40] Houseley, J., LaCava, J., and Tollervey, D. RNA-quality control by the exosome. *Nat Rev Mol Cell Biol 7*, 7 (Jul 2006), 529–539.

[41] Hu, Z., Killion, P. J., and Iyer, V. R. Genetic reconstruction of a functional transcriptional regulatory network. *Nat Genet 39*, 5 (May 2007), 683–687.

[42] Huber, W., von Heydebreck, A., Sültmann, H., Poustka, A., and Vingron, M. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics 18 Suppl 1* (2002), S96–104.

[43] 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. Functional discovery via a compendium of expression profiles. *Cell 102*, 1 (Jul 2000), 109–126.

[44] Ihaka, R., and Gentleman, R. R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics 5*, 3 (1996), pp. 299–314.

[45] Imasaki, T., Calero, G., Cai, G., Tsai, K.-L., Yamada, K., Cardelli, F., Erdjument-Bromage, H., Tempst, P., Berger, I., Kornberg, G. L., Asturias, F. J., Kornberg, R. D., and Takagi, Y. Architecture of the Mediator head module. *Nature 475*, 7355 (Jul 2011), 240–243.

[46] Jackowiak, P., Nowacka, M., Strozycki, P. M., and Figlerowicz, M. RNA degradome–its biogenesis and functions. *Nucleic Acids Res 39*, 17 (Sep 2011), 7361–7370.

[47] Kaderali, L., Dazert, E., Zeuge, U., Frese, M., and Bartenschlager, R. Reconstructing signaling pathways from RNAi data using probabilistic Boolean threshold networks. *Bioinformatics 25*, 17 (Sep 2009), 2229–2235.

[48] Kim, H. G., Choi, S. K., and Lee, H. M. New algorithm in the basin hopping Monte Carlo to find the global minimum structure of unary and binary metallic nanoclusters. *J Chem Phys 128*, 14 (Apr 2008), 144702.

[49] Kim, Y. J., Björklund, S., Li, Y., Sayre, M. H., and Kornberg, R. D. A multiprotein mediator of transcriptional activation and its interaction with the C-terminal repeat domain of RNA polymerase II. *Cell 77*, 4 (May 1994), 599–608.

[50] Klipp, E., Liebermeister, W., Wierling, C., Kowald, A., Lehrach, H., and Herwig, R. *Systems Biology: A Textbook*. Wiley, 2009.

[51] Koleske, A. J., and Young, R. A. An RNA polymerase II holoenzyme responsive to activators. *Nature 368*, 6470 (Mar 1994), 466–469.

[52] Koonin, E. V., Wolf, Y. I., and Aravind, L. Prediction of the archaeal exosome and its connections with the proteasome and the translation and transcription machineries by a comparative-genomic approach. *Genome Res 11*, 2 (Feb 2001), 240–252.

[53] KORNBERG, R. D. Mediator and the mechanism of transcriptional activation. *Trends Biochem Sci 30*, 5 (May 2005), 235–239.

[54] KOSCHUBS, T. *Structure and functional architecture of the Mediator middle module from budding yeast.* PhD thesis, Ludwig-Maximilians-University of Munich (LMU), 2010.

[55] KOSCHUBS, T., LORENZEN, K., BAUMLI, S., SANDSTRÖM, S., HECK, A. J. R., AND CRAMER, P. Preparation and topology of the Mediator middle module. *Nucleic Acids Res 38*, 10 (Jun 2010), 3186–3195.

[56] KOSCHUBS, T., SEIZL, M., LARIVIÈRE, L., KURTH, F., BAUMLI, S., MARTIN, D. E., AND CRAMER, P. Identification, structure, and functional requirement of the Mediator submodule Med7N/31. *EMBO J 28*, 1 (Jan 2009), 69–80.

[57] KOSHLAND, D. E. The application and usefulness of the ratio k(cat)/K(M). *Bioorg Chem 30*, 3 (Jun 2002), 211–213.

[58] LARIVIÈRE, L., SEIZL, M., AND CRAMER, P. A structural perspective on Mediator function. *Curr Opin Cell Biol* (Feb 2012).

[59] LARIVIÈRE, L., SEIZL, M., VAN WAGENINGEN, S., RÖTHER, S., VAN DE PASCH, L., FELDMANN, H., STRÄSSER, K., HAHN, S., HOLSTEGE, F. C. P., AND CRAMER, P. Structure-system correlation identifies a gene regulatory Mediator submodule. *Genes Dev 22*, 7 (Apr 2008), 872–877.

[60] LAURITZEN, S. *Graphical models.* Oxford science publications. Clarendon Press, 1996.

[61] LEBRETON, A., TOMECKI, R., DZIEMBOWSKI, A., AND SÉRAPHIN, B. Endonucleolytic RNA cleavage by a eukaryotic exosome. *Nature 456*, 7224 (Dec 2008), 993–996.

[62] LI, Z., AND SCHERAGA, H. A. Monte Carlo-minimization approach to the multiple-minima problem in protein folding. *Proc Natl Acad Sci U S A 84*, 19 (Oct 1987), 6611–6615.

[63] LOEFFLER, M., AND ROEDER, I. Tissue stem cells: definition, plasticity, heterogeneity, self-organization and models–a conceptual approach. *Cells Tissues Organs 171*, 1 (2002), 8–26.

[64] MACISAAC, K. D., WANG, T., GORDON, D. B., GIFFORD, D. K., STORMO, G. D., AND FRAENKEL, E. An improved map of conserved regulatory sites for Saccharomyces cerevisiae. *BMC Bioinformatics 7* (2006), 113.

[65] MACKAY, D. *Information theory, inference, and learning algorithms.* Cambridge University Press, 2003.

[66] Marjoram, P., Molitor, J., Plagnol, V., and Tavare, S. Markov chain Monte Carlo without likelihoods. *Proc Natl Acad Sci U S A 100*, 26 (Dec 2003), 15324–15328.

[67] Markowetz, F., Bloch, J., and Spang, R. Non-transcriptional pathway features reconstructed from secondary effects of RNA interference. *Bioinformatics 21*, 21 (Nov 2005), 4026–4032.

[68] Markowetz, F., Kostka, D., Troyanskaya, O. G., and Spang, R. Nested effects models for high-dimensional phenotyping screens. *Bioinformatics 23*, 13 (Jul 2007), i305–i312.

[69] The MathWorks Inc. *Matlab version r2009b*. Natick, Massachusetts, 2009.

[70] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics 21* (1953), 1087–1092.

[71] Midtgaard, S. F., Assenholt, J., Jonstrup, A. T., Van, L. B., Jensen, T. H., and Brodersen, D. E. Structure of the nuclear exosome component Rrp6p reveals an interplay between the active site and the HRDC domain. *Proc Natl Acad Sci U S A 103*, 32 (Aug 2006), 11898–11903.

[72] Minka, T. P. Expectation-Maximization as lower bound maximization, 1998.

[73] Mitchell, P., Petfalski, E., Shevchenko, A., Mann, M., and Tollervey, D. The exosome: a conserved eukaryotic RNA processing complex containing multiple 3'–>5' exoribonucleases. *Cell 91*, 4 (Nov 1997), 457–466.

[74] Morrison, S. J., Shah, N. M., and Anderson, D. J. Regulatory mechanisms in stem cell biology. *Cell 88*, 3 (Feb 1997), 287–298.

[75] Neal, R. M. Sampling from multimodal distributions using tempered transitions. *Statistics and Computing 6* (1996), 353–366.

[76] Neal, R. M., and Hinton, G. E. A view of the EM algorithm that justifies incremental sparse and other variants. In *Learning in Graphical Models*, M. Jordan, Ed. MIT Press, 1998, pp. 355–370.

[77] Niederberger, T., Etzold, S., Lidschreiber, M., Maier, K. C., Martin, D. E., Fröhlich, H., Cramer, P., and Tresch, A. MC EMiNEM Maps the Interaction Landscape of the Mediator. *PLoS Computational Biology* (2012), In press.

[78] Niederberger, T., Hartung, S., Hopfner, K.-P., and Tresch, A. Processive RNA decay by the exosome: merits of a quantitative Bayesian sampling approach. *RNA Biol 8*, 1 (2011), 55–60.

[79] R FOUNDATION FOR STATISTICAL COMPUTING. *R: A Language and Environment for Statistical Computing.* Vienna, Austria, 2011.

[80] RIEGER, M. A., HOPPE, P. S., SMEJKAL, B. M., EITELHUBER, A. C., AND SCHROEDER, T. Hematopoietic cytokines can instruct lineage choice. *Science 325*, 5937 (Jul 2009), 217–218.

[81] ROBB, L. Cytokine receptors and hematopoietic differentiation. *Oncogene 26*, 47 (Oct 2007), 6715–6723.

[82] ROCKE, D. M., AND DURBIN, B. A model for measurement error for gene expression arrays. *J Comput Biol 8*, 6 (2001), 557–569.

[83] ROEDER, I., AND LOEFFLER, M. A novel dynamic model of hematopoietic stem cell organization based on the concept of within-tissue plasticity. *Exp Hematol 30*, 8 (Aug 2002), 853–861.

[84] SARRAZIN, S., AND SIEWEKE, M. Integration of cytokine and transcription factor signals in hematopoietic stem cell commitment. *Semin Immunol 23*, 5 (Oct 2011), 326–334.

[85] SCHAEFFER, D., TSANOVA, B., BARBAS, A., REIS, F. P., DASTIDAR, E. G., SANCHEZ-ROTUNNO, M., ARRAIANO, C. M., AND VAN HOOF, A. The exosome contains domains with specific endoribonuclease, exoribonuclease and cytoplasmic mRNA decay activities. *Nat Struct Mol Biol 16*, 1 (Jan 2009), 56–62.

[86] SCHMID, M., AND JENSEN, T. H. The exosome: a multipurpose RNA-decay machine. *Trends Biochem Sci 33*, 10 (Oct 2008), 501–510.

[87] SCHNEIDER, C., LEUNG, E., BROWN, J., AND TOLLERVEY, D. The N-terminal PIN domain of the exosome subunit Rrp44 harbors endonuclease activity and tethers Rrp44 to the yeast core exosome. *Nucleic Acids Res 37*, 4 (Mar 2009), 1127–1140.

[88] SEGAL, E., FRIEDMAN, N., KAMINSKI, N., REGEV, A., AND KOLLER, D. From signatures to models: understanding cancer using microarrays. *Nat Genet 37 Suppl* (Jun 2005), S38–S45.

[89] SEGAL, E., SHAPIRA, M., REGEV, A., PE'ER, D., BOTSTEIN, D., KOLLER, D., AND FRIEDMAN, N. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat Genet 34*, 2 (Jun 2003), 166–176.

[90] SMINCHISESCU, C., WELLING, M., AND HINTON, G. A Mode-Hopping MCMC sampler. Tech. rep., Department of Computer Science, University of Toronto, 2003.

[91] SMYTH, G. K. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Stat Appl Genet Mol Biol 3* (2004), Article3.

[92] SMYTH, G. K. Limma: linear models for microarray data. In *Bioinformatics and Computational Biology Solutions using R and Bioconductor*, R. Gentleman, V. Carey, S. Dudoit, and W. H. R. Irizarry, Eds. Springer, New York, 2005, pp. 397–420.

[93] SOUTOURINA, J., WYDAU, S., AMBROISE, Y., BOSCHIERO, C., AND WERNER, M. Direct interaction of RNA polymerase II and mediator required for transcription in vivo. *Science 331*, 6023 (Mar 2011), 1451–1454.

[94] STAALS, R. H. J., BRONKHORST, A. W., SCHILDERS, G., SLOMOVIC, S., SCHUSTER, G., HECK, A. J. R., RAIJMAKERS, R., AND PRUIJN, G. J. M. Dis3-like 1: a novel exoribonuclease associated with the human exosome. *EMBO J 29*, 14 (Jul 2010), 2358–2367.

[95] STARK, C., BREITKREUTZ, B.-J., CHATR-ARYAMONTRI, A., BOUCHER, L., OUGHTRED, R., LIVSTONE, M. S., NIXON, J., AUKEN, K. V., WANG, X., SHI, X., REGULY, T., RUST, J. M., WINTER, A., DOLINSKI, K., AND TYERS, M. The BioGRID Interaction Database: 2011 update. *Nucleic Acids Res 39*, Database issue (Jan 2011), D698–D704.

[96] STARK, C., BREITKREUTZ, B.-J., REGULY, T., BOUCHER, L., BREITKREUTZ, A., AND TYERS, M. BioGRID: a general repository for interaction datasets. *Nucleic Acids Res 34*, Database issue (Jan 2006), D535–D539.

[97] STOREY, J. D., AND TIBSHIRANI, R. Statistical significance for genomewide studies. *Proc Natl Acad Sci U S A 100*, 16 (Aug 2003), 9440–9445.

[98] TAKAGI, Y., CALERO, G., KOMORI, H., BROWN, J. A., EHRENSBERGER, A. H., HUDMON, A., ASTURIAS, F., AND KORNBERG, R. D. Head module control of mediator interactions. *Mol Cell 23*, 3 (Aug 2006), 355–364.

[99] TAKAGI, Y., AND KORNBERG, R. D. Mediator as a general transcription factor. *J Biol Chem 281*, 1 (Jan 2006), 80–89.

[100] TOMECKI, R., KRISTIANSEN, M. S., LYKKE-ANDERSEN, S., CHLEBOWSKI, A., LARSEN, K. M., SZCZESNY, R. J., DRAZKOWSKA, K., PASTULA, A., ANDERSEN, J. S., STEPIEN, P. P., DZIEMBOWSKI, A., AND JENSEN, T. H. The human core exosome interacts with differentially localized processive RNases: hDIS3 and hDIS3L. *EMBO J 29*, 14 (Jul 2010), 2342–2357.

[101] TRESCH, A. *Nessy: NESted effects models for SYstems biology*, 2007. R package version 1.0.

[102] TRESCH, A., AND MARKOWETZ, F. Structure learning in Nested Effects Models. *Stat Appl Genet Mol Biol 7*, 1 (2008), Article9.

[103] USKAT, D. Statistische Analyse zellulärer Entscheidungsprozesse: Ein Faktorgraph-Modell der Differenzierung hämatopoetischer Stammzellen, 2010.

[104] VAN DE PEPPEL, J., KETTELARIJ, N., VAN BAKEL, H., KOCKELKORN, T. T. J. P., VAN LEENEN, D., AND HOLSTEGE, F. C. P. Mediator expression profiling epistasis reveals a signal transduction pathway with antagonistic submodules and highly specific downstream targets. *Mol Cell 19*, 4 (Aug 2005), 511–522.

[105] VASKE, C. J., HOUSE, C., LUU, T., FRANK, B., YEANG, C.-H., LEE, N. H., AND STUART, J. M. A factor graph nested effects model to identify networks from genetic perturbations. *PLoS Comput Biol 5*, 1 (Jan 2009), e1000274.

[106] WALES, D. J., AND DOYE, J. P. K. Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms. *J Phys Chem 101* (1997), 5111–5116.

[107] WASSERMAN, L. *All of Statistics: A Concise Course in Statistical Inference.* Texts in Statistics. Springer, 2004.

[108] 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., VÉRONNEAU, 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. Functional characterization of the S. cerevisiae genome by gene deletion and parallel analysis. *Science 285*, 5429 (Aug 1999), 901–906.

[109] YEANG, C.-H., MAK, H. C., MCCUINE, S., WORKMAN, C., JAAKKOLA, T., AND IDEKER, T. Validation and refinement of gene-regulatory pathways on a network of physical interactions. *Genome Biol 6*, 7 (2005), R62.

[110] ZELLER, C., FRÖHLICH, H., AND TRESCH, A. A bayesian network view on nested effects models. *EURASIP J Bioinform Syst Biol 2009*, 1 (2008), 195272.

[111] ZHANG, F., SUMIBCAY, L., HINNEBUSCH, A. G., AND SWANSON, M. J. A triad of subunits from the Gal11/tail domain of Srb mediator is an in vivo target of transcriptional activator Gcn4p. *Mol Cell Biol 24*, 15 (Aug 2004), 6871–6886.