An efficient strategy to infer biochemical networks by means of statistical calibration of mechanistic models

Juho Timonen

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 23rd October, 2017

Thesis supervisor:

Prof. Harri Lähdesmäki

Thesis advisor:

D.Sc. (Tech.) Jukka Intosalmi

Aalto University
School of Science

AALTO UNIVERSITY
SCHOOL OF SCIENCE

ABSTRACT OF THE
MASTER'S THESIS

Author: Juho Timonen

Title: An efficient strategy to infer biochemical networks by means of statistical calibration of mechanistic models

Date: 23rd October, 2017 Language: English Number of pages: 8+61

Degree programme: Mathematics and Operations Research

Major: Applied Mathematics Code: SCI3053

Supervisor: Prof. Harri Lähdesmäki

Advisor: D.Sc. (Tech.) Jukka Intosalmi

Various fields of science employ systems of ordinary differential equations (ODEs) to model the behaviour of dynamical systems, such as gene regulatory networks. However, the system model often contains uncertainty in both its structure and the model parameters. When experimental data are available, the model parameters can be calibrated using well-established statistical techniques and also different model structures can be compared in the light of their statistical evidence. If the set of alternative model structures is small enough, it is possible to evaluate the validity of each individual model separately. However, for biochemical networks, the number of viable model configurations is often enormous, which renders it computationally impossible to draw inferences about the network structure using such an exhaustive strategy. This thesis introduces a novel computationally efficient approach to obtain probabilistic structure inferences for general ODE models. The proposed approach relies on exploring the discrete set of alternative models using Markov chain Monte Carlo methods. Inference problems involving simulated data are used to demonstrate that the method is suitable for efficiently extracting information about the characteristics of the likely models. Furthermore, the method is applied to infer the structure of the transiently evolving core regulatory network that steers the T helper 17 (Th17) cell differentiation. The obtained results are in agreement with earlier studies that suggest that the Th17 differentiation program involves three sequential phases.

Keywords: Ordinary differential equation models, Bayesian inference, Markov chain Monte Carlo, gene regulatory networks, T helper cells

AALTO-YLIOPISTO
PERUSTIETEIDEN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Juho Timonen

Työn nimi: Tehokas strategia biokemiallisten verkkojen päättelyyn mekanististen mallien tilastollisen sovittamisen avulla

Päivämäärä: 12.10.2017 Kieli: Englanti Sivumäärä: 8+61

Koulutusohjelma: Mathematics and Operations Research

Pääaine: Sovellettu matematiikka Koodi: SCI3053

Työn valvoja: Prof. Harri Lähdesmäki

Työn ohjaaja: TkT Jukka Intosalmi

Differentiaaliyhtälösysteemejä käytetään monilla tieteenaloilla mallintamaan dynaamisia systeemejä, kuten geenisäätelyverkkoja. Systeemiä kuvaavassa mallissa on kuitenkin usein epävarmuutta sekä sen rakenteen että mallin parametrien osalta. Kun kokeellista dataa on saatavilla, mallien parametrit voidaan sovittaa käyttäen vakiintuneita tilastollisia menetelmiä, ja myös erilaisia malleja voidaan vertailla niiden tilastollisen todennäköisyyden avulla. Jos vaihtoehtoisia malleja on vain vähän, voidaan jokainen yksittäinen malli validoida erikseen. Biokemiallisten verkkojen tapauksessa mahdollisia mallikonfiguraatioita on usein lukemattomia, minkä takia yllä kuvattu tapa verkkojen rakenteen päättelyyn on laskennallisesti mahdotonta. Tässä työssä esitellään uusi laskennallisesti tehokas lähestymistapa tehdä probabilistisia päätelmiä differentiaaliyhtälömallien rakenteesta. Ehdotettu lähestymistapa perustuu diskreetin mallijoukon tutkimiseen Markov Chain Monte Carlo -menetelmillä. Työssä muotoillaan simuloituun dataan liittyviä ongelmia, joilla näytetään, että menetelmällä voi tehokkaasti saada tietoa todennäköisimmistä mallirakenteista. Menetelmää sovelletaan myös erään auttaja-T-solujen alityypin (Th17) erilaistumista ajavan aikariippuvan ydinverkon rakenteen päättelyyn. Saadut tulokset ovat linjassa aiempien tutkimusten kanssa, joiden mukaan Th17-solujen erilaistuminen tapahtuu kolmessa peräkkäisessä vaiheessa.

Avainsanat: Differentiaaliyhtälömallit, Bayesiläinen tilastotiede, Markov chain Monte Carlo -menetelmät, geenisäätelyverkot, auttaja-T-solut

Preface

I wish to thank Prof. Harri Lähdesmäki and Dr. Jukka Intosalmi for providing me with the fascinating scientific problem considered in this study. Jukka has been the key person in instructing both the computations and the writing of this thesis, and he has taught me a lot about Bayesian inference, mathematical modeling of gene regulation and optimization. Harri is to thank for guiding the direction of this thesis and providing solutions to fundamental questions that have arisen. Henrik Mannerström and Markus Heinonen also deserve credit for some helpful discussions and comments. In addition, I thank the whole group of Computational systems biology for pleasant company. Other people who have been helpful at the workplace include Iskander Zharkenov, Linh van Nguyen and Pradeep Eranti.

I am thanking Guild of Physics for introducing me to the university life at the Otaniemi campus. Furthermore, thanks to my fellow students who have made my university time memorable. These include for instance the people who sat in the board of the Guild of Physics during years 2014-15.

The computational resources utilized in this study were provided by the Aalto Science-IT project and CSC – IT Center for Science, Finland.

Otaniemi, 23rd October, 2017

Juho Timonen

Contents

Symbols and abbreviations

Symbols

$s \in S$	s is a member of set S
$X \subset S$	set X is a subset of S
$x \sim p$	random varible x has the distribution p
e	Euler's number
$\mathbb{N}$	set of positive integers
$\mathbb{R}^n$	n-dimensional space of real numbers
$N\left(\cdot \mid \mu, \sigma^2\right)$	probability density function of the one-dimensional normal distribution with mean μ and variance σ^2
$N_d\left(\cdot \mid \mu, \mathbf{\Sigma}\right)$	probability density function of the d-dimensional normal distribution with mean vector μ and covariance matrix $\mathbf{\Sigma}$
Z	binary matrix defining a LEM model configuration
$\{M\}_{ij}$	element on the ith row and jth column of matrix M
$p(\cdot)$	(marginal) probability distribution
$p(\cdot \mid x)$	probability distribution conditional on x
θ	vector of model parameters
$\mathcal{D}$	data

Operators

$\mathrm{A} \times \mathrm{B}$	Cartesian product of sets A and B
$\dfrac{\mathrm{d}}{\mathrm{d}t}$	derivative with respect to variable t
$\sum_i$	sum over index i
$\sum_{M \in \mathcal{M}}$	sum over set $\mathcal{M}$
$\prod_i$	product over index i
$\|\mathbf{v}\|$	Euclidean norm of the vector $\mathbf{v}$
$\|A\|_F$	Frobenius norm of matrix A
$d(Z_1, Z_2)$	distance between models Z_1 and Z_2
$\mathcal{N}_k(Z)$	k-neighborhood of model Z
log	natural logarithm
$[G]$	concentration or abundance of a molecular species G

Abbreviations

AIC	Akaike information criterion
BATF	Basic leucine zipper ATF-like transcription factor
BDF	Backward differentiation formula
BIC	Bayesian information criterion
CD4	Cluster of differentiation 4
DNA	Deoxyribonucleic acid
FPKM	Fragments per kilobase per million reads
IVP	Initial value problem
IL	Interleukin
IRF4	Interferon regulatory factor 4
LEM	Latent effect mechanistic
LMM	Linear multistep method
MCMC	Markov chain Monte Carlo
mRNA	Messenger RNA
ODE	Ordinary differential equation
OVL	Overlapping coefficient
RORγt (RORC)	Retinoic acid receptor-related orphan receptor gamma t (RORC gene)
RNA	Ribonucleic acid
STAT3	Signal transducer and activator of transcription 3
TGFβ	Transforming growth factor β
TF	Transcription factor
Th17	T helper 17

1 Introduction

Nonlinear ordinary differential equations (ODEs) are commonly used in systems biology to model biochemical networks, such as signaling pathways or gene regulatory networks. This is because ODE modeling provides a highly expressive mathematical framework that can be used to describe the transient behaviour of the system. These models are often mechanistic, meaning that the system components and mechanisms have meaningful real-life interpretations. Therefore, ODE modeling facilitates deep understanding of the system mechanisms in a way that cannot necessarily be achieved by just describing statistical dependencies in the data. However, when ODE models of biochemical reaction networks are constructed, there is often uncertainty in both the model parameters and the model structure [25]. This uncertainty typically brings about a large number of alternative well-motivated, hypothetical models that depict the underlying biochemical network. This goal of this thesis is to develop a framework that facilitates inferring the network structure in such situations. The proposed approach can be applied to ODE model structure inference problems emerging in other fields of science, too.

In general, parameters of an ODE model can be calibrated using statistical techniques if a sufficient amount of data is available (see e.g. [25], [50], [51]). Whereas the parameter estimation problem is rather well studied, the problem of inferring the model structure, given a large set of viable model configurations, remains unsolved in practice. The number of the alternative models can span from two to hundreds of thousands, and therefore the need for reliable, automatized, and computationally efficient strategies to infer the most likely model structures is urgent.

ODE models can be ranked, for instance, by using different kinds of information criteria that estimate the models' predictive accuracy, such as the Akaike information criterion (AIC) [1]. Alternatively, this can be done using cross-validation, where only a subset of the data are used to fit the ODE model and the the remaining data are used to evaluate the model's predictive performance [20]. Because the kinetic rate parameters of biochemical models are strictly positive by definition, continuous model expansion approaches have to be ruled out.

A completely different approach is to assess the statistical evidence for alternative models within the Bayesian framework [61]. This approach has been successfully

used in several studies (see e.g. [11], [33], [34], [64]). In this thesis, the parameter estimation and model ranking problems are formulated within the Bayesian framework, which allows us to make use of the discrete posterior probability distribution over all alternative model structures. This framework allows us to obtain probability statements that describe the uncertainty related to different model mechanisms, instead of searching for a point estimate in the model space. In general, the full discrete posterior distribution over different models can be obtained by marginalizing out the model parameters. The drawback of the Bayesian approach is that these computations can be very expensive. To overcome this issue, we approximate the marginalized likelihoods via the Bayesian information criterion (BIC) [57]. Even though this allows computationally a relatively light marginal likelihood approximation, we often have an enormous amount of viable models, meaning that computing the BIC for each of them is not feasible.

The abundance of possible model configurations gives rise to a need for a method that finds the good models by exploring the model space cleverly. For this purpose, we propose using discrete space Markov Chain Monte Carlo (MCMC) methods and provide an implementation of a Metropolis-type algorithm with a simple yet efficient proposal distribution. The performance of the new strategy is demonstrated using toy problems that involve realistically simulated noisy data and a rather restricted set of possible models. Using these test cases, we show that our MCMC-based approach can be used to accurately infer the true data-generating network mechanisms even when only a small fraction of all possible models are evaluated.

The proposed strategy is also applied to infer the transiently evolving core molecular network that steers the T helper 17 (Th17) cell differentiation. To capture the rewiring effects during the differentiation process, we utilize the recently developed latent effect mechanistic (LEM) modeling approach [34]. In this study, we also extend the LEM modeling approach to allow rigorous statistical testing about the type of the latent process. In our Th17 cell differentiation application, this extension enables us to test hypotheses about how many sequential phases are involved in the differentiation processes. In addition, the novel approach also enables us to obtain probabilistic predictions on the molecular interactions that are active in different phases of Th17 cell differentiation. For some of the models that are found to best describe the Th17 network, we perform further analyses using the profile likelihood approach [40], which allows us to assess the uncertainty of the parameter estimates and identifiability of the parameters.

This thesis is structured as follows. Section 2 introduces ODE modeling of biochemical systems, with particular emphasis on modeling of gene regulatory networks. Theoretical background behind numerically solving ODE systems, especially stiff ones, is presented in Section 3, whereas Section 4 focuses on the data-driven statistical machinery that is essential for performing model structure inferences. The novel model space exploration algorithm is formulated and presented along with its computational implementation in Section 5. Section 6 displays the results of both the simulated data experiments and the Th17 application. Finally, Section 7 summarizes the thesis with discussion about the obtained results.

2 Mathematical modeling of biochemical systems

Mathematical modeling offers rigorous means to quantify interactions in molecular cell biology and has become an important tool as both data collection methods and computational capacities have evolved [31]. In many applications of systems biology, mathematical modeling is necessary, since it facilitates testing hypotheses about biological systems substantially more cheaply and quickly than experimental validation. However, in order to employ mathematical tools, one must be able to transform any initial hypotheses about the system in a well-defined, quantitative form [2]. In this section we describe how to construct ordinary differential equation (ODE) models for complex biochemical systems, such as gene regulatory networks. The 2013 book by Ingalls [31] is used as the primary source. Furthermore, a recently developed formalism for dynamically evolving ODE systems, called the LEM [34] model, is introduced.

2.1 Mechanistic ODE modeling

Models that endeavour to mimic the actual molecular mechanisms so that model variables have counterparts in real life are called mechanistic. However, even such models are always abstractions of reality, and thus it is important to understand the assumptions that one is making when a model is built. An especially expressive and commonly used mathematical framework to mechanistically model dynamic systems is provided by ordinary differential equations (ODEs) [25].

The construction of an ODE model is usually started from the elementary level of chemical reactions between different molecular species. A well-known principle to model a chemical reaction is the law of mass action. It states that the rate at which a reaction occurs is proportional to the product of the reactant concentrations. On the molecular level, this relies on the assumption that more collisions between the reactants happen as the abundance of the reactants grows. In order to use the law of mass action, one should therefore assume that the reactants are well-stirred. Furthermore, there should be a large amount of each molecular species present, so

that modeling concentrations as continuous variables is reasonable.

Formulating the law of mass action mathematically leads us to ODEs. For instance, we can consider the reaction

$$\mathrm{R}_1 + \mathrm{R}_2 \longrightarrow \mathrm{P}, \tag{1}$$

where R_1, R_2 are reactants and P is the product of the reaction. The law of mass action states that the rate at which the concentration of P grows can be expressed mathematically as the time derivative

$$\frac{\mathrm{d}[\mathrm{P}]}{\mathrm{d}t} = k[\mathrm{R}_1][\mathrm{R}_2], \tag{2}$$

where k is a rate constant. Here [A] is used to denote the concentration of a molecular species A, but generally it can be any physical quantity that is proportional to abundance of A.

The law of mass action is the basis for other common reaction rate laws, such as Michaelis–Menten kinetics, Hill kinetics and the generalized mass action rate law. All the above approaches rely on defining the time derivative of different concentrations, and consequently, we end up with a mathematical formulation consisting of ODEs. An ODE formulation can in principle be derived for any set of reactions, if the system is assumed to be well-stirred. A more accurate way to model a system of molecular species would be to simulate the positions and velocities of all individual molecules, and change the molecular populations appropriately if their collision results in a reaction [24]. A detailed motivation for such modeling and discussion about the assumptions that are needed to turn this model into a set of reaction rate equations, i.e. ODEs, can be found in [23] and [24].

2.2 Gene regulatory networks

In this thesis, the main focus is on modeling gene regulatory networks, and we begin by introducing the basic concepts related to gene expression. These concepts are presented as they are given in [2]. The hereditary information of cells is stored by a large macromolecule called deoxyribonucleic acid (DNA). A specific part of this sequence is called a gene and information of DNA is read in a process called gene expression. This fundamental process produces proteins, which are large molecules that are responsible for most of the cellular functions. Gene expression consists of two states, transcription and translation, both of which are complex processes involving a variety of biochemical reactions. These production mechanisms are illustrated in Figure 1. In transcription, a segment of DNA is copied into a macromolecule called ribonucleic acid (RNA) and in translation, RNA molecules are then used as templates to synthesize a protein. This flow of genetic information from DNA to RNA and from RNA to protein occurs in all living cells. Some proteins control the transcription of other genes by binding into the DNA sequence at a regulatory site of the target gene. Such regulators are called transcription factors (TF), and a system where gene products regulate the rate of each others' production is called a gene regulatory network.

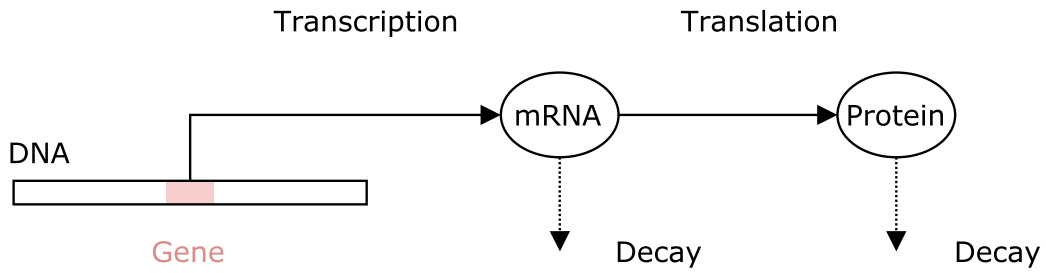


Figure 1: Illustration of the phases involved in gene expression. In transcription, a gene, i.e. a segment of a DNA strand, is read to produce messenger RNA (mRNA). Its information is used in translation to produce a protein. Both RNA and protein molecules also degrade during the process. The drawing has been inspired by Figure 7.1 in [31].

In Section 2.1 we presented how the law of mass action can be used when modeling chemical reactions. Similar approach can be applied to model gene expression mathematically, which we demonstrated here, returning to the presentation of [31]. Since transcription and translation are complex processes, modeling gene expression using the law of mass action involves many simplifications. Furthermore, gene expression in a single cell can involve only very few molecules, and thus representing abundances as continuous concentrations is problematic. However, in a large population of cells, we can model gene expression using a mass action formalism where we interpret the differential equations as descriptions of the average behavior in the population.

Let us consider a network of n genes and denote their time-varying abundances by $y_1, \ldots, y_n$. These abundances are averages over a population of cells. In this study, the value y_i refers to the mRNA level of the corresponding gene i, since the experimental data used in this thesis are mRNA reads. Possible mechanisms in the network are for example regulated and unregulated expression of the genes, as well as degradation. Unregulated basal expression of gene i can be modeled simply as

$$\frac{\mathrm{d}y_i}{\mathrm{d}t} = k_i^{\mathrm{b}}, \tag{3}$$

if it is assumed to have a constant rate k_i^{b}, that captures all the reactions involved in transcription and translation. Unregulated expression is considered happening independent of the gene itself, as opposed to degradation which of course can only happen if the gene is expressed in the cell. Degradation can be modeled as exponential decay, where the rate is dependent of the abundance of the gene i itself, and is given by

$$\frac{\mathrm{d}y_i}{\mathrm{d}t} = -k_i^{\mathrm{d}} y_i, \tag{4}$$

where k_i^{d} is the degradation rate of gene i. If another gene j enhances the expression

of gene i, expression of gene i can be modeled by the rate equation

$$\frac{\mathrm{d}y_i}{\mathrm{d}t} = k_{ij}^{\mathrm{act}} y_j, \tag{5}$$

where the activation rate constant k_{ij}^{act} captures the strength of this mechanism. Another type of regulation is repressible expression, where a gene j expresses a product that inhibits the expression of gene j. Rate of change in concentration of gene i is then given by the equation

$$\frac{\mathrm{d}y_i}{\mathrm{d}t} = -k_{ij}^{\mathrm{inh}} y_i y_j, \tag{6}$$

where k_{ij}^{inh} is the corresponding inhibition rate. Two gene products j and k can also act cooperatively to upregulate another gene i. Concentration of gene i then changes at rate

$$\frac{\mathrm{d}y_i}{\mathrm{d}t} = k_{ijk}^{\mathrm{sact}} y_j y_k, \tag{7}$$

where k_{ijk}^{sact} is the rate constant of this synergistic activation. Gene regulatory networks can also involve other mechanisms, but this study only involves ODE models comprised of basal activation, degradation, induced activation, inhibition and synergistic activation. Note that it is also possible for a gene to activate or inhibit itself. Autoactivation or autoinhibition can be modeled by setting $i = j$ in Equation 5 or 6, respectively.

If all mechanisms involved in a regulatory network are known, one can count the total rate equations for each gene by adding the components from different mechanisms that affect its expression. For example, let us consider the regulatory network of five genes A, B, C, D and E (Figure 2). The abundance of gene A now has a time derivative that is obtained by combining all the mechanisms that affect its expression. Since A is activated by E and inhibited by B and D, we get the total rate

$$\frac{\mathrm{d[A]}}{\mathrm{d}t} = k_{AE}^{\mathrm{act}}[\mathrm{E}] - k_{AB}^{\mathrm{inh}}[\mathrm{A}][\mathrm{B}] - k_{AD}^{\mathrm{inh}}[\mathrm{A}][\mathrm{D}], \tag{8}$$

where k_{AE}^{act}, k_{AB}^{inh} and k_{AD}^{inh} are the rate constants of the regulatory mechanisms and $[\cdot]$ denotes abundance (mRNA reads). Because abundances of the activating and inhibiting genes themselves are time-dependent, analyzing the behaviour of systems such as the one in Figure 7 is impossible without quantitative mathematical modeling, especially if the network contains feedback loops.

More generally, a mathematical description of an entire regulatory network results in a system of ODEs

$$\frac{\mathrm{d}y_i}{\mathrm{d}t} = f_i\left(t, \mathbf{y}(t), \theta\right), \tag{9}$$

where θ is a vector of all rate constants and other possible parameters and $\mathbf{y}(\mathrm{t}) = [y_1, \ldots, y_n]^{\mathrm{T}}$. Now, assuming that the rate constants and an initial concentration of each gene are known, we can solve the system outputs $y_i(t)$ on some time interval of interest. Consequently, this modeling approach provides us with a rigorous and useful means to predict the network dynamics by computer simulation.

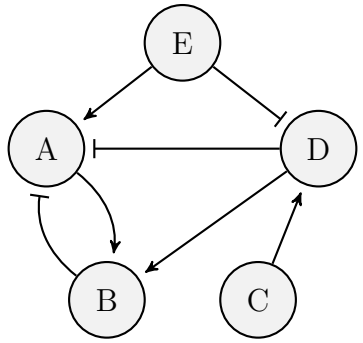


Figure 2: An example network of regulatory interactions between five genes A, B, C, D and E. Activation and inhibition mechanisms are represent using arrows ($\rightarrow$) and turnstiles ($\dashv$), respectively.

2.3 Latent effect mechanistic models

It is not always realistic to assume that a certain mechanism of a regulatory network is active at the same strength during the whole time course under investigation. Thus we might want to include time-variation in the corresponding mathematical model. Such approach is the latent effect mechanistic (LEM) model [34]. The LEM formalism is briefly presented here referring to the original article [34], where a more detailed description can be found.

The LEM model couples a standard ODE system with a latent process, that describes the time evolution of the network components. Such modeling might be needed for example when a certain part of DNA is initially covered with epigenetic marks that prevent a transcription factor A from binding a region called the promoter and the gene B is not expressed. As time passes, enzymatic signals can remove these marks and eventually the promoter region becomes clear, allowing the transcription factor to bind the promoter, and expression of the gene to begin. This is illustrated in Figure 3.

In example case shown in Figure 3, the transcription factor A could be a product of gene A that is in a regulatory network with gene B. Now, assume that we wish to include this transient behaviour of the activation mechanism where A activates B into an ODE model for the rate $\frac{\mathrm{d}[B]}{\mathrm{d}t}$, which is defined using the principles presented in the previous section. This can be done by multiplying the corresponding component $k_{\mathrm{AB}}^{\mathrm{act}}[A]$ of the rate equation by a weight function that has an initial value zero and eventually reaches one. This change can be a rapid switch-like step or a curve that rises smoothly. To make computations easier, one effectively wants to use a continuous and differentiable function. Smoothness is justified by the fact that the removal of the epigenetic marks probably happens at slightly different times in different cells, and thus the function represents the proportion of cells where the marks have been removed.

We now present a general LEM model with n different state variables $y_1, \ldots, y_n$ that can for example correspond to abundances of different genes in a regulatory network driving a cell differentiation program, and M distinct latent states that can

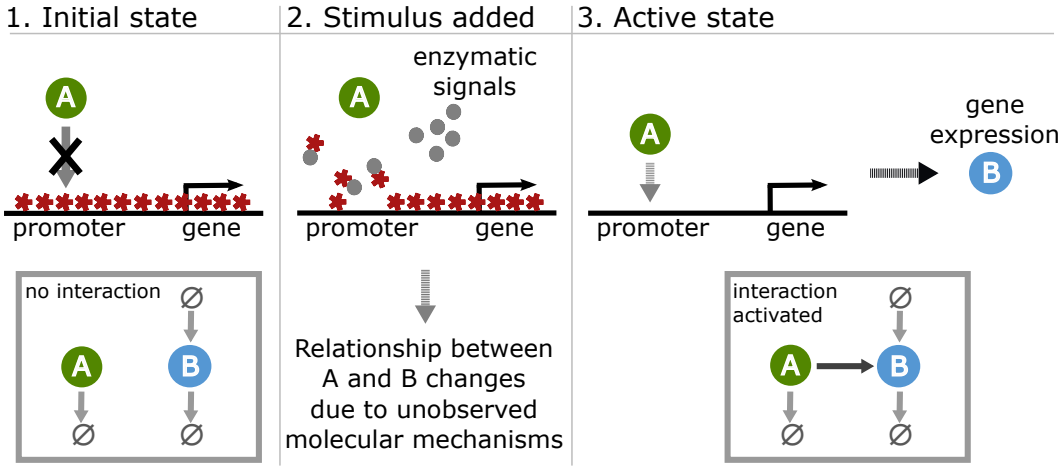


Figure 3: Illustration of a transient silencing mechanism that causes a state transition in expression of a gene. In the initial state the gene B is not expressed due to epigenetic marks that bind the promoter region. As these marks are removed, transcription factor A can bind the promoter and thus activate expression of gene product B. The figure is from [34].

correspond to distinct phases in the course of the cellular differentiation. The state of the system at time t is given by $y(t,\theta) : [0,T] \times \mathbb{R}^d \to \mathbb{R}^n$, where θ is a parameter vector and d is the number of parameters. An essential part of the LEM model is a latent process $x(t,\theta) : [0,T] \times \mathbb{R}^d \to \mathbb{R}^M$ that describes the time-evolution of the M different latent states as a function of time. The model parameters θ can therefore comprise for example reaction rates and parameters that adjust the latent process.

Assume that there exists a set of N different mechanisms that can be present in the network, and we have constructed functions $f_j(y,\theta)$, $j = 1,\ldots,N$ to capture them mathematically. An $N \times M$ matrix Z is used for storing the information about the network configuration of a specific model Z. The element $\{Z\}_{jk}$ determines if the mechanism corresponding to function f_j is active in the kth latent state or not (values 1 and 0, respectively). Different configurations of this matrix therefore correspond to alternative structures of the network.

After we have defined the viable network mechanisms and the latent process, a mathematical definition of the LEM model is then comprised by the ODE system

$$\frac{\mathrm{d}y_i}{\mathrm{d}t} = \sum_{j \in \mathcal{I}_i} f_j(y,\theta) w_j(t,x,Z,\theta), \qquad t \in [0,T] \tag{10}$$

where $\mathcal{I}_i \subset \{1,\ldots,N\}$ is the index set for which the function $f_j(y,\theta)$ is affecting the rate of change in y_i if and only if $j \in \mathcal{I}_i$. The state-dependent behavior of the functions f_j is conveyed from the jth row of the matrix Z to the final ODE system by the weight function

$$w_j(t,x,Z,\theta) = \frac{\sum_{k=1}^{M} \{Z\}_{jk} x_k(t,\theta)}{\sum_{k=1}^{M} x_k(t,\theta)}, \tag{11}$$

which takes values between 0 and 1. Figure 4 illustrates how a set of mechanisms and the configuration matrix Z define an example 3-phase network of three genes. Also an example design of a possible 3-phase latent process has been included in the figure.

To conclude, quantitative dynamic mathematical modeling is essential in order to reveal dynamic behaviour of complex systems of multiple molecular species and their regulatory mechanisms. ODE modeling offers a rigorous framework for this task and it is employed in this thesis. The ODE systems that model gene expression are constructed by condensing the reactions involved in transcription and translation into a single rate constant. This study focuses only on modeling gene regulatory networks where the gene abundances are averages over a population of cells. A special case of an ODE model is the LEM model, which captures dynamically evolving ODE structures.

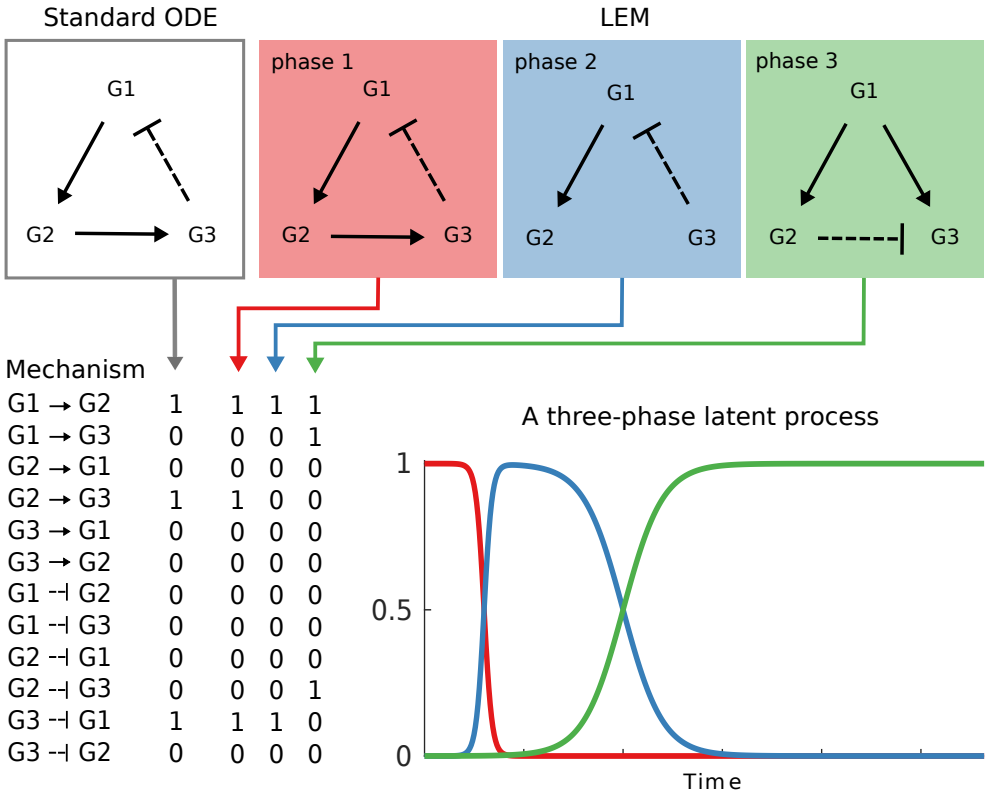


Figure 4: Illustration of a standard ODE and a three-phase LEM model. The three subnetworks associated with the three phases of the LEM model are drawn on red, blue and green backgrounds. The solid arrows represent activation links and the dashed turnstiles are inhibition links. Each row of the binary matrix corresponds to one mechanism and each column to one latent state, such that the element on row i and column j defines if mechanism i is present in phase j. An example design of a time-dependent latent process is illustrated with red, blue and green lines. Each component describes the strength of the mechanisms in the associated subnetwork as a function of time. The standard ODE model that corresponds to a stationary network structure can be defined with a single vector, since it is a LEM model with only one phase.

3 Numerical solution of ODE systems

Ordinary differential equations appear in various fields of science. In Section 2 we showed how to motivate ODE modeling of gene regulatory networks. However, mathematically expressing the rate at which abundances of molecular species change is usually beneficial only if it is possible to solve the output of the system, i.e. the abundances as a function of time. Even though differential equations like Equation 3 and 4 possess straightforward analytic solutions, they are often parts of an ODE system, where feedback of more complex mechanisms can cause analytically intractable dynamics. In fact, it turns out that practically all ODE systems in scientific applications are nonlinear, and thus numerical methods are needed to integrate them. This section focuses on how to efficiently compute a reliable approximate numerical solution for an ODE system, given some initial condition, i.e. the state of the system at the beginning of the time window of interest. The main focus is ultimately on stiff ODE systems. To provide some theoretical discussion, we present sufficient conditions for existence and uniqueness of a solution. Furthermore, some results regarding stability and order of different numerical methods are introduced to provide theoretical justifications for the methods used in this thesis.

3.1 Initial value problems

In general, a first-order initial value problem (IVP) consists of the ODE system

$$\frac{\mathrm{d}\mathbf{y}(t)}{\mathrm{d}t} = \mathbf{f}\left(t, \mathbf{y}(t)\right) \tag{12}$$

and an initial condition $\mathbf{y}(t_0) = \mathbf{y_0}$. Here $\mathbf{y} : [t_0, t_{\max}] \to \mathbb{R}^n$ is the state of the system and $\mathbf{f} : [t_0, t_{\max}] \times \mathbb{R}^n \to \mathbb{R}^n$ is the function that describes how the rate of change in the state depends on the system state and time t. We use $\frac{\mathrm{d}\mathbf{y}(t)}{\mathrm{d}t}$ to denote the vector that contains componentwise time derivatives $\mathbf{y}(t)$. Higher order problems involving a formula for the pth time derivative of $\mathbf{y}(t)$, where $p > 1$ can also be written as first-order ODE systems by defining additional equations (see [26], pages 12–13). Furthermore, this thesis only deals with first order IVPs, which is why we only focus on solving $\mathbf{y}(t)$ in Equation 12.

The question we begin the theoretical discussion with is whether a given IVP even possesses a unique solution. In order to present sufficient conditions for this, we begin from assumptions about the right-hand side of Equation 12. We say that the function $\mathbf{f} : [t_0, t_{\max}] \times \mathbb{R}^n \to \mathbb{R}^n$ satisfies the Lipschitz condition in the variable $\mathbf{y}$, if for all $t \in [t_0, t_{\max}]$ and $\mathbf{y}, \mathbf{y}^* \in \mathbb{R}^n$, we have

$$\|\mathbf{f}(t, \mathbf{y}) - \mathbf{f}(t, \mathbf{y}^*)\| \leq L \|\mathbf{y} - \mathbf{y}^*\| \tag{13}$$

for some constant $L > 0$ [8]. Here $\|\cdot\|$ denotes the Euclidean norm. Intuitively, this means that changes in the function are restricted by a large enough constant L, and all IVPs considered in Section 2 satisfy this property. Now, if we assume that $\mathbf{f}$ is continuous in the variable t and satisfies the Lipschitz condition in $\mathbf{y}$, then the IVP in Equation 12 has a unique solution $\mathbf{y} : [t_0, t_{\max}] \to \mathbb{R}^n$ [8]. For proof, see [8] page 23.

3.2 Numerical methods

Numerical methods that approximate the solution of an initial value problem in the interval $t \in [t_0, t_{\max}]$ involve computing the output $\mathbf{y}(t)$ at a grid of points $t_0, t_0 + h, t_0 + 2h, \ldots, t_{\max}$ sequentially starting from t_0 which is known [58]. Here h is called the step size, which is assumed to be a small constant. We now present the basics of different types of numerical methods for solving IVPs. For notational convenience, we present the methods for the case where the problem consists of only one equation

$$\frac{\mathrm{d}y(t)}{\mathrm{d}t} = f(t, y(t)) \tag{14}$$

and an initial condition $y(t_0) = y_0$. We note that all the numerical methods discussed in the following can also be applied to systems of ODEs, taking into account that one is then dealing with vectors [58]. In this section, we use y_i to denote a numerical approximation for $y(t_i)$ and exploit the shortened notation $f_i = f(t_i, y_i)$. This notation should not be confused with different state variables y_i and the functions f_j in Section 2.

When analyzing different methods, we utilize the concept of global error at t_i, which is given by $e_i = y(t_i) - y_i$ [26]. We say that a numerical method converges to the solution $y(t)$ of the IVP in Equation 14 at the point $t = t^*$ if the global error at $t_i = t^*$ satisfies

$$|e_i| \to 0, \tag{15}$$

when $h \to 0$ [26]. We are only interested in methods that satisfy this property, i.e. do not include systematic bias but instead are arbitrarily accurate when arbitrary computation power is available. In addition to convergence itself, we are interested in the rate of convergence of different methods. We say that the order of a method is p, if p is the largest integer for which there exists constants C and h_0, such that

$$|e_i| \leq Ch^p \tag{16}$$

for all $0 < h < h_0$ [26].

A simple method for sequentially obtaining the values $y_1, y_2, \ldots$, called Euler's method, proceeds by the iterative formula

$$y_{i+1} = y_i + h \cdot f_i, \tag{17}$$

i.e. moving to the direction of the derivative at the current point, until $t_{\max}$ is reached [8]. It can be proven that Euler's method is convergent (see [8], page 68). However, the method has bad stability properties and generally requires an impractically small step size in order to reach desired accuracy, as its order is one [8]. Thus we need to study more sophisticated methods for practical use.

Well-known families of numerical methods that can achieve a higher order than Euler's method include Taylor series methods, Runge–Kutta methods and linear multistep methods [26]. Taylor series methods demand computing higher order derivatives of the right-hand side of Equation 12, which in applications is often computationally too difficult [26]. Runge–Kutta methods improve accuracy by ways that require a rather expensive amount of evaluations of $\mathbf{f}(t, \mathbf{y}(\mathbf{t}))$ in Equation 12 [58]. This study focuses on and utilizes the family of linear multistep methods. A particularly important class of such methods work by backward differentiation.

3.3 Linear multistep methods

Euler's method uses the previously computed values of the solution and its derivative only from one previous iterate, which is why it is called a one-step method [26]. We now introduce the class of linear multistep methods (LMM), that can utilize this history more extensively. The general formula for obtaining the approximative solution sequentially with a k-step method is

$$y_{i+k} + \alpha_{k-1} y_{i+k-1} + \ldots + \alpha_0 y_n = h \cdot (\beta_k f_{i+k} + \beta_{k-1} f_{i+k-1} + \ldots + \beta_0 f_i)\,, \tag{18}$$

where $\beta_k = 0$ means that the method is explicit [26]. Otherwise the method is called implicit, because both sides of the equation depend on y_{i+k}. Note that in order to use a k-step method, we need to have additional starting values $y_{k-1}, y_{k-2}, \ldots, y_1$, which have to be computed first using some other method such as Euler's [26].

A simple example of an explicit multistep method is the early two-step method by Adams and Bashforth, here denoted AB(2), where $\alpha_1 = -1$, $\alpha_0 = 0$, $\beta_2 = 0$, $\beta_1 = 3$ and $\beta_0 = -1$ [26]. The iterative formula thus becomes

$$y_{i+2} = y_{i+1} + \frac{h}{2} \cdot (3 f_{i+1} - f_i)\,. \tag{19}$$

In order to demonstrate this LMM and compare it to Euler's method, let us consider an example IVP

$$\frac{\mathrm{d}y(t)}{\mathrm{d}t} = 1 + e^{-0.5t} - y(t)\ , \qquad y(0) = 0 \tag{20}$$

that has the analytic solution $y(t) = 1 - 3e^{-t} + 2e^{-0.5t}$. Here $y(t)$ could represent the abundance of a gene A, and the part $1 + e^{-0.5t}$ could be the output of another gene B that activates A. The part $-y(t)$ then corresponds to degradation of gene A. We

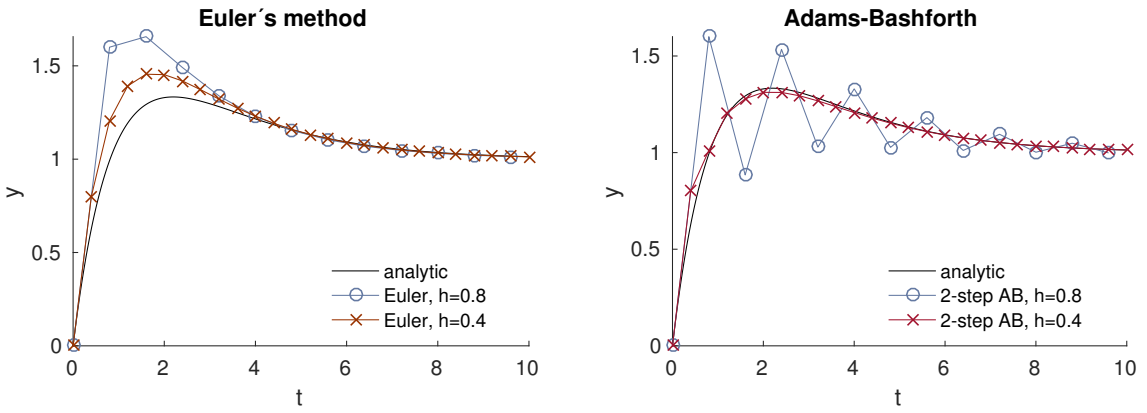


Figure 5: Demonstration of Euler's method and a two-step Adams-Bashforth method. Marked lines represent numerical approximations for the solution of the IVP in Equation 20 with step sizes $h = 0.8$ and $h = 0.4$. Analytic solution has been plotted for reference. Global error is proportional to h for Euler's method, whereas for the two-step Adams-Bashforth method it is proportional to h^2.

study how the two methods perform in solving this IVP on the interval $t \in [0, 10]$. Figure 5 shows the analytic solution and numerical approximations obtained with the methods using two different step sizes. In AB(2), the first step is taken with Euler's method, which clearly exaggerates the overshoot in the beginning. For $h = 0.8$, AB(2) experiences large oscillations around the real solution whereas Euler's method gives a smoother solution, which however is also clearly off. Since AB(2) is a second order method [26], it is not surprising that it gives a clearly better approximation when h is halved.

In order to solve IVPs reliably, we are interested in types of LMMs that converge to the real solution of the system (Equation 15). We adopt the style of [26] to present theoretical considerations for a k-step LMM defined by Equation 18. For convergence results, we start by assuming that the additional starting values $y_{k-1}, y_{k-2}, \ldots, y_1$ are computed using a convergent method.

The coefficients in Equation 18 can be used to define the characteristic polynomials

$$\rho(r) = \alpha_k r^k + \alpha_{k-1} r^{k-1} + \ldots + \alpha_0 \tag{21}$$

$$\sigma(r) = \beta_k r^k + \beta_{k-1} r^{k-1} + \ldots + \beta_0, \tag{22}$$

which we normalize so that $\alpha_k = 1$. We say that a method is consistent, if $\rho(1) = 0$ and $\rho'(1) = \sigma(1)$, i.e.

$$\sum_{j=0}^{k} \alpha_j = 0 \quad \text{and} \quad \sum_{j=0}^{k} j\alpha_j = \sum_{j=0}^{k} \beta_j. \tag{23}$$

Furthermore, a k-step LMM is called zero-stable if all roots of the polynomial $\rho(r)$ satisfy $|r| \leq 1$ and those roots for which $|r| = 1$ are simple. A theorem by Dahlquist [14] states that an LMM is convergent if and only if it is both consistent and zero-stable. Using this fact, it is simple to check that for example the presented AB(2) method is convergent.

When choosing an appropriate LMM to use in an application, one must take into account not only convergence, but also other properties such as the order of the method. Thus, one should consider a result called the first Dahlquist barrier [14, 15], which states that if an LMM is zero-stable, its order p satisfies

1. $p \leq k + 2$ if k is even
2. $p \leq k + 1$ if k is odd
3. $p \leq k$ if $\beta_k \leq 0$ (this is true in particular for all explicit methods).

3.4 Stiff ODEs and BDF methods

The concept of stiffness [13] can be defined in various ways, and the definitions are often rather fuzzy. A simple definition is that a problem is stiff is explicit methods perform badly in solving it [27]. Biochemical reaction systems that involve reaction rates that differ by orders of magnitude are often stiff [55]. This motivates focusing on methods that are designed for stiff problems.

A stability property that is suitable for studying behaviour of numerical methods that solve stiff systems is called A-stability [27]. In order to introduce A-stability for LMMs [16], we return to following the presentation in [26] and consider the standard test problem

$$y'(t) = \lambda y(t), \qquad y(0) = y_0, \tag{24}$$

where $\lambda \in \mathbb{C}$ and $\mathrm{Re}(\lambda) < 0$. The analytic solution $y(t) = y_0 e^{\lambda t}$ has the property $y(t) \to 0$ as $t \to \infty$. This is a property that is desirable for a solution given by a numerical method to have, too. We say that the region of absolute stability of an LMM is D, if for all fixed $\hat{h} = h\lambda \in D$ and any given starting values, solutions to the test problem in Equation 24 tend to zero when $t \to \infty$. Moreover, an LMM is said to be A-stable, if its region of absolute stability contains the entire left complex half-plane.

Unfortunately, A-stability is a very demanding property. This is quantified by the second Dahlquist barrier [16], which states that

1. an explicit LMM cannot be A-stable
2. the maximum order of an A-stable LMM is two.

In practice, one wants to use methods that have an order higher than two. Therefore, methods for which the region of absolute stability includes not all, but a large part of the negative half-plane, and in particular the negative real axis, have been designed. Efficient methods of this kind are the backward differentiation formulae (BDF) [58]. BDF methods [13] are implicit linear multistep methods, and the iterative formula for a k-step BDF method is

$$y_{i+k} + \alpha_{k-1} y_{i+k-1} + \ldots + \alpha_0 y_n = h \cdot \beta f_{i+k}, \tag{25}$$

where $\beta \neq 0$ [58]. The coefficients for a k-step BDF method can be set so that they maximize the order accuracy (confer [58], page 349). This yields unique coefficients

for $k = 1, 2, 3, 4, 5, 6$ (see for example [8], page 333). For $k \geq 7$, BDF methods are not zero-stable and thus not very useful [26].

In summary, numerical methods for ordinary differential equations are a well-studied field. Important theoretical aspect to check is that the methods one applies are convergent. When designing practical linear multistep methods, one should keep in mind that there exists theoretical results for them that restrict the relationship between stability and order of a convergence. An important class of explicit LMMs are the BDF methods, which we will utilize in this study.

Above we discussed methods for which the step size h is kept constant during the whole process of integrating an numerical solution. In practice, many numerical software adapt the step size by taking shorter steps at regions where the solution experiences rapid variations and longer steps when the solution is in a more steady state [26]. This can improve efficiency, because same accuracy can be reached with fewer steps or better accuracy with the same number of steps [26]. Ideally, the process of adapting the step sizes suitably is automatic and inexpensive compared to the computational cost avoided using it [26]. For an introduction to such strategies, see [26], chapter 11.

4 Statistical inference

Quantitative mathematical models of biochemical systems, such as signal transduction pathways or gene regulatory networks, often have to be calibrated using experimental data. This can be due to uncertainty in the model parameters or the model structure itself [25]. The unknown parameters can be for instance reaction rates or initial concentrations. Calibrating the parameters of a dynamical mathematical model so that it agrees with experimental data, is in fact a central task in systems biology. The data usually consists of time course measurements of the system components, such as concentrations of different molecular species. Usually the measurement times have the role of an independent variable, and are assumed to be known exactly, as opposed to the observations of the system state variables, which are modeled as noisy versions of the actual values. This noise then motivates probabilistic model calibration, which is a process of tuning a model such that its output is as likely as possible. Statistical inference techniques can also be applied to choose the most likely model, when there are numerous alternative hypotheses about the mechanisms of a biochemical system. A generally recommended approach for model evaluation and comparison is to study a model's expected predictive accuracy [20]. However, estimating the predictive accuracy for example by performing cross-validation is not a suitable technique for the models considered here, since fitting an ODE model using only a subset of the data can give very biased predictions. Furthermore, if the cross-validation is done using multiple folds, the computations related to fitting a high-dimensional ODE model several times are too expensive. Therefore we employ Bayesian analysis [7, 20, 37, 53] to calibrate the model parameters and to perform model comparison.

4.1 Bayesian inference

In Bayesian data analysis, probability models are harnessed to produce inferences from observed data and hence obtain information about unobserved quantities of interest. To be more exact, Bayesian inference gives us a probability distribution on the model parameters or some other unobserved quantities, like predictions for new observations. Here, we present the basics of Bayesian inference, using the book by

Gelman et al. [20] as the main source.

In the following we assume that we have collected a data set $\mathcal{D}$ and designed a model whose parameters are contained in the vector $\theta \in \Theta$, where Θ is the space of possible parameter vector values. Bayesian analysis is initialized by defining the joint probability distribution of all observable and unobservable quantities involved. Here, the observable quantities are the data whereas the parameters are unobservable, so our joint distribution is $p(\mathcal{D}, \theta)$. Setting this distribution can be difficult and demands knowledge about the underlying application and the data collection process. A reasonably specified joint probability distribution is often expressed as the product $p(\mathcal{D}, \theta) = p(\mathcal{D} \mid \theta)p(\theta)$, where $p(\mathcal{D} \mid \theta)$ is called likelihood and $p(\theta)$ is our prior distribution. Likelihood expresses the probability at which the model generates the observed data when the parameters have values θ. For example, let us consider an ODE model with n components and a data set containing K time course observations of each state variable. The data set is then $\mathcal{D} = \{\mathcal{D}_{ik} \mid i = 1 \ldots, n; k = 1, \ldots, K\}$, where $\mathcal{D}_{ik} = \{t_i, y_{ik}^\dagger\}$ contains the measured value $y_{ij}^\dagger$ of the state variable j at time point t_i. Evaluating the likelihood of the data, given parameters θ, then involves computing the model output $y_{ij} = y_i(t_i, \theta)$ at the data points, which inevitably involves solving the ODE system. If the measurements are assumed to be independent and identically distributed, the likelihood function is given by

$$p(\mathcal{D} \mid \theta) = \prod_{i=1}^{n} \prod_{k=1}^{K} g(y_{ik}^\dagger \mid y_{ik}, \theta), \tag{26}$$

where $g(y_{ik}^\dagger \mid y_{ik}, \theta)$ expresses the likelihood of a single data point $\mathcal{D}_{ij}$. Note that the term likelihood function means that $p(\mathcal{D} \mid \theta)$ is a function of θ. Defining a likelihood model generally involves assumptions about the distribution of the measurements, and the parameter vector θ may also contain parameters of that distribution. For example, if the data was assumed to be normally distributed with mean y_{ik} and variance σ^2, then $g(y_{ik}^\dagger \mid y_{ik}, \theta)$ would be the probability density function value of the corresponding normal distribution at $y_{ik}^\dagger$. The prior on the other hand contains all beliefs of the parameter distribution before any data are seen. Thus, both the likelihood and the prior involve assumptions that require knowledge of the scientific application in question.

After observing data, we can condition on it using an equation called Bayes' formula to obtain the posterior distribution

$$p(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D}, \theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D} \mid \theta)p(\theta)}{p(\mathcal{D})}, \tag{27}$$

where the normalizing constant

$$p(\mathcal{D}) = \int_{\theta \in \Theta} p(\mathcal{D} \mid \theta)p(\theta)\mathrm{d}\theta \tag{28}$$

is called marginal likelihood. Now the posterior distribution, which can be seen as a compromise between the prior and the observed data, includes all information that we have about the parameters. Bayesian analysis has the salutary property that it

lets the data speak, meaning that the effect of the prior on the posterior – and thus the subjectivity contained in it – diminishes as more data are observed. Furthermore, the sensibility of other subjectively set schemes involved in the Bayesian analysis, such as the assumptions that are made when the likelihood model is determined, can often be tested statistically if the available data set is abundant.

Bayesian treatment offers us a posterior distribution $p(\theta \mid \mathcal{D})$ that is a probabilistic description of all available information about the unobservable parameters θ. However, often the parameter values themselves are not of interest, and instead one can wish to harness the information to provide predictions about other potentially observable quantities. For instance, if the data consists of time course measurements of a dependent variable y at some time points, one can evaluate a prediction y^* for y at any time point t^*. In order to obtain a prediction that captures the whole posterior and thus all information that we have, one can use the posterior predictive distribution

$$p\left(y^* \mid t^*, \mathcal{D}\right) = \int_{\theta \in \Theta} p(y^* \mid t^*, \theta) p(\theta \mid \mathcal{D}) \mathrm{d}\theta. \tag{29}$$

Note that this approach is not limited to a single independent variable t and can be extended to any set of covariates. The benefit of this fully Bayesian approach is that in addition to a single prediction, such as the expectation $\mathrm{E}\left[p\left(y^* \mid t^*, \mathcal{D}\right)\right]$, the posterior predictive distribution offers confidence bounds for y^*. Predictions given by a model can then be used to assess the quality of the model in question by measuring how well they agree with the data points.

4.2 Bayesian model ranking

Besides parameter posterior analysis of systems biology models, another central challenge in the field is to find the most suitable models when different model hypotheses are viable [61]. Different model structures can for example correspond to alternative network configurations of a gene regulatory network. In Section 4.1 we presented the Bayesian parameter estimation methodology assuming that the system model structure is known. In this section, we consider a case where we have defined a set $\mathcal{M} = \{M_1, M_2, \ldots, M_K\}$ of alternative well-defined mathematical models. When experimental data are available, statistical methods can then be applied to rank these models by evaluating a quantity that is proportional to the posterior probability of the model. This section presents the Bayesian model ranking scheme, mainly referring to [25].

To capture the uncertainty over K different candidate models, we aim to obtain a model posterior distribution $p(M_k \mid \mathcal{D})$ over $k = 1, \ldots, K$. Before observing data, we set a model prior distribution $p(M_k)$ that describes which models are likely *a priori*. After seeing data $\mathcal{D}$, the Bayes' formula gives the posterior as

$$p(M_k \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid M_k) p(M_k)}{\sum_{i=1}^{K} p(\mathcal{D} \mid M_i) p(M_i)}, \tag{30}$$

where $p(\mathcal{D} \mid M_k)$, called the model marginal likelihood or evidence, is given by

$$p(\mathcal{D} \mid M_k) = \int_{\theta \in \Theta_{M_k}} p(\mathcal{D} \mid M_k, \theta) p(\theta \mid M_k) \mathrm{d}\theta. \tag{31}$$

Table 1: Interpretation of the Bayes' factor in favour of model 1 as opposed to model 2, according to Jeffreys [37].

$\log_{10}(B_{12})$	B_{12}	Interpretation
0 to 1/2	1 to 3.2	Only worth a bare mention
1/2 to 1	3.2 to 10	Substantial
1 to 2	10 to 100	Strong
> 2	> 100	Decisive

Here $p(\theta \mid M_k)$ now describes the prior belief about the parameters θ, given that the model is assumed to be M_k. The term $p\left(\mathcal{D} \mid \theta, M_k\right)$ expresses how likely it is to see the data when the model is M_k and parameters take the value θ. Now, competing hypotheses can be compared by the corresponding model posterior probabilities [61]. Note that model comparison can be done using only the unnormalized posterior probabilities $p(\mathcal{D} \mid M_k)p(M_k)$.

The evidence contained in the data in favour of one model as opposed to another can be expressed in the form of a Bayes' factor [61]. The Bayes' factor between models M_1 and M_2 is given by

$$B_{12} = \frac{p(\mathcal{D} \mid M_1)}{p(\mathcal{D} \mid M_2)}, \tag{32}$$

i.e. the ratio of their marginal likelihoods. One way to interpret these factors is given by Jeffreys [37] who suggested intervals on the $\log_{10}$ scale, with the corresponding interpretations shown in Table 1. Using Bayes factors is mainly sensible when the candidate model set is truly discrete and when it is reasonable to assume that either one or the other model is a good description of the data [20]. Alternative model ranking techniques, such as cross-validation, are based on estimating the expected predictive error of the models [20].

4.3 Parameter identifiability analysis

Sometimes the goal is to solely calibrate a model by determining a single point estimate for the parameters such that the model agrees with experimental data in an optimal way. This is done by optimizing an objective function, such as the likelihood $p(\mathcal{D} \mid \theta)$ that was introduced in Section 4.1 [48]. For some models it might be the case that given the observed data, the optimization problem is underdetermined and there does not exist a unique point in the parameter space that maximizes the posterior probability [38]. This obstacle is called non-identifiability and it can be either structural or practical [38].

Structural non-identifiability [5] of a parameter originates from the formulation of the model itself and is independent of any data [48, 49]. For example if the parameters θ_1 and θ_2 appear in a model only as the product $\theta_1\theta_2$, there are infinitely many combinations of the two parameters that yield the same model output, and consequently θ_1 and θ_2 are both structurally non-identifiable. In this case, the

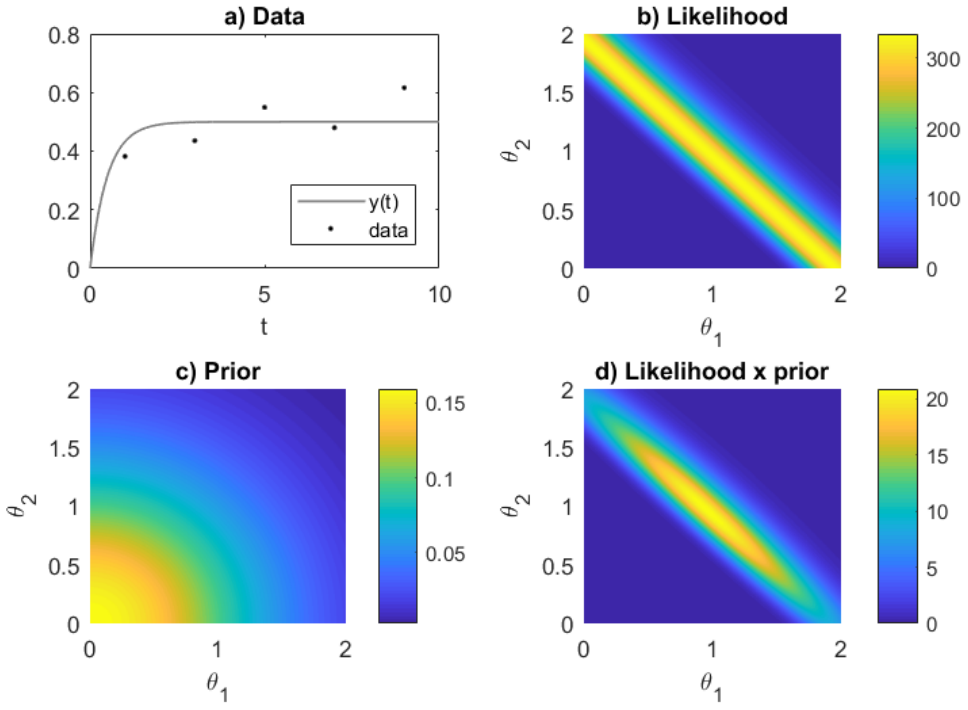


Figure 6: Demonstration of a case where structurally non-identifiable parameters become identifiable in the Bayesian sense. a) Noisy data generated from the model in Equation (34) with parameter values $\theta_1 = \theta_2 = 1$. b) The likelihood surface does not have a single isolated maximum. c) An informative prior distribution. d) The parameter posterior distribution is proportional to the product of the likelihood and the prior distribution. This surface has a unique maximum.

problem can be eliminated by replacing the product with a single parameter which possibly is then identifiable [38]. For simple models, a similar check can be carried out easily, but generally structural non-identifiability can be difficult to detect for complicated models that appear in applications [48].

Practical non-identifiability on the other hand can arise even if the model is structurally identifiable [49]. This happens when the data set is not informative enough to determine the parameters [38]. For example for linear regression models, this is clearly the case if there are less data points than parameters.

If the non-identifiability problem cannot be solved by reparametrization, one can modify the target function of the parameter calibration by applying a regularization that makes the parameters indentifiable. In Bayesian inference, parameters that are non-identifiable with respect to likelihood, become identifiable, when an informative prior distribution is used. For example, let us consider the ODE

$$\frac{\mathrm{d}y(t)}{\mathrm{d}t} = 1 - (\theta_1 + \theta_2)y(t), \tag{33}$$

which with the initial condition $y(0) = 0$ has the analytical solution

$$y(t) = \begin{cases} t\ , & \text{if } \theta_1 + \theta_2 = 0 \\ (\theta_1 + \theta_2)^{-1} \left(1 - e^{-(\theta_1+\theta_2)t}\right) & \text{otherwise} \end{cases} . \tag{34}$$

The parameters θ_1, θ_2 are structurally non-identifiable, since increasing one can always be compensated by decreasing the other. Figure 6a displays a noisy data set generated from this model with $\theta_1 = 1$ and $\theta_2 = 1$ and Figure 6b shows the likelihood surface, when normal noise is assumed. Because all parameter combinations that satisfy $\theta_1 + \theta_2 = C$ yield the same output, the likelihood surface attains its maximal value on a line in the parameter space. Here, C is not exactly 2 due to noise in the data. If one has reason to believe that the real parameter values are small and wishes to include this regularization in the analysis, a standard bivariate normal prior (Figure 6c) could be used. The posterior, which is proportional to the product of the prior and likelihood, then also has a unique maximum (Figure 6d). When posterior inferences are made, data still speaks through the likelihood, making the parameter combinations that add up to around two clearly stand out, but the prior information makes the values around $(1, 1)$ the most probable combinations.

4.4 Comparison of hypotheses about model structure

Section 4.2 demonstrated how Bayesian methodology can be applied to compare individual models. This analysis can be generalized further to allow comparison of groups of models, that can have an arbitrary number of members. Consequently, we can perform ranking of different hypotheses about the model structure, by grouping together models that share some property of interest.

We assume that there are K different models in total and h alternative model structure hypotheses $H_1, \ldots, H_h$. Each hypothesis H_k corresponds to an index set $\mathcal{I}_k \subset \{1, \ldots, K\}$ such that all $\mathcal{I}_k$ are disjoint and $\bigcup_{k=1}^{h} \mathcal{I}_k = \{1, \ldots, K\}$. Now, conditioning on a hypothesis H and model M, fully Bayesian treatment involves expressing the parameter posterior in the form

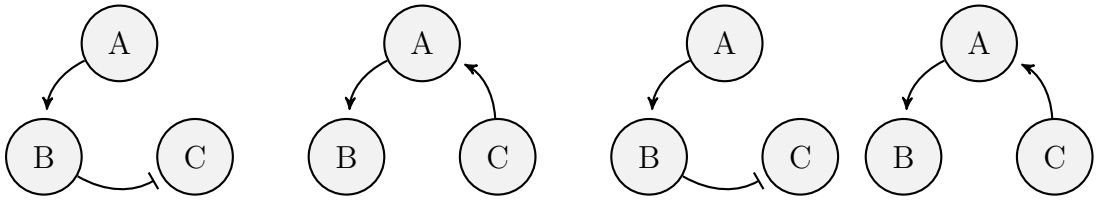
$$p(\theta \mid \mathcal{D}, M, H) \propto p(\mathcal{D} \mid M, H, \theta) p(\theta \mid M, H), \tag{35}$$

where $p(\theta \mid M, H)$ is now the parameter prior, given H and M. The marginal likelihood $p(\mathcal{D} \mid M, H)$ is obtained by integrating out the rate parameters like in Equation 31. Furthermore, the posterior distribution over all the models, conditioned with H, becomes

$$p(M \mid \mathcal{D}, H) = \frac{p(\mathcal{D} \mid M, H) p(M \mid H)}{\sum_{i=k}^{K} p(\mathcal{D} \mid M_k, H) p(M_k \mid H)}, \tag{36}$$

where $p(M \mid H)$ is the prior over models, given H. To rank the different hypotheses, we again compare their posterior probabilities

$$p(H_k \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid H_k) p(H_k)}{\sum_{i=1}^{h} p(\mathcal{D} \mid H_i) p(H_i)}, \tag{37}$$



(a) One-phase model M_1 (b) One-phase model M_2 (c) Two-phase model M_3

Figure 7: Three alternative configurations of a regulatory network. Activated and repressible expression are represent using arrows ($\rightarrow$) and turnstiles ($\dashv$), respectively. The first two are stationary models with a single network wiring, where as the third one has two sequential phases.

where the values $p(H_k)$, $k = 1, \ldots, h$ comprise the prior distribution over alternative hypotheses. This involves computing the likelihood of data given a hypothesis, i.e.

$$p(\mathcal{D} \mid H_k) = \sum_{i \in \mathcal{I}_k} p(\mathcal{D} \mid M_i, H_k) p(M_i \mid H_k). \tag{38}$$

Equation 37 provides a well-defined probabilistic measure that can be used to assess distinct hypotheses about the model structure. This is beneficial, since for instance LEM modeling, which was introduced in Section 2.3, has applications that involve several alternative hypotheses about the latent process. In particular, in the Th17 cell differentiation application considered in this study, the number of latent states in the driving gene regulatory networks is unknown. Models with different numbers of latent states can then be grouped together as a hypothesis about the model structure, and the different hypotheses can be ranked in a well-defined manner.

To clarify the hypothesis testing for LEM models, let us consider three alternative models M_1, M_2 and M_3. Models M_1 and M_2 corresponds to a stationary network structure in Figures 7a and 7b, respectively. The model M_3 in Figure 7c is a two-phase model where the subnetworks of M_1 and M_2 are the two sequential states. We wish to compare the hypotheses H_m: "There are m latent states." for $m = 1, 2$, assuming that the computed marginal likelihoods for the models are $p(\mathcal{D} \mid M_1, H_1) = 10$, $p(\mathcal{D} \mid M_2, H_1) = 20$ and $p(\mathcal{D} \mid M_3, H_2) = 35$. If we use a uniform prior over models given either of the hypotheses, we get $p(\mathcal{D} \mid H_1) = \frac{10+20}{2} = 15$ and $p(\mathcal{D} \mid H_2) = 35$. If the prior over hypotheses is also uniform, this yields the posterior probabilities

$$p(H_1 \mid \mathcal{D}) = \frac{15}{15 + 35} = 0.3 \qquad \text{and} \qquad p(H_2 \mid \mathcal{D}) = \frac{35}{15 + 35} = 0.7. \tag{39}$$

4.5 Probabilistic inference of model mechanisms

In model structure inference, the goal can be to obtain probability statements that capture the uncertainty related to different model mechanisms, instead of just looking for a single best model. If the posterior distribution over alternative model

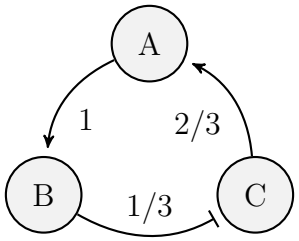


Figure 8: Example of probabilistic predictions obtained by analysis of networks in Figures 7a and 7b.

configurations is completely available, we can determine how probable the individual model mechanisms are, by averaging over all the model configurations. This yields so-called posterior weights for each mechanism.

To perform this inference task, we formulate two alternative hypotheses H_1: "mechanism j is present" and H_0: "mechanism j is not present. We denote the subset of models that contain the mechanism j by $\mathcal{I}_j$. Let us assume that we have a model posterior distribution $p : \mathcal{M} \to [0, 1]$. Now, a posterior weight that captures the probability of hypothesis H_1 is given by

$$\omega_p^j = \frac{\sum_{M \in \mathcal{I}_j} p(M)}{\sum_{M \in \mathcal{M}} p(M)}, \tag{40}$$

In particular, if the model space $\mathcal{M}$ consists of LEM models, each of which can be defined with a binary matrix Z (see Section 2.3), then the matrix

$$W_p = \frac{\sum_{Z \in \mathcal{M}} p(Z) Z}{\sum_{Z \in \mathcal{M}} p(Z)} \tag{41}$$

contains the posterior-weighted averages for each element in the model configuration matrix over all possible models. We note that computing this matrix for LEM models is sensible only when the number of latent states M is fixed, but can be carried out separately for different M. The element (j, k) of this matrix can be interpreted as the probability of the corresponding mechanism j being present in the kth latent state.

As an example, we can study the networks in Figure 7, by assuming the same marginal likelihoods and priors as given in previous section. If we fix the hypothesis H_1, the posterior probabilities given by Equation 36 are $p(M_1 \mid \mathcal{D}, H_1) = \frac{1}{3}$ and $p(M_2 \mid \mathcal{D}, H_1) = \frac{2}{3}$. Figure 8 shows the resulting probabilistic predictions for the mechanisms given by Equation 40. Of course, it is clear that the mechanisms that are present in all possible models get a probability of 1 whereas mechanisms not present in any model have probability 0. Thus, defining a model space that is large enough to capture the structure uncertainty is important.

In the applications of this thesis, however, the posterior weights have to be computed by using an approximation q for p, since it often is not feasible to compute the posterior probability for all viable models. This then results in an approximated

posterior weight matrix W_q. The following chapter focuses on how to obtain a good approximation q efficiently. This approximative distribution can then also be used to perform the hypothesis testing presented in the previous section.

This chapter introduced the basics of Bayesian inference, which is the cornerstone of the data-driven ODE model structure inference in this thesis. We have rigorously presented a probabilistic framework for comparing different model structures. The focus of this thesis will now move to computational techniques that make the marginal likelihood computations and the model structure inference possible even for large sets of high-dimensional models.

5 A novel model structure inference strategy based on efficient exploration of the model space

The previous chapter presented the theoretical basis for the statistical model structure inference considered in this study. In this section, we introduce a novel strategy for performing structure inference for ODE models. Fully Bayesian treatment of such nonlinear ODE models is not feasible, since the integrals that appear in marginal likelihood formulas (Equation 31) are not analytically tractable [25]. Thus, also parameter posterior (Equation 27) and our desired model posterior (Equation 30) are not available in closed form. Parameter dimension of the models is often large and thus numerical integration has to be ruled as well. Various approximative methods for computing the model marginal likelihood have been developed [18], but even so the reliable evaluation of the full model posterior distribution is restricted to very small model spaces due to computational cost. Throughout the rest of this thesis, the term model posterior can also refer to the model posterior conditional on some hypothesis (Equation 36).

In applications that are encountered in fields such as systems biology, the set of alternative models can often be very large due to uncertainty in the model structure. On the other hand, if the data are informative enough, it can turn out that a small fraction of the alternative model configurations stand out as remarkably more likely than most models. This is why we propose using Markov chain Monte Carlo (MCMC) methods to explore the discrete model space cleverly and consequently, to obtain a good approximation for the model posterior distribution at a rather bearable computational cost. Computing this approximation is then an efficient strategy for obtaining meaningful inferences about the model structure. In particular, we explicitly formulate a Metropolis algorithm that can be used to perform structure inference for LEM models (Section 2.3).

5.1 Markov chain Monte Carlo methods

Markov chain Monte Carlo (MCMC) is a very popular and widely used strategy to solve integration and optimization problems faced for example in machine learning, physics and statistics [3]. It is often used to sample, i.e. draw realizations of x from a distribution $p(x)$ that is not available in a closed form. Such distributions are for example many posterior distributions encountered in Bayesian analysis [54]. MCMC relies on the fact that samples from a target distribution $p : \mathcal{X} \to [0, 1]$ can be generated by exploring the state space $\mathcal{X}$ using a Markov chain which has p as its invariant distribution [3]. This facilitates analysis of distributions that are not available in closed form but can only be evaluated at different points. Here we present the basic MCMC theory, following [54].

A Markov chain on state space $\mathcal{X}$ is a stochastic process defined by a sequence of random variables $X_t \in \mathcal{X}$, such that for all $t = 1, 2, \ldots$, the conditional probability of X_{t+1} given all previous variables in the sequence depends only on X_t. The joint distribution of all X_t is determined by the distribution of the initial state X_1 and a transition kernel K, which is a conditional probability density such that $X_{t+1} \sim K(X_{t+1} \mid X_t)$. For any $A \subset \mathcal{X}$, the transition probability from X_t to A satisfies

$$P(X_{t+1} \in A \mid X_t) = \int_A K(x \mid X_t)dx. \tag{42}$$

The MCMC methods used in this study are based on random walks in the state space $\mathcal{X}$. This means that X_{t+1} is generated by adding a (small) change to X_t, in the sense of the metric that is associated with $\mathcal{X}$. In this study, we only consider homogenous Markov chains, i.e. ones where $K(X_{t+1} \mid X_t)$ is independent of t. Furthermore, we will only utilize Markov chains on a countably finite state space $\mathcal{X} = \{x_1, \ldots, x_s\}$. In this case, a probability mass distribution on $\mathcal{X}$ can be expressed as a vector $\pi = [\pi_1, \ldots, \pi_s]$ and the transition kernel can be expressed as a matrix T, where the element $\{T\}_{ij} = K(X_{t+1} = x_j \mid X_t = x_i)$.

A distribution π is called the invariant or stationary distribution of a Markov chain, if it follows from $X_1 \sim \pi$ that $X_t \sim \pi$ for all t [22]. In order to use a Markov chain for MCMC, it must have a unique invariant distribution π, independent of the initial state [54]. To guarantee this, the matrix T must have two properties called irreducibility and aperiodicity [3]. Irreducibility means that the chain can reach all other states no matter where it is started, whereas aperiodicity prohibits getting trapped in cycles [3]. A sufficient condition to ensure that a particular distribution π is an invariant distribution of a Markov chain is the detailed balance condition

$$\pi_i\{T\}_{ji} = \pi_j\{T\}_{ij}, \tag{43}$$

for all $i, j \in \{1, \ldots, s\}$ [3]. These concepts can also be extended to allow $\mathcal{X}$ to be continuous. However, such theory is not presented here, since only discrete-space MCMC algorithms are used in this thesis.

Designing practical MCMC methods can be done by defining transition rules that ensure aperiodicity and irreducibility and satisfy the detailed balance condition [3]. Then, a realization or realizations of the chain can be simulated to obtain samples

from the target distribution [20]. Different state-of-the-art MCMC implementations can differ for example by their number of chains, adaptivity properties or use of gradient information [4]. The most popular basic types of MCMC methods are Gibbs sampling [21] and the Metropolis-Hastings algorithm [29, 45], which is a random walk Monte Carlo algorithm that we will utilize in this study. If the current state is x, a Metropolis-Hastings algorithm proceeds by drawing a candidate value x^* from a proposal distribution $q(x^*|x)$ [3]. This candidate is then accepted to become the next state of the Markov chain with probability $\min\{1, r\}$, where

$$r = \frac{p(x^*)q(x \mid x^*)}{p(x)q(x^* \mid x)}, \tag{44}$$

and otherwise the chain remains in the state x [3]. It is straightforward to show that methods of this kind have $p(x)$ as their invariant distribution (see [20], pages 279–280).

5.2 Exploring the posterior distribution over model structures using MCMC

In our model structure inference problem, the idea is to exploit the fact that a Markov chain related to any MCMC method that has the model posterior distribution $p(M \mid \mathcal{D})$ as its target distribution gets attracted towards the high probability models that we are interested in. Because our model space is discrete, visiting a model for which the (unnormalized) posterior probability has already been evaluated does not provide us with any more information. If aperiodicity, irreducibility and the detailed balance are satisfied, we already know that the proportion of times the model is sampled will approach the posterior probability of that model, since the model posterior distribution is the invariant distribution of the chain. This is why we our focus is not directly in sampling from the posterior distribution, but rather just searching for its high probability regions. The main realization is that when the posterior probability has been evaluated for high probability models, the entire posterior can be approximated in a manner that is sufficient for reliable model structure inferences. Consequently, we need not worry about common issues of continuous space MCMC sampling like burn-in or correlation of consecutive states.

Choosing an efficient proposal distribution is vital for MCMC methods to have desirable properties, such as rapid convergence to the invariant distribution, high proportion of accepted moves, or good mixing of the chains [56]. In order to establish an efficient proposal distribution in the model space, we wish to define a sensible distance relation between the different models. In this section, we assume that the model space $\mathcal{M}$ is a set of all $N \times M$ binary matrices, each corresponding to a LEM model (see Section 2.3). For model configurations $Z, Y \in \mathcal{M}$, we define their distance relation $d : \mathcal{M} \times \mathcal{M} \to 0 \cup \mathbb{N}$ as

$$d(Z, Y) = \|Y - Z\|_F^2 \tag{45}$$

where $\|A\|_F$ is the Frobenius norm of matrix A. Note that when Y and Z are binary matrices, $d(Z, Y)$ is the number of their differing elements. It is easy to check that

$d(\cdot,\cdot)$ is symmetric, non-negative and subadditive, and therefore $(\mathcal{M}, d)$ is a metric space. Now, we define the k-neighborhood of a model Z as

$$\mathcal{N}_k(Z) = \{Y \in \mathcal{M} :\ 1 \leq d(Z,Y) \leq k\}. \tag{46}$$

If $Y \in \mathcal{N}_k(Z)$ with $k \ll N \times M$, then the models Z and Y have a high degree of similarity and many common components. Consequently, they are likely to have rather comparable posterior probabilities. Thus, a random walk Monte Carlo algorithm with a proposal distribution that has most weight on models in k-neighborhoods of the current state with small k presumably allows reasonably smooth moving with a practical proportion of accepted proposals.

5.2.1 A Metropolis-type algorithm for model posterior sampling

We now explicitly formulate a discrete space Metropolis algorithm to be used in our structure inference. The Metropolis algorithm [45] is a special case of Metropolis-Hastings, where the proposal distribution $q(Z^* \mid Z)$ is symmetric, i.e. $q(Z^* \mid Z) = q(Z \mid Z^*)$ for all $Z, Z^* \in \mathcal{M}$. We assume that for any $Z \in \mathcal{M}$, we can evaluate an unnormalized version $\pi_H(Z) \propto p(\mathcal{D} \mid Z, H)p(Z \mid H) > 0$ of its posterior probability, possibly given a hypothesis H. We use a proposal distribution that is uniform over the 1-neighbors of the current model, i.e.

$$q(Z^* \mid Z) = \begin{cases} \frac{1}{NM}, & \text{if } Z^* \in \mathcal{N}_1(Z) \\ 0, & \text{otherwise} \end{cases} \tag{47}$$

which clearly is symmetrical, because $Z^* \in \mathcal{N}_1(Z)$ if and only if $Z \in \mathcal{N}_1(Z^*)$. If the current state is Z, one Metropolis step consists of the following parts:

1. Draw a proposal model Z^* from the discrete proposal distribution $q(Z^* \mid Z)$.

2. Accept transition to Z^* with probability

$$A(Z^* \mid Z) = \min\left\{1, \frac{\pi_H(Z^*)}{\pi_H(Z)}\right\} \tag{48}$$

   and reject it with the probability $1 - A(Z^* \mid Z)$.

Now the transition probability from state Z to Z^* is the product $K(Z^* \mid Z) = q(Z^* \mid Z)A(Z^* \mid Z)$. For any models Z and Z^* such that $Z^* \in \mathcal{N}_1(Z)$, we have

$$\pi_H(Z)K(Z^* \mid Z) = \min\left\{\frac{\pi_H(Z)}{NM}, \frac{\pi_H(Z^*)}{NM}\right\} = \pi_H(Z^*)K(Z \mid Z^*). \tag{49}$$

Furthermore, if $Z^* \notin \mathcal{N}_1(Z)$, also $Z \notin \mathcal{N}_1(Z^*)$ and both sides of Equation 49 equal to zero, meaning that the detailed balance condition is satisfied. Since $\pi_H(Z) > 0$ for all $Z \in \mathcal{M}$, there is always a positive probability of changing any element of the matrix Z, so all states are always accessible. Furthermore, possibility of rejection disallows periodicity. It follows that the unique invariant distribution of the resulting Markov chain is $\pi_H(Z)$, independent of which model the chain is started from.

5.2.2 Model posterior distribution approximation

We start from a setting where we have first defined a model space consisting of well-motivated ODE models that are plausible under hypotheses and prior information concerning the model components and parameters. After measuring time course data of the ODE model components, our goal is to obtain information about the actual underlying model structure. Typically the full model posterior $p(Z \mid D, H)$ is computationally out of reach, and therefore we begin the task by approximating this discrete probability distribution with a distribution that has a relatively small support. Of course, for this approximation to be good, the support must contain a remarkable proportion of the high posterior probability models. In our approach, we use the Metropolis-type algorithm to efficiently find a subset of models that is sufficient for the approximation task and reliable structure inference.

Assume that we have run any search algorithm that has provided us with the (possibly approximated) marginal likelihood values $p(D \mid Z, H)$ for each $Z \in \mathcal{Z} \subseteq \mathcal{M}$. The model posterior approximation is then the distribution $p_a : \mathcal{M} \to [0, 1]$, where

$$p_a(Z) = \frac{p(D \mid Z, H)p(Z \mid H)}{\sum_{Y \in \mathcal{Z}} p(D \mid Y, H)p(Y \mid H)} \tag{50}$$

for all $Z \in \mathcal{Z}$ and $p_a(Z) = 0$ for all $Z \in \mathcal{M} \backslash \mathcal{Z}$.

Assessing the converge of the Markov chains is an essential part of classical MCMC sampling [22]. Diagnostics that measure convergence are often based on the use of multiple independent MCMC chains, that can be started from different initial points [20]. In this study, too, we rely on using several independent chains in order to assess the reliability of the obtained results and to decide when to terminate the search. The model posterior approximations given by different MCMC chains are different if the chains have not reached the same high probability regions. Because our target distribution is discrete and we are not actually sampling in a traditional MCMC sense, we will not use the standard convergence diagnostics. Instead, we focus on measuring the similarity of distributions obtained from different independent chains using Equation 50 in order to assess the convergence of the approximations.

Common metrics that compare the similarity of two distributions p and q include for example the Bhattacharyya distance [6] and Kullback-Leibler divergence [41]. The latter one is commonly used in machine learning, but is not defined for distributions that do not have their whole domain as their support. We choose to use the overlapping (OVL) coefficient [32, 63], which measures the similarity by computing the area that the two distributions share. Because the integral of any probability distribution is 1, the OVL is bounded between 0 and 1, so that $\mathrm{OVL}(p, q) = 1$ if and only if $p = q$. A formula for computing the OVL for discrete probability mass distributions $p, q : \mathcal{M} \to [0, 1]$ is

$$\mathrm{OVL}(p, q) = \sum_{Z \in \mathcal{M}} \min \{p(Z), q(Z)\} . \tag{51}$$

5.2.3 Efficient marginal likelihood estimation

Our MCMC-based model space exploration is only possible, if the marginal likelihood

$$p(\mathcal{D} \mid Z, H) = \int_\theta p(\mathcal{D} \mid Z, H, \theta) p(\theta \mid Z, H) \mathrm{d}\theta \tag{52}$$

can be evaluated up to a normalizing constant. For nonlinear ODE models, this integral is not analytically tractable, which forces the use of approximative methods [25]. When the parameter dimension is high, estimation of the marginal likelihood is challenging, but there exist various approaches that differ in both accuracy and computational cost [10, 18]. In this study, we use the an approximation based on the Bayesian information criterion (BIC) [57]. It can be seen as a special case of a more general method called Laplace's method. A carrying assumption behind Laplace's method is that the d-dimensional surface to be integrated, denoted by $l(\theta) = p(\mathcal{D} \mid Z, H, \theta) p(\theta \mid Z, H)$, can be approximated with a d-variate normal distribution $N_d(\theta \mid \tilde{\theta}, \Sigma)$ [18]. The approximating distribution has its mean at $\tilde{\theta} = \max_\theta l(\theta)$ and its covariance matrix is $\Sigma = (-H_l)^{-1}$, where H_l denotes the Hessian of $l(\theta)$ [18]. The resulting integral can be now evaluated analytically, giving the marginal likelihood approximation

$$\int_\theta l(\theta) \mathrm{d}\theta = (2\pi)^{d/2} |\Sigma|^{1/2} p(\mathcal{D} \mid Z, H, \tilde{\theta}) p(\tilde{\theta} \mid Z, H), \tag{53}$$

where $|\cdot|$ denotes the matrix determinant [18]. If the parameter prior $p(\theta \mid Z, H)$ is uniform over all θ, the logarithm of the approximation reduces to

$$\log p(\mathcal{D} \mid Z, H) \approx \log p(\mathcal{D} \mid Z, H, \tilde{\theta}) - \frac{1}{2} \log(|H_l|) + C, \tag{54}$$

where C is a constant and $\tilde{\theta}$ now is the same as the maximum likelihood estimate $\theta_{\mathrm{ML}} = \max_\theta p(\mathcal{D} \mid Z, H, \theta)$ [52]. If the data $\mathcal{D}$ contains D data points and the model parameters are identifiable, i.e. θ_{ML} is unambiguous, the term $-\frac{1}{2} \log(|H_l|)$ can be considered roughly proportional to $-\frac{d}{2} \log(D)$ [57]. This leads us to the unnormalized logarithmic marginal likelihood approximation

$$\log p(\mathcal{D} \mid Z, H) \approx \log p(\mathcal{D} \mid Z, H, \tilde{\theta}) - \frac{d}{2} \log(D), \tag{55}$$

which is the BIC [52]. The first term expresses the likelihood of the data in the case where model parameters take their most likely values, and the second term penalizes the model complexity. BIC extends the normality assumption behind Laplace's method to the extreme by condensing the approximating normal distribution into a single point that carries all information about the surface that is integrated. One should acknowledge that this is a very strong assumption and usually not valid for nonlinear ODE models that can exhibit multimodal parameter posterior surfaces [10]. However, since Equation 55 can be evaluated simply by determining the maximum likelihood parameters θ_{ML}, BIC can provide us with adequate information for model comparison in a reasonable computation time. Moreover, BIC has can be considered practical, and it has been used in works involving real life applications [34, 44].

5.3 Computational implementation

5.3.1 Implementation details

The model structure inference and identifiability computations are implemented in MATLAB (The MathWorks Inc., Natick, MA, USA). Using BIC for the estimation of marginal likelihood reduces the computational problem to that of finding the maximum likelihood parameters θ_{ML}. This nonlinear optimization problem is rather difficult due to its high dimension, multimodality of the likelihood surface, and possible ill-posedness resulting from non-identifiability [46]. We tackle the problem by following recommendations by [50], where various deterministic and stochastic optimization algorithms were compared for parameter estimation in systems biology. Their results indicate that deterministic derivative-based optimization with a multi-start strategy outperforms various stochastic and other algorithms by orders of magnitude in both reliability and efficiency. Here, we present the implementation details that are mostly adopted from the calculations and guidelines presented in [50] and [51].

We denote the data by $\mathcal{D} = \{\mathcal{D}_{ik} \mid i = 1, \ldots, n;\ k = 1, \ldots, K_i\}$, where $\mathcal{D}_{ik} = \{t_k, y^\dagger_{ik}\}$ contains the measured value $y^\dagger_{it}$ of component y_i at time t_k and K_i is the number of measurements of component i. It is possible that $t_k = t_{k'}$ for some $k \neq k'$, which means that we have multiple replicates at that time point. All computations of this study assume that the measurement noise for $y^\dagger_{it}$ is normally distributed with mean y_{ik} and standard deviation $\sigma_{ik} = \alpha + \beta y_{ik}$. This means that there exists a basal noise level α and a component βy_{ik} which depends on the signal strength. The likelihood function (Equation 26) for a model Z thus takes the form

$$\mathcal{L}(\theta) = p(\mathcal{D} \mid \theta, Z, \alpha, \beta) = \prod_{i=1}^{n} \prod_{k=1}^{K_i} N(y^\dagger_{it} \mid y_{ik}, (\alpha + \beta y_{ik})^2), \tag{56}$$

where parameters α and β can either be contained in the vector θ or predetermined constants. Above, $N(\cdot \mid \mu, \sigma^2)$ denotes the probability distribution function of the one-dimensional normal distribution with mean μ and standard deviation σ, i.e.

$$N(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{(x-\mu)^2}{2\sigma^2}\right). \tag{57}$$

The maximum likelihood parameters are then given by

$$\theta_{\mathrm{ML}} = \arg\max_{\theta} \mathcal{L}(\theta) = \arg\min_{\theta} L(\theta), \tag{58}$$

where

$$L(\theta) = -2\log\mathcal{L}(\theta) = \sum_{i=1}^{n} \sum_{k=1}^{K_i} \left[2\log(\sqrt{2\pi}) + 2\log(\alpha + \beta y_{ik}) + \left(\frac{y^\dagger_{ik} - y_{ik}}{\alpha + \beta y_{ik}}\right)^2 \right].$$

In an optimization perspective, the function $L(\theta)$ has the interesting property that after small modifications, it can be expressed as a sum of squared components.

This is possible, if we add a constant c, which ensures that $2\log(\alpha+\beta y_{ik})+c \geq 0$ for all i,k and thus avoid square roots of a negative value. Because the term $2\log(\sqrt{2\pi})$ does not depend on θ, the final optimization problem for obtaining the maximum likelihood parameters is then to minimize the target function

$$L_c(\theta) = \sum_{i=1}^{n}\sum_{k=1}^{K_i}\left[\left(\sqrt{2\log(\alpha+\beta y_{ik})+c}\right)^2 + \left(\frac{y_{ik}^{\dagger}-y_{ik}}{\alpha+\beta y_{ik}}\right)^2\right], \tag{59}$$

which is a sum of $S = 2n\sum_{i=1}^{n} K_i$ squared components. This problem has the exact same solution as the one in Equation 58, since c is a constant. We use the value $c = 50$, and also set a safety correction for the standard deviations $\sigma_{ik} = \alpha+\beta y_{ik}$ such that $\sigma_{ik} = \exp(-\frac{c}{2}+10^{-6})$ if it still occurs that $2\log(\sigma_{ik})+c<0$. This correction should actually affect the target function $L_c(\theta)$, but we ignore its effect by assuming that $2\log(\sigma_{ik})+c<0$ only occurs very rarely and taking it into account would not make any difference in practice.

Solving the least squares optimization problem in Equation 59 is done using the trust-region-reflective algorithm of the LSQNONLIN optimization routine in MATLAB. The employed routine requires restricting the parameter space by defining bounds for the parameters θ. In general, trust region algorithms work by approximating the target function at the point θ with a simpler function that adequately imitates the the behavior of the target function in a neighborhood of θ [60]. For nonlinear least-squares problems, the MATLAB implementation exploits their special structure to achieve more efficient performance [60].

When the target function surface entails multiple local optima, local search algorithms, such as the trust-region-reflective algorithm used in this study, can terminate at different local optima depending on the starting point of the search [25]. In order to avoid this suboptimal performance, we employ a multistart strategy where the algorithm is started with several initial points that are sampled using a latin hypercube scheme. This approach prohibits any two starting points from being accidentally close to each other, and therefore provides a better coverage of the space (see [50] for a more detailed explanation). In this study, we set the amount of optimization starts to ten times the dimension of the parameter space. For accelerated performance, these starts are computed in parallel.

Optimization involves a large number of target function evaluations, and on each evaluation, an ODE system needs to be solved. In order to speed up the optimization, the we supply the Jacobian matrix of the target function to the LSQNONLIN routine. The Jacobian consists of derivatives of each component of the target function with respect to each parameter. In order to present these derivatives here, we denote components of target function in Equation 59 such that

$$L_c(\theta) = \sum_{i=1}^{n}\sum_{k=1}^{K_i}\left[r_{ik}(\theta)^2 + \tilde{r}_{ik}(\theta)^2\right], \tag{60}$$

where

$$r_{ik}(\theta) = \sqrt{2\log(\alpha+\beta y_{ik})+c} \qquad \text{and} \qquad \tilde{r}_{ik}(\theta) = \frac{y_{ik}^{\dagger}-y_{ik}}{\alpha+\beta y_{ik}}. \tag{61}$$

Derivatives of the components with respect to parameter θ_j are then

$$\frac{\mathrm{d}r_{ik}}{\mathrm{d}\theta_j} = \frac{\beta(\alpha+\beta y_{ik})^{-1}}{\sqrt{2\log(\alpha+\beta y_{ik})+c}} \cdot \frac{\mathrm{d}y_{ik}}{d\theta_j} \qquad \text{and} \qquad \frac{\mathrm{d}\tilde{r}_{ik}}{\mathrm{d}\theta_j} = -\frac{\alpha+\beta y_{ik}^{\dagger}}{(\alpha+\beta y_{ik})^2} \cdot \frac{\mathrm{d}y_{ik}}{d\theta_j}, \quad (62)$$

where $\mathrm{d}y_{ik}/\mathrm{d}\theta_j$, i.e. the derivatives of the ODE output with respect to the parameters, are called sensitivities. In order to compute the sensitivities reliably and efficiently, we employ a strategy where the sensitivities comprise additional ODE systems, that are solved simultaneously with the original system [42] (see Supplementary material of [51] for details). Because different reaction rate and other parameters can have different orders of magnitude, parameter optimization is done on logarithmic scale. The derivatives on the logarithmic parameter scale are given by the transformation

$$\frac{\mathrm{d}}{\mathrm{d}\theta_j^{\log}} = \theta_j \frac{\mathrm{d}}{\mathrm{d}\theta_j}. \quad (63)$$

If during the optimization, for some parameter values the ODE system cannot be solved and thus the likelihood cannot be evaluated, a very small likelihood value is assigned for that parameter combination.

5.3.2 CVODES for solving ODE systems

The numerical solution of ODE systems is performed using the CVODES solver included in the SUNDIALS package [30], which is high-quality implementation that originates from extensive history of research and development in ODE methods and software. The CVODES solver is meant for both stiff and non-stiff initial value problems given explicitly in the form $\frac{\mathrm{d}\mathbf{y}(\mathbf{t})}{\mathrm{d}t} = \mathbf{f}(t, \mathbf{y}(t), \theta)$ and is capable of computing sensitivities $\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}\theta}$ simultaneously with the original ODE system [30].

In Section 2.1 we introduced the Backward Differentiation Formulae (BDF), that are linear multistep methods (LMM) suitable for the solution of stiff ODE systems. Sophisticated linear multistep implementations like CVODES utilize techniques where the step size h as well as the coefficients α and β of Equation 18 are altered adaptively, which affects the order of the method. It is very hard to derive stability results for such methods, but numerical experiments have indicated that variable coefficient methods are clearly more stable than fixed coefficient formulas when the step size is altered frequently [35]. CVODES utilizes BDF methods in the so-called fixed-leading coefficient form, where the leading coefficient α_0 in Equation 25 is fixed, but other coefficients are changed according to recent history of the step sizes [30]. The q order of the implemented method varies between 1 and 5 [30]. The iterative BDF formula in Equation 25 for this method has the form

$$\sum_{j=0}^{q} \alpha_{i,j} y_{i-j} + h_i \beta_{n,0} f_{i-1} = 0, \quad (64)$$

where $\alpha_{i,0} = -1$ [30]. Because the method is implicit, an nonlinear system needs to be numerically solved on each iteration [30]. For this task, the solver offers several methods based on Newton iteration. For a more detailed description of the methods and some stability discussion, see the original article and [9, 35].

6 Results

In this section we demonstrate the functionality of the novel structure inference strategy that was introduced in the previous section. The performance is tested with experiments involving both simulated and real data. We first formulate ODE model structure inference problems that involve realistically simulated data. These problems are constructed so that the set of all viable models is small enough to permit exhaustive computation of the posterior probability for each model. When the full model posterior distribution is available, running the Metropolis algorithm does not increase the computational burden anymore, which allows us to study how the approximated posterior distribution over the alternative model structures evolves as we run the algorithm. In particular, we wish to test how efficiently the algorithm can find an approximation that has essentially the same information as the full posterior. This is monitored by observing how the overlapping coefficient (OVL) in Equation 51 between the approximation and the full posterior behaves as a function of the approximation support size. Throughout the rest of this thesis, we often refer to the size of the approximation support simply as the number of (evaluated) models. This number is the amount of models that have been proposed and possibly accepted by the Metropolis sampler, and it reflects the computational burden that effectively increases only when a previously unseen model is proposed. In addition, we check how reliable the inferred posterior weights in Equation 41 are, if an approximation of the model posterior is used to make the inferences. To motivate an additional parallelization strategy, we also present how combining information from parallel chains can be beneficial.

We present three different simulated data experiments, one considering only stationary ODE models and the other two involving LEM models with one to three phases. In the LEM experiments, we also demonstrate testing hypotheses about the number of latent states.

To test the novel approach in a real world application, we utilize a data set by [12] and apply the method to infer the structure of a core regulatory network driving Th17 cell differentiation. In this application, the set of possible models cannot be restricted enough to allow brute force inference by evaluating all the viable models, which motivates the use of our efficient strategy. We also apply the concept of profile

likelihood to test identifiability of a well fit LEM model to this data.

6.1 Experiments with simulated data

All the experiments here involve noisy data that is simulated from an ODE model that is one of the viable models under investigation in the inference. The simulated measurement points $y^{\dagger}_{ik}$ are generated so that $y^{\dagger}_{ik} = y_{ik} + \varepsilon(y_{ik}, \alpha, \beta)$, where y_{ik} is the output of the data generating ODE system for component i at time t_k. The added measurement noise has a normal distribution with heteroscedastic variance that has a basal component and a component proportional to the output, i.e.

$$\varepsilon(y_{ik}, \alpha, \beta) \sim N(0, (\alpha + \beta y_{ik})^2). \tag{65}$$

In each experiment, we use fixed noise parameters $\alpha = 10^{-4}$ and $\beta = 0.035$. We assume that this distribution of the measurements is known as well as the values of α and β. We generate the same number of data points for each gene and three replicates at each time point. All experiments involve three genes, and thus for the likelihood in Equation 26 we use the model

$$p(\mathcal{D} \mid \theta, Z) = \prod_{i=1}^{3} \prod_{k=1}^{3\times K} N(y^{\dagger}_{ik} \mid y_{ik}, (\alpha + \beta y_{ik})^2), \tag{66}$$

where K is the number of distinct measurement time points.

All models in these experiments are constructed using the LEM formalism with one to three latent states. For a given model, the ODE system (Equation 10) is constructed by coupling a standard ODE system built from the model mechanisms according to the equations given in Section 2.2 with a latent process $x(t, \theta)$. LEM models with only one latent state are stationary ODE models, and thus for such models we have $x(t, \theta) = 1$. For models with two or three latent states, we use a design based on sigmoidal curves, that have their own parameters. To be explicit, we utilize a latent design process

$$\begin{cases} x_1(t, \lambda_1, \tau_1) = 1 - S(t, \lambda_1, \tau_1) \\ x_2(t, \lambda_1, \tau_1) = S(t, \lambda_1, \tau_1) \end{cases}, \tag{67}$$

for two-phase models, and

$$\begin{cases} x_1(t, \lambda_1, \tau_1, \lambda_2, \tau_2) = 1 - S(t, \lambda_1, \tau_1) \\ x_2(t, \lambda_1, \tau_1, \lambda_2, \tau_2) = S(t, \lambda_2, \tau_1 + \tau_2) - S(t, \lambda_1, \tau_1) \\ x_3(t, \lambda_1, \tau_1, \lambda_2, \tau_2) = S(t, \lambda_2, \tau_1 + \tau_2) \end{cases}, \tag{68}$$

for models with three phases, where

$$S(t, \lambda, \tau) = \frac{1}{1 + \exp(-\lambda(t - \tau))}. \tag{69}$$

The function in Equation 69 is a sigmoidal curve rising from 0 to 1 such that λ defines how rapidly the rise occurs and τ the time of this rise. For a given model

with m latent states, the parameter vector θ contains not only the rate parameters of the model mechanisms, but also the extra $2(m-1)$ latent process parameters. When performing model calibration, these extra parameters are also optimized along with the rate parameters.

In each experiment, we use a uniform prior distribution over models, meaning that $p(Z \mid \mathcal{D}, H) \propto p(\mathcal{D} \mid Z, H)$. Furthermore, we use fixed initial values $y_i(0) = 0.01$ for each $i = 1, 2, 3$. These initial values are small compared to the magnitude of the measurements, which is motivated by the fact that the experimental data used in this study has the same property. The initial values are assumed to be known exactly. All experiments involve some mechanisms that are fixed, i.e. appear in all feasible models. We note that the parameters of these fixed mechanisms are nevertheless estimated from the data.

6.1.1 Experiment 1

In the first experiment, we consider standard ODE models with three genes. We denote the genes by A, B and C and for each gene, we simulate three replicates of measurements at time points 0.25, 0.5, 1, 2, 3, 4, 5, 6 and 7. We consider only models where each gene is affected by basal activation and degradation. These mechanisms are thus included in the ODE system of each model. Further, we allow the genes to interact through all possible activating and inhibiting mechanisms, excluding autoactivation and autoinhibition. The model space can then be expressed as the union of all 12×1 binary matrices, where each row corresponds to one activation or inhibition link. There are thus $2^{12} = 4096$ different models. The activation and inhibition links of the data generating model as well as the generated data are shown in Figure 9. Because all the models are stationary ODE models, the only model parameters are the rates of basal activation and degradation for each gene, along with the rates of model-specific activation and inhibition mechanisms between the genes. The kinetic rate parameter values used in the data simulation are in Table 2. In the maximum likelihood optimization, we set the allowed range of the parameters to $[0.001, 30]$ for all rates.

The Metropolis algorithm is started from the empty model, i.e. from a 12×1 matrix of zeros. The algorithm was set to terminate when the overlap between the obtained approximative distribution and the full model posterior distribution passes 99%. This happened after 3550 iterations, after which 339 models were in the support of the approximative distribution, i.e. had been proposed and possibly accepted. Figure 10 displays the overlapping coefficient (OVL) between the posterior approximation and the real posterior as a function of the number of evaluated models during the course of the Metropolis algorithm. We see that the overlap approaches one and is very close to it after 300 models. This means that the algorithm finds the models with high posterior probability and only a fraction of models is needed to obtain a good posterior approximation. Figure 10 also presents the posterior weights of each link computed with approximations obtained after 80 and 100 evaluated models along with the weights computed using the full posterior distribution. We see that already after 80 models, when the overlapping coefficient is only 0.35, we have

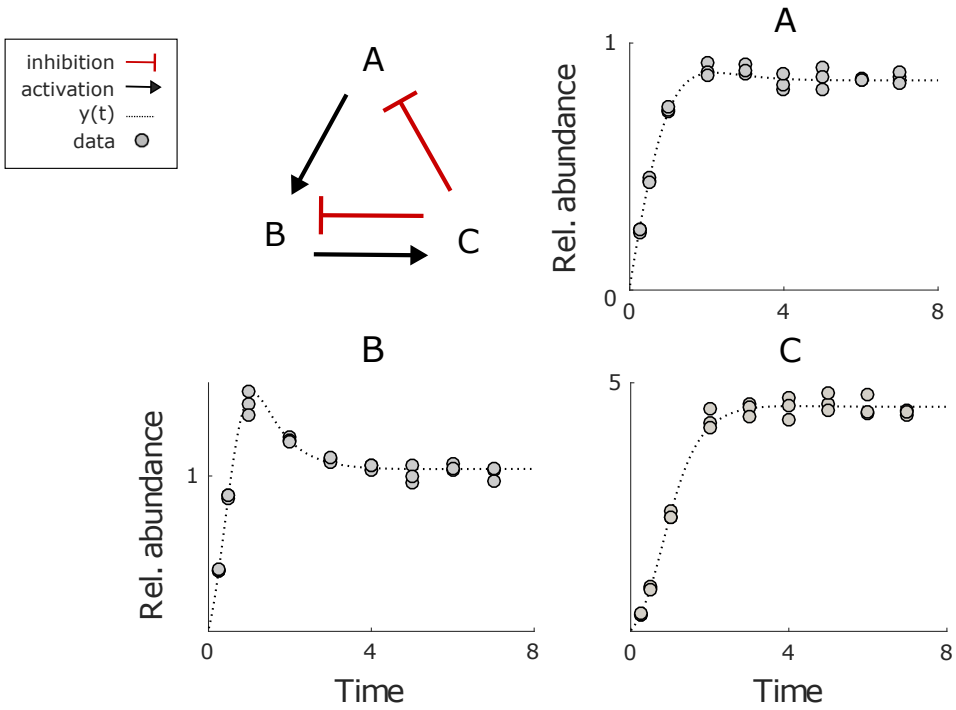


Figure 9: Illustration of the stationary data generating model and the simulated data used in Experiment 1. The black arrows and red turnstiles represent activating and inhibiting links, respectively. In addition, each gene experiences a basal activation and degradation. The dotted line represents the underlying model response $y(t)$ for each gene A, B, and C. The noisy measurements, which are relative abundances of the genes at time points $t = 0.25, 0.5, 1, 2, 3, 4, 5, 6, 7$, are plotted using gray circles.

inferred most of the posterior weights rather accurately. However, there is a notable error in the predictions for the mechanisms for which the full posterior distribution does not give a prediction close to 0 or 1 ($B \to A$ and $C \to A$). After evaluating 100 models, the overlapping coefficient has reached 0.80, and all predictions are very close to the ones computed using the full posterior information.

The results of this experiment demonstrate that for a standard ODE model structure inference problem, the algorithm indeed can provide a good model posterior approximation with a computational effort that is only a fraction from the effort required to compute the full model posterior distribution. It is clear that if this approximation is good, then also the structure inference is reliable. However, the results show that it is possible to obtain somewhat accurate predictions about the model structure even when the obtained approximation only partly overlaps with the full model posterior.

The inference results in Figure 10 show that the inference gives a posterior weight close to one for those mechanisms that actually were in the data generating model. This highlights the practicality of the BIC as a marginal likelihood estimation technique. Additional experiments where the data set was generated with different random seeds revealed that this is not a lucky coincidence, assuming that the data are informative enough, i.e. the noise parameter values are small enough. In this

Table 2: Kinetic rate parameter values of the data generating models in each simulated data experiment.

Mechanism	Experiment 1	Experiment 2	Experiment 3
basal activation of A	1	1.5	1
basal activation of B	1	-	-
basal activation of C	1	-	-
activation $A \to B$	5	3	3
activation $A \to C$	-	-	0.5
activation $B \to C$	2.5	2	-
activation $C \to B$	-	1.5	1.5
inhibition $C \dashv A$	0.15	-	-
inhibition $C \dashv B$	1	-	-
degradation of A	0.5	0.5	0.3
degradation of B	0.5	2	1
degradation of C	0.8	3	1.5

Table 3: Latent process parameter values of the data generating models in the experiments involving multiphase LEM models.

Interpretation	Parameter	Experiment 2	Experiment 3
rate of first state transition	λ_1	1.5	2.5
time of first state transition	τ_1	6	3
rate of second state transition	λ_2	-	1
time between state transitions	τ_2	-	5

experiment, the real data generating model actually received the highest BIC value of all 4096 models. Also test with different noise levels were run (data not shown), which showed that the less noise in the data, the more likely BIC ranks the real model among the best ones.

6.1.2 Experiments 2 and 3

In the second and third experiment, we consider LEM models that have from one to three latent states. We generate three replicates of measurements at time points $1, 2, 3, 5, 8, 12, 18, 24$ and compare hypotheses H_M: "There are M latent states." for each $M = 1, 2, 3$. The experiments involve three genes A, B, and C, and we assume that basal activation of A and degradation of each gene are fixed mechanisms. We construct the models from the four activations links shown in Figure 11a. Under the hypothesis H_M, the full model space contains all possible $4 \times M$ binary matrices Z, where the element $\{Z\}_{jk}$ determines if the mechanism $j \in \{1, 2, 3, 4\}$ is active in the latent state $k \in \{1, \ldots, M\}$ (see Section 2.3). Therefore, there are $2^{4 \times M}$ different

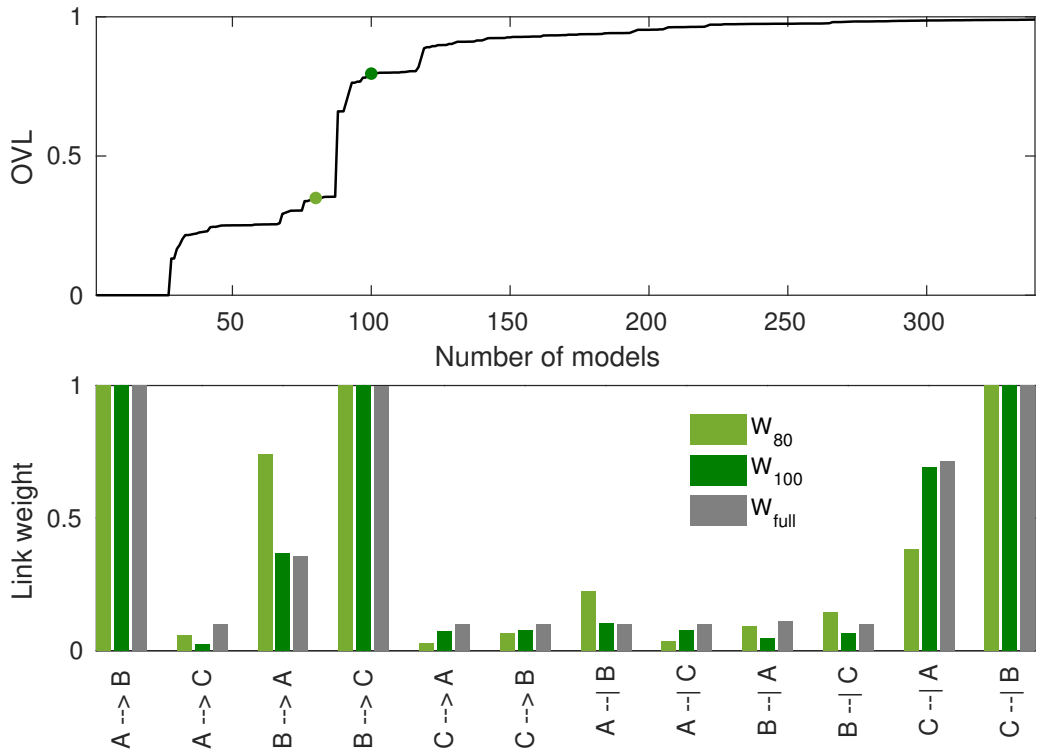


Figure 10: Visualization of the model structure inference in Experiment 1. The top panel shows the value of the overlapping coefficient between the actual full model posterior and the approximation obtained at different amounts of evaluated models. In the bottom panel, the bars correspond to link weights W_{80} and W_{100} after evaluating 80 and 100 models, respectively, along with the actual weights W_{full} computed using the full model posterior. After 80 models we have OVL ≈ 0.35 and after 100 models OVL ≈ 0.80. The total number of possible models is 4096.

viable models under the hypothesis H_M.

In Experiment 2, the data are generated from a two-phase model and, in Experiment 3, from a three-phase model. This allows us to study if the comparison of hypotheses works as expected. Illustrations of these data generating models are in Figure 11b and Figure 11c. The kinetic rate parameter values used in the data generation are in Table 2 and the used latent process parameters in Table 3. The resulting noisy data sets and the used latent processes are shown in Figure 12.

For kinetic rate parameters, the allowed range is $[0.001, 50]$ in Experiment 2 and $[0.001, 30]$ in Experiment 3. In both experiments, the range for the softness parameters λ_1 and λ_2 is $[0.75, 3]$. This allows both quite rapid and slow changes between the latent states. If the model has three phases, then the allowed range is $[1, 12]$ for the first state transition time τ_1 and $[3, 12]$ for τ_2, which describes the length of the time interval between the two state transitions. These bounds ensure that all the three states are clearly present with any parameter combination. Otherwise, if the data should favour models with only one or two phases, optimization of the latent parameters could shrink a three-phase model into something that is effectively a

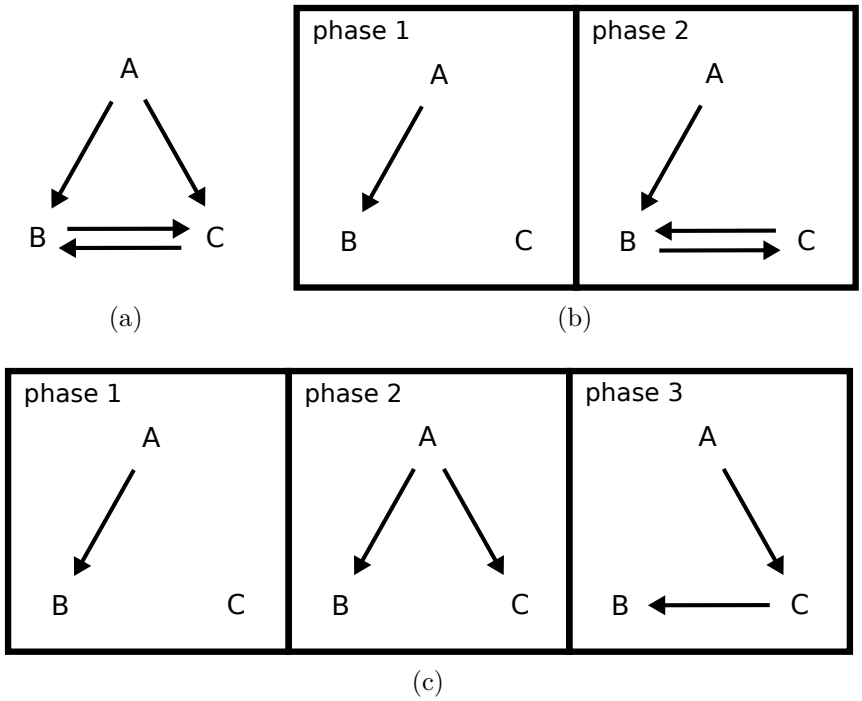


Figure 11: Illustration of the possible activation mechanisms in the simulated data experiments that involve LEM models. (a) All possible activation links. In addition to these interactions, it is assumed that gene A has basal activation and all genes are allowed to degrade at some unknown rate. (b) Activation mechanisms of the two-phase data generating model used in Experiment 2. (c) Activation mechanisms of the three-phase data generating model used in Experiment 3.

model with less states. This is because any component of Equation 68 can be forced to have only negligibly small values on the studied time interval if the parameter range is arbitrary. For two-phase models, only τ_1 is needed, and we use the range $[1, 24]$ for it. Similarly, this range forces both components of Equation 67 attain a meaningful strength on the time scale of the data set. We also note that if the softness parameters τ were allowed to have arbitrarily large values, serious overfitting could occur at least for models with many parameters that are non-indentifiable with respect to the data. In general, the allowed parameter ranges in this study are a compromise between flexibility of the latent dynamics and a reasonable physical interpretation. The used latent process formulation with extreme parameter values allows negative values for the second component in Equation 68, but within our bounds these negative values are negligibly small.

For each hypothesis H_M, $M = 1, 2, 3$, the Metropolis chain in the corresponding model space is started from the $4 \times M$ matrix of zeros. Figure 13 shows the overlap of the full model posterior distribution and the approximation obtained after different amounts of evaluated models for each $M = 1, 2, 3$. Again, the algorithm was halted when the OVL reached 0.99. In the case $M = 1$, there are only 16 models, and at some point the OVL jumps effectively from 0 to 1, because one model is clearly

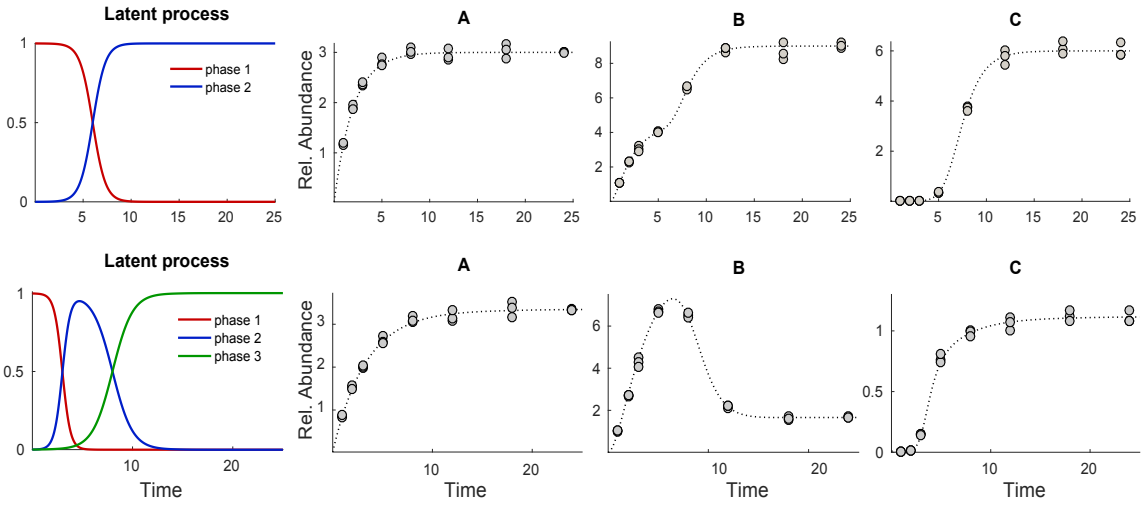


Figure 12: Simulated data sets used in Experiments 2 and 3. The top row exhibits the two-phase latent process used in Experiment 2 along with the output of the model in Figure 11b and the simulated noisy measurements of the relative abundances of genes A, B and C. The bottom row displays the three-phase latent process, output of model in Figure 11c and simulated data used in Experiment 3. The dotted lines are the model outputs and the filled gray dots represent the simulated measurements at time points $t = 1, 2, 3, 5, 8, 12, 18, 24$.

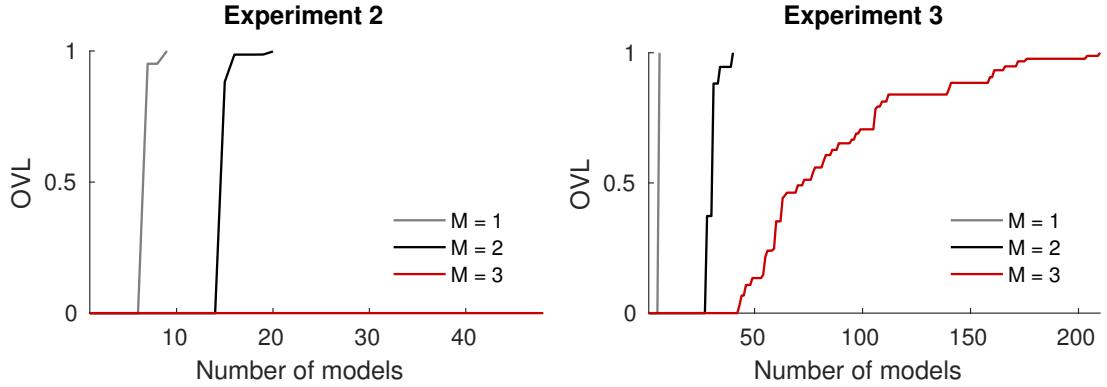


Figure 13: Overlap between the obtained approximative distribution and the full model posterior distribution in the LEM model experiments. The lines indicate the value of the overlapping coefficient as a function of evaluated models for the cases $M = 1, 2, 3$ in Experiments 2 and 3.

dominating the whole posterior mass. In the cases $M = 2$ and $M = 3$, there are 256 and 4096 models, respectively, and we notice that an overlap close to one is reached by having to evaluate only a relatively small proportion of the models. Exact run lengths and numbers of evaluated models under each hypothesis in both experiments are in Table 4. In Experiment 2, the algorithm for $M = 3$ does not terminate after 10^4 iterations because the sampler gets stuck in a local optimum where it can only

Table 4: Statistics about the runs of the Metropolis algorithm in the LEM model experiments involving simulated data. The algorithm was set to halt when the overlap between the approximative distribution and the full model posterior reaches 99%. The numbers indicate how many iterations this took and how many models had to be evaluated during the course of the Metropolis algorithm. The latter number reflects the computational burden. Total number of models was 2^{4M} under a hypothesis H_M.

	Experiment 2			Experiment 3		
Hypothesis	H_1	H_2	H_3	H_1	H_2	H_3
Iterations	16	34	10^4	10	100	648
Models evaluated	9	20	48	6	40	210

Table 5: Posterior probabilities for different hypotheses H_m about the number of latent states M in the LEM model experiments involving simulated data.

	$p(H_1 \mid \mathcal{D})$	$p(H_2 \mid \mathcal{D})$	$p(H_3 \mid \mathcal{D})$
Experiment 2	0	0.9955	0.0045
Experiment 3	0	$9.8 \cdot 10^{-223}$	1.0000

escape with a very low probability (or even zero due to numerical limitations). This issue could be avoided by starting multiple chains from different initial models (see Section 6.1.3 below) or by using a proposal distribution that can propose moves also to k-neighbors with $k > 1$ (see Equation 46).

To perform the hypothesis comparison task, we assume a uniform prior distribution over the hypotheses and compute the posterior probabilities $p(H_M \mid \mathcal{D})$ for each $M = 1, 2, 3$ using Equation 38 and Equation 37. In both experiments, these values are computed from the full posterior distributions and are shown in Table 5. The values indicate that the hypothesis testing works as expected, since in both experiments, the correct hypothesis gets virtually all of the posterior probability mass. One could draw the conclusion that the BIC is able to perform the inference correctly, since in Experiment 2, it rules out one-phase models that do not fit the data well, yet penalizing the overly complex three-phase models suitably. In Experiment 3, the BIC and the resulting posterior inference over hypotheses indicate that models with three phases fit the data much better than other models. Here, the values $p(D \mid H_M)$ were computed using the full model posterior, but since the approximations given by the novel search strategy are good, it is clear that the same result is obtained without having to compute the full posterior exhaustively.

6.1.3 Combining information from multiple independent chains

The previous experiments demonstrated that a single Metropolis-type sampler can indeed find the high-probability models and thus give a reliable approximation for the model posterior distribution. However, as stated in the previous section, it is possible that a metropolis chain gets stuck in a local optimum and does not escape it in a finite number of iterations. The inference results can then be different for

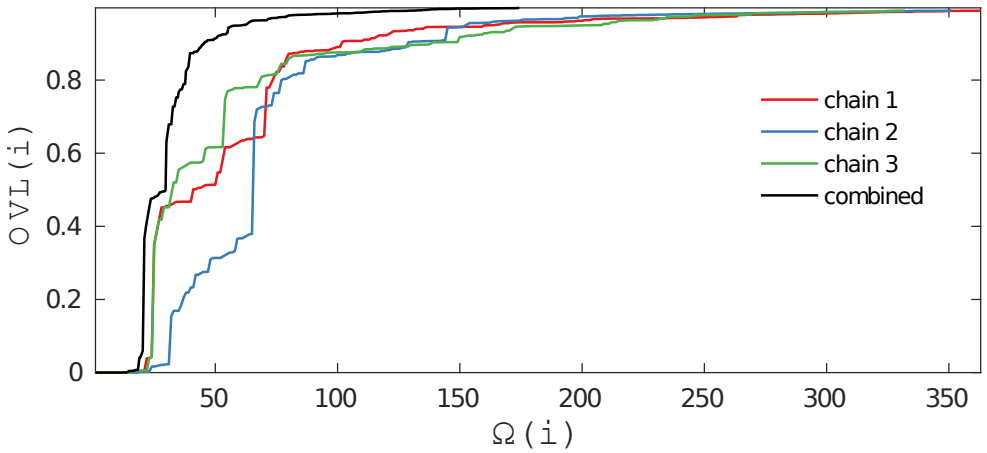


Figure 14: Illustration of the benefit obtained by using multiple parallel chains. The coloured traces represent the overlapping coefficient between the posterior approximation given by that chain and the full model posterior as a function of $\Omega(i)$ which is approximately proportional to the computation time after iteration i . The black trace represents the OVL between the full model posterior and the approximation created by combining the information of the independent chains. The approximation computed using information from all the three parallel chains reaches overlap of one in a shorter time.

chains that are started from different initial models. In order to obtain more reliable results, one can start several independent chains in parallel, possibly from different initial models, and create the posterior approximation using all evaluated models from each chain.

Comparing different simulated chains is a common strategy for assessing the convergence of MCMC runs in continuous spaces [20]. For example, when sampling a multimodal distribution, results have a higher chance of being reliable, if all chains yield similar samples. On the other hand, if one chain has only sampled one mode and another chain has only sampled another mode for the same number of iterations, one cannot combine the samples. This is because the combined set of samples cannot generally be seen as a sample from the target distribution, since we have equally many samples from both modes, even though one of the modes might have a considerably larger total probability mass. However, in the case of a discrete model space, our information consists of the posterior probabilities for each model that has been sampled at least once by at least one chain. All this information can now be used to create the posterior approximation. This can be especially benefical, if the independent chains can be run in parallel.

We demonstrate this strategy by returning to the model structure inference problem and the data set considered in Experiment 1 (see Figure 9). We start three independent chains from randomly chosen initial models, and test how much faster we would reach a good OVL if the chains were run in parallel and their information was combined after each iteration. In order to do this, we define a quantity $\Omega(i)$,

which for any MCMC method in the discrete model space is given by

$$\Omega(i) = \frac{\text{Total number of evaluated models after iteration } i}{\text{Number of parallel chains used by the method}} \tag{70}$$

and therefore approximately proportional to the computation time. For the method of using only a single chain, this is obviously just the number of evaluated models, which was used to represent computational burden in the earlier experiments. Figure 14 shows the OVL between the full model posterior and the approximations obtained by each chain on their own and the method of combining their information as a function of $\Omega(i)$. Clearly, the latter method provides better results in a shorter time.

6.2 Applying the method to Th17 cell differentiation data

The experiments presented in the previous section all involved a simulated data set and a model space small enough to be exhaustively evaluated for reference. We now move on to an application in which the model space cannot be restricted to such a limited set of simple networks. This application utilizes a data set consisting of mRNA measurements of genes that are involved in the Th17 cell differentiation program [12]. Th17 cell differentiation has been shown to occur in three sequential phases [65], and the LEM model has been applied to model the core regulatory network that drives the differentiation [34]. In the earlier study [34], model selection between the alternative model configurations was performed using a greedy forward-backward stepwise search. This approach resulted in an inference consisting of a single model and its calibrated parameters. This thesis extends the obtained results by exploring the alternative models and condensing information about a large set of high-probability models into probabilistic predictions about the mechanisms. To provide some background information to be considered in the application, this section starts from the basic concepts involved in T cell differentiation, referring to [2].

6.2.1 Vertebrate immune system

Developed living organisms have an immune system that protects them from infections that otherwise would be deadly. This system is capabable of recognizing and reacting to various foreign macromolecules, collectively called pathogens. Vertebrates depend on both innate immune system, which includes general defense reactions, and a more sophisticated adaptive immune system. The cells that are responsible for adaptive immune response are called lymphocytes, and they belong to a class of white blood cells. Two main classes of lymphocytes are B cells, which mature in the bone marrow, and T cells that mature in the thymus.

Lymphocytes can exist in the form of naïve cells, effector cells, or memory cells. In an adaptive immune response, a foreign substance, an antigen, causes some of the naïve B and T cells to multiply and differentiate into effector cells. These have different but equally important functions, as effector B cells secrete antibodies and effector T cells express a variety of mediators called interleukins (IL), or cytokines. Part of the naïve cells proliferate and mature into memory cells, which in the future

are more effective against the same antigen. Immunological memory depends on both lymphocyte proliferation and differentiation.

T cells can be further divided into different classes, which include cytotoxic T cells, helper T cells and regulatory T cells. Effector cytotoxic T cells kill infected cells and effector regulatory T cells can repress the activity of other immune cells. Effector helper T cells participate in stimulating the responses of other immune cells, such as B cells and cytotoxic T cells. Helper T cells are also known as $CD4^+$ T cells, according to the protein CD4 found on their surface. These cells have a major role in the regulation of the human immunity system [62]. Different well characterized subsets of effector $CD4^+$ T cells, which can be separated by different cytokine expression profiles and immune regulatory function, are Th1, Th2 and Th17, and induced regulatory T cells [39, 66].

6.2.2 Th17 cell differentiation

Differentiation of a naïve $CD4^+$ T cell towards one of the effector T helper (Th) cell lineages begins, when it encounters an antigen in the presence of cytokine signals [66]. The type of the present cytokine signals determines the lineage towards which cells will develop [66]. For example, cells in precence of IL12 develop into Th1 whereas presence of IL4 directs the development of Th2 cells [2]. In this study, we focus on the Th17 subset, which requires two cytokines for its differentiation. These are the transforming growth factor β (TGFβ) and interleukin 6 (IL6) [2, 39]. Th17 cells express IL17 and are important in the control of some infections and in wound healing [2]. Their discovery [28, 47] has shaped our understanding of the pathogenetic basis related to various immune-mediated diseases, such as psoriasis, rheumatoid arthritis, multiple sclerosis, inflammatory bowel disease, and asthma [59, 62].

Differentiation programs of all Th cell subtypes involve a network of transcription factors consisting of positive and negative regulation mechanisms [66]. The main transcription factors involved in the network that drives Th17 differentiation are the signal transducer and activator of transcription 3 (STAT3) and the retinoic acid receptor-related orphan receptor gamma t (RORγt) [66]. The latter is usually considered the master regulator of the Th17 lineage and it is induced in naive $CD4^+$ T cells within 8 hours after stimulation in the presence of TGFβ and IL6 [66]. Also various other transcription factors such as the interferon regulatory factor 4 (IRF4) and the basic leucine zipper ATF-like transcription factor (BATF), are needed for the full differentiation program [12].

Differentiation of naïve helper T cells to effector Th17 cells is a very complex process involving many interacting molecular species, and the roles of different transcription factors involved in it remain unknown. However, many experimental data sets show that the key transcription factors exhibit interesting dynamics during the course of differentiation [12, 33]. Thus, dynamic mathematical modeling can reveal simplified, yet interesting dynamics that drive the differentiation. Data-driven mathematical modeling is capable of capturing these dependencies and dynamics that originate from the molecular kinetics, and it has been applied recently [33, 34].

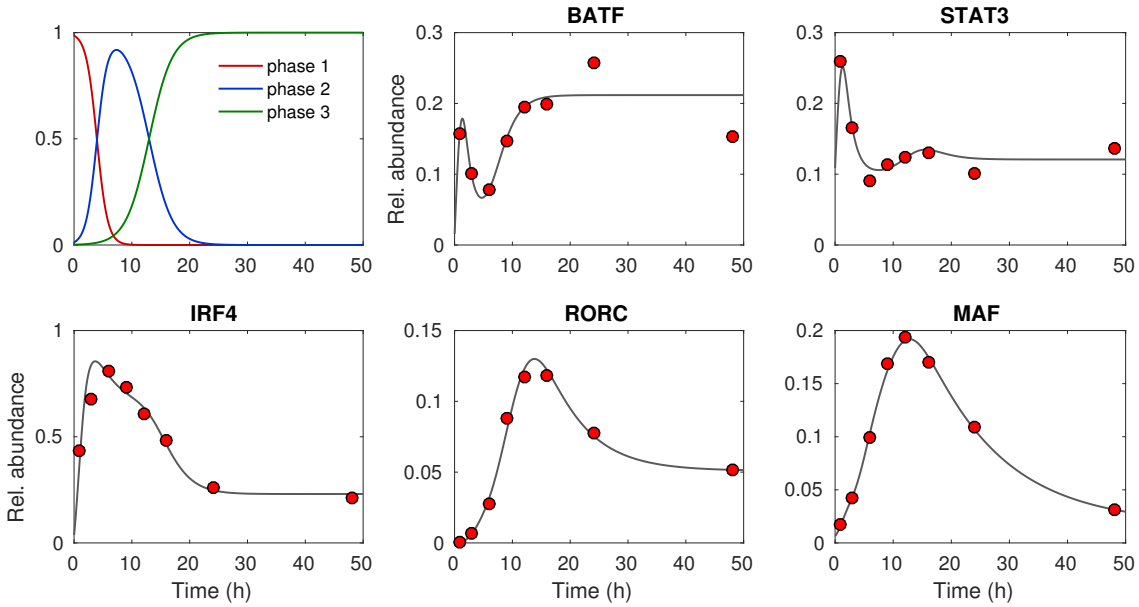


Figure 15: Illustration of the used experimental data set and best found model. Relative values of the FPKM measurements at time-points 0, 1, 3, 6, 9, 12, 16, 24, and 48 hours are plotted with using red dots. Measurements at $t = 6$ h are averages of three replicates, and measurements at $t = 0$ h are not shown. The gray curves represent the output of the best found model with its maximum likelihood parameters. The red, blue and green curves of the top-left panel represent the latent process for the best model.

6.2.3 Experimental data

The core gene regulatory network studied in this thesis consists of five transcription factors RORγt, STAT3, IRF4, BATF and transcription factor Maf (MAF). The network was experimentally derived in [12], and the corresponding genes that encode these proteins have the same names, except for RORC that encodes RORγt. The data set, also provided by [12], consists of measured RNA fragments per kilobase per million (FPMK) values at time-points 0, 1, 3, 6, 9, 12, 16, 24, and 48 hours. The measurements were done by purifying cells from lymph nodes and spleen of wild-type mice, culturing the cells in Th17 conditions and harvesting a proportion of cells at the mentioned time points to perform RNA sequencing [12]. At time $t = 6$ h, there are three replicates but we treat those as a single measurement that is the mean of three replicates in order to get results that can be compared with the findings of the earlier study [34]. The FPKM values are divided by 1000 before the inference. The resulting relative measurements of each gene are shown in Figure 15. The initial values for the ODE models are not estimated from data, but instead fixed according to the measurements at time $t = 0$ h. Consequently, these measurements at $t = 0$ h are not considered as part of the data in likelihood computations.

6.3 Results for the Th17 application

In this application, the set of alternative models is extremely large, and we face very heavy computations. We present results of the structure inference for the core regulatory network that steers Th17 cell differentiation, under different hypotheses about the number of phases in the differentiation. Furthermore, for one specific model, we analyze the uncertainty of the model parameters by testing parameter identifiability using the profile likelihood approach [40].

6.3.1 Network structure inference

All 15 possible activation, inhibition and synergistic activation mechanisms motivated by [12] are shown in Figure 16a. We construct the alternative LEM models by first setting the fixed part of the network such that each gene degrades at a constant rate and experiences a basal activation. An exception to this is RORC for which we do not allow basal activation. If the model has M latent states, it can then be determined by a $15 \times M$ binary matrix Z, where each row corresponds to one of the mechanisms shown in Figure 16a (see Section 2.3).

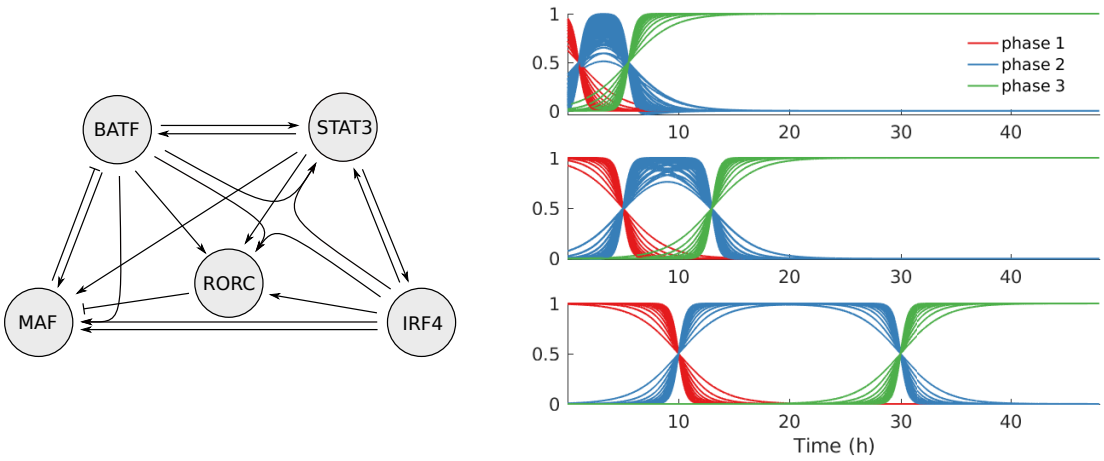
The LEM model for a known configuration Z is again built by first constructing the ODE according to rate laws explained in Section 2.2. If the model has multiple phases, this ODE system is then coupled with a latent process as explained in Section 2.3. The latent process for multiphase models is again built using Equations 67 and 68, similarly as in the simulated data experiments. We use the parameter bounds $[0.001, 100]$ for kinetic rate parameters and $[0.5, 3]$ for λ_1 and λ_2. Furthermore, for models with three latent states, we use the bounds $[1, 20]$ for τ_1 and $[4.5, 20]$ for τ_2 and for models with two latent states, our bounds for τ_1 are $[1, 40]$. Possible realizations of three-phase latent processes allowed with our formulation and the above parameter ranges are demonstrated in Figure 16b.

In likelihood computations, an underlying assumption that we make is that the measurements are normally distributed with heteroscedastic variance. The error is modeled by Equation 65, and we fix the parameter values $\alpha = 10^{-4}$ and $\beta = 0.035$. Also an approach where these parameters are estimated simultaneously with the model parameters was applied, but it was discarded since models tended to fit strongly towards the first few data points and give very small values for α. Since the data contains 8 measurements of all the 5 genes, the likelihood in Equation 26 takes the form

$$p(\mathcal{D} \mid \theta, Z) = \prod_{i=1}^{5} \prod_{k=1}^{8} N(y_{ik}^{\dagger} \mid y_{ik}, (\alpha + \beta y_{ik})^2), \tag{71}$$

where y_{ik} is the output of model Z for gene i at time t_k, when the parameters take values θ. The corresponding measurement is denoted by $y_{ik}^{\dagger}$.

The inference is performed separately for the three different hypotheses about M, the number of phases in the cellular differentiation. The number of alternative models in each case is $2^{15 \times M}$ ($\approx 3.3 \times 10^4$ for $M = 1$, 1.1×10^9 for $M = 2$ and 3.5×10^{13} for $M = 3$), which means that exhaustive computation of the full model posterior is not computationally feasible. Since we do not have the full model posterior for reference,



(a) (b)

Figure 16: Illustration of the allowed network mechanisms and possible latent process realizations in the Th17 cell application. (a) Full network of all the allowed activation, inhibition and synergistic activation mechanisms between the five genes. Activation mechanisms are represented by single arrows, inhibition mechanisms by turnstiles, and synergistic activation mechanisms by arrows with combined heads. Figure from [34]. (b) Examples of three-phase latent processes possible within the allowed parameter ranges. The top panel is the extreme case where both τ_1 and τ_2 have their minimum values. In each panel, the curve families are drawn using a fixed value for τ_1 and τ_2, and altering the values of λ_1 and λ_2 within the range $[0.5, 3]$.

we tackle the model structure inference problem by starting multiple independent Metropolis samplers, and check if the model posterior approximations (Equation 50) provided by the independent chains converge close to each other. The idea behind this approach is that if two independent chains give similar posterior distribution approximations, they likely have explored the same high probability regions of the model space. The more chains have found the same high probability region, the less likely it is that there exist high probability models that have not yet been found. This has an analogy to traditional MCMC convergence diagnostics, which also rely on monitoring several independent sequences [20]. Each chain is started from the empty model with different random seeds. This starting point can be motivated by a guess that the high probability models to have more zeros than ones. This guess however does not have enough basis to be included this in the prior distribution over the model configurations, and instead we again use a uniform prior. We start four independent chains in the case $M = 3$, and two independent chains in the cases $M = 1$ and $M = 2$.

Statistics about the MCMC runs are shown in Table 6. In the space of one-phase models, we are able to run the chains until a million iterations, since the chains mostly move inside a relatively small set of high probability models. For two and three-phase models, the chains reached 4000-8000 iterations, during a five-day run

Table 6: Statistics about the MCMC runs in the Th17 cell application under alternative hypotheses about the number of phases in the differentiation. The numbers indicate how long each chain was run and how many models the chains found. Furthermore, the proportion of accepted MCMC moves is given for each chain.

Hypothesis	H_1		H_2		H_3			
Chain	ch. 1	ch. 2	ch. 1	ch. 2	ch. 1	ch. 2	ch. 3	ch. 4
# Iterations	10^6	10^6	7315	7987	4174	4420	4271	4421
# Models	4561	4478	3205	3410	3722	3860	3771	3908
Acceptance rate	0.164	0.148	0.147	0.136	0.243	0.228	0.237	0.229

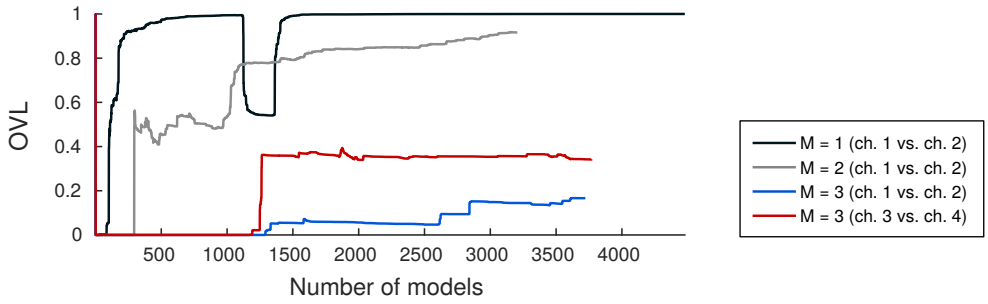


Figure 17: Illustration of the converge of different chains in the Th17 core network structure inference under different hypotheses about the number of phases in cellular differentiation. Lines represent the overlapping coefficients (OVL) between approximative model posterior distributions obtained from different independent chains as a function of evaluated models.

on a high performance computing cluster with 12 cores. Each independent chain covered 3000-4000 evaluated models. The overlapping coefficients (OVL) between the two independent chains for $M = 1$ and $M = 2$, and between chains 1 and 2 as well as chains 3 and 4 for $M = 3$, are shown in Figure 17. We notice that in the first two cases the OVL approaches one and we get model posterior approximations that agree well with each other. Thus, it is likely that the algorithm has explored the high probability regions accurately in both cases. In the first case, there is a temporary drop in the OVL trace when one chain moves to a new high posterior probability region and the other chain finds it only later. For $M = 3$, the model space is very large and our approximations obtained by independent chains only partially overlap after approximately 3700 evaluated models.

The posterior weight approximations for each mechanism in each phase from the different chains are presented in Figure 18. For one- and two-phase models, the weights are naturally similar since our posterior approximations overlap remarkably. In addition, for the four chains that explore three-phase models, the obtained posterior weights are rather close to each other even though the OVL between the corresponding

model posterior approximations does not reach one. The simulated data experiments, where good approximations to the weights were obtained even when the model posterior approximation did not fully overlap with the real model posterior, support the belief that the algorithm has provided us with meaningful information about the model structure given the mechanisms included in the network.

Hypotheses about the number of phases M are compared by computing the corresponding posterior probabilities $p(H_M \mid D)$, where H_M suggest that there are M different phases. The hypothesis H_3 gets practically all the posterior probability mass ($P(H_3 \mid \mathcal{D}) \approx 1$), since models with three phases fit the data much better than ones with only one or two phases. Output of the best ranking model is shown against the data in Figure 15. The high-probability models under hypothesis H_3 have many common properties, such as the positive feedback loop of BATF and STAT3 in the initial phase. The best models also have very similar parameter values for the fitted latent process: the second phase is activated at time 4 h and third at around 13 h.

To summarize, the results of this analysis support the earlier findings that Th17 lineage specification occurs in three sequential phases [65]. In addition, the posterior weights for the different mechanisms (Figure 18) obtained here with the novel and efficient search strategy coincide with the point estimate that was obtained in the earlier LEM model analysis [34] through a greedy search.

6.3.2 Identifiability results

To assess the uncertainty in the model parameter estimation and identifiability of the model parameters (Section 4.3), we apply the profile likelihood [40, 49] approach to some of the best found models. Given a fixed model configuration, the profile likelihood for the parameter θ_j is defined by

$$\mathrm{PL}_j(p) = \max_{\theta \in \{\theta \mid \theta_j = p\}} \log p(\mathcal{D} \mid \theta), \tag{72}$$

where $p(\mathcal{D} \mid \theta)$ is the likelihood function (Equation 26) [40]. This means that the parameter profile is a one-dimensional function, where the value $\mathrm{PL}_j(p)$ is the log likelihood computed with parameters θ, where θ_j is fixed to p and θ_i, $i \neq j$ are reoptimized. Profile likelihood can be employed to assess the uncertainty of the parameter estimated by computing confidence intervals for the parameters. For confidence level α, the confidence interval is the set

$$\mathrm{CI}_j(\alpha) = \{p \mid -2\ \mathrm{PL}_j(p) \leq \min_{\theta} -2 \log p(\mathcal{D} \mid \theta) + \Delta(\alpha)\}, \tag{73}$$

where $\Delta(\alpha)$ is the corresponding threshold [40]. When the amount of data approaches infinity, the threshold is given by

$$\Delta(\alpha) = \mathrm{icdf}(\chi^2_1, \alpha), \tag{74}$$

where $\mathrm{icdf}(\chi^2_1, \cdot)$ is the inverse cumulative distribution function of the χ^2-distribution with one degree of freedom [40].

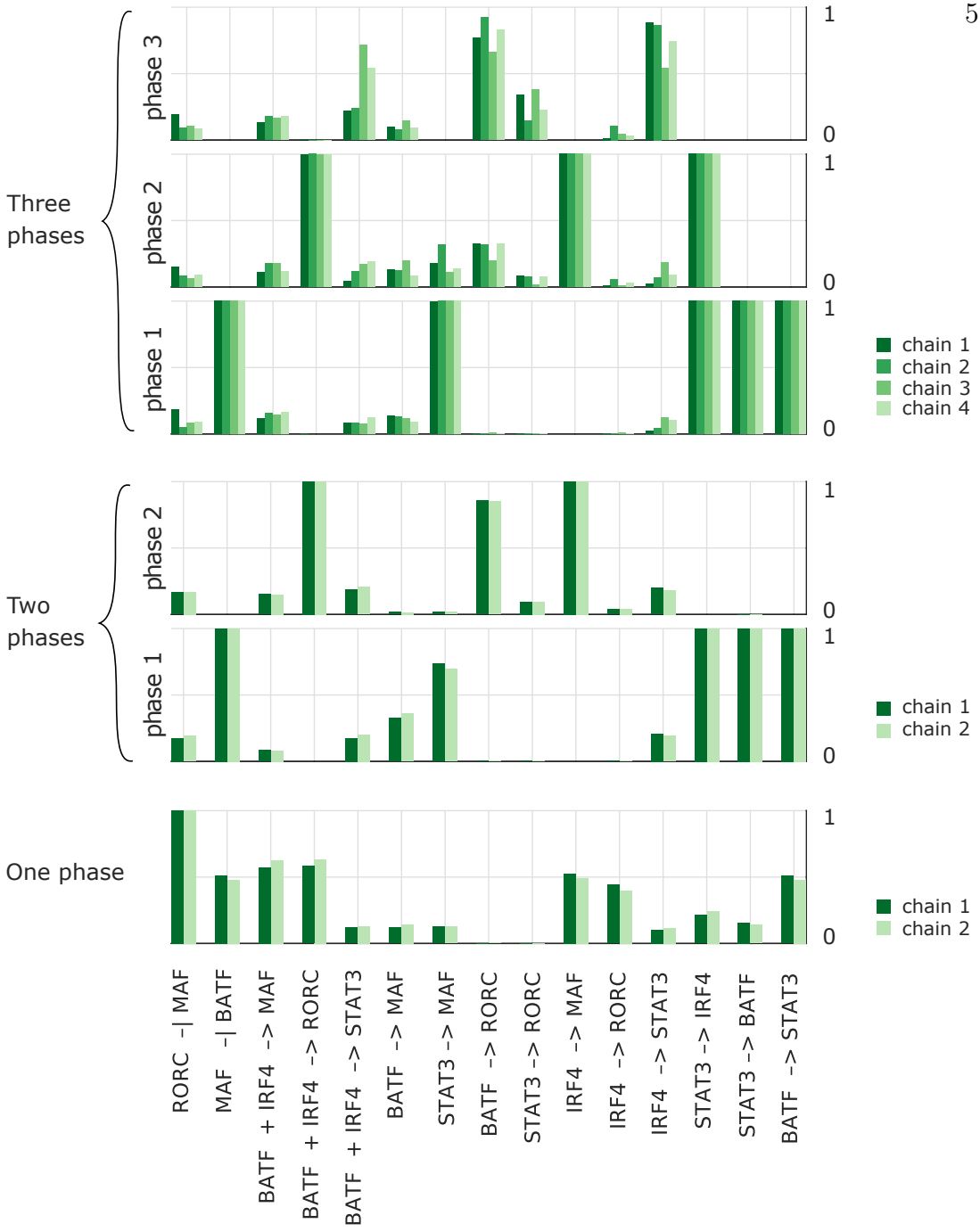


Figure 18: Th17 core network inference results under different hypotheses about the number of distinct phases in the cellular differentiation. The bar length represents the posterior weight of the corresponding mechanism in the corresponding phase. Two independent chains were run for one- and two-phase models, and four chains for three-phase models.

Idea of the profile likelihood is that if the parameter profile is flat for θ_j, the parameter is structurally non-identifiable, since any change in it can be compensated by changing other parameters θ_i, $i \neq j$ [40]. If the profile likelihood entails a unique minimum, it is either identifiable or practically non-identifiable. The latter case is

Table 7: The extended parameter bounds used in parameter optimization of profile likelihood computations.

	Kinetic rate params.	λ_1	λ_2	τ_1	τ_2
Allowed range	$[10^{-5}, 5 \cdot 10^4]$	[0.05,6]	[0.05, 6]	[1, 40]	[0.45,40]

indicated by a profile likelihood curve that does not exceed the threshold in at least one direction [40].

Profile likelihood computations are very expensive compared to normal model parameter calibration, since if the parameter dimension is d, and we wish to compute all the parameter profiles $\mathrm{PL}_j(p)$ at a grid of points $p \in \{p_{1j}, \ldots, p_{kj}\}$, we have to solve a $d-1$-dimensional optimization problem dk times. In addition, finding a suitable grid from the neighborhood of the maximum likelihood parameter is not straightforward.

The profile likelihood is computed for the five best ranking three-phase models obtained from the four chains. Because our optimization involves defining bounds for the parameters, it is possible that a change in the profiled parameter cannot be compensated enough. This can result in a profile likelihood that indicates identifiability in cases where the threshold is only exceeded because the parameters are limited by their optimization bounds. In order to avoid this, we extend the normal parameter bounds remarkably. The used bounds are in Table 7.

Figure 19 shows profile likelihood plots for the best found model along with the 95% confidence threshold. It can be seen that most of the parameters are identifiable, but four of the reaction rate parameters are practically non-identifiable since their profile flattens to the right. The dynamics behind the non-identifiable parameters are clearly related, since they all are mechanisms that affect STAT3. The second and fifth best models have very similar parameter profiles, since they differ from the best model only by having one mechanism active in one additional phase. The third and fourth best model have some alternative mechanisms, for instance the synergistic activation of STAT3 by BATF and IRF4. In both models, the rate of this mechanism is non-identifiable like all the other mechanisms affecting STAT3, whereas the remaining parameters are identifiable.

To summarize, most of the parameters included in the best models are identifiable, but the STAT3 production and decay rates are non-identifiable. Since the basal activation and degradation of STAT3 are fixed mechanisms, this problem is likely to be present in all the possible models. This will hinder the optimization performance and reduce the strength of the BIC as a marginal likelihood approximation method. In order to correct this problem, one could try a reparametrization that relates the degradation of STAT3 to its activation rate parameters. One should note that even though the latent process parameters are all identifiable, for example in the best found model the softness parameter λ_2 has its best profile likelihood value outside the range that was allowed for the parameters during the exploration of the model space.

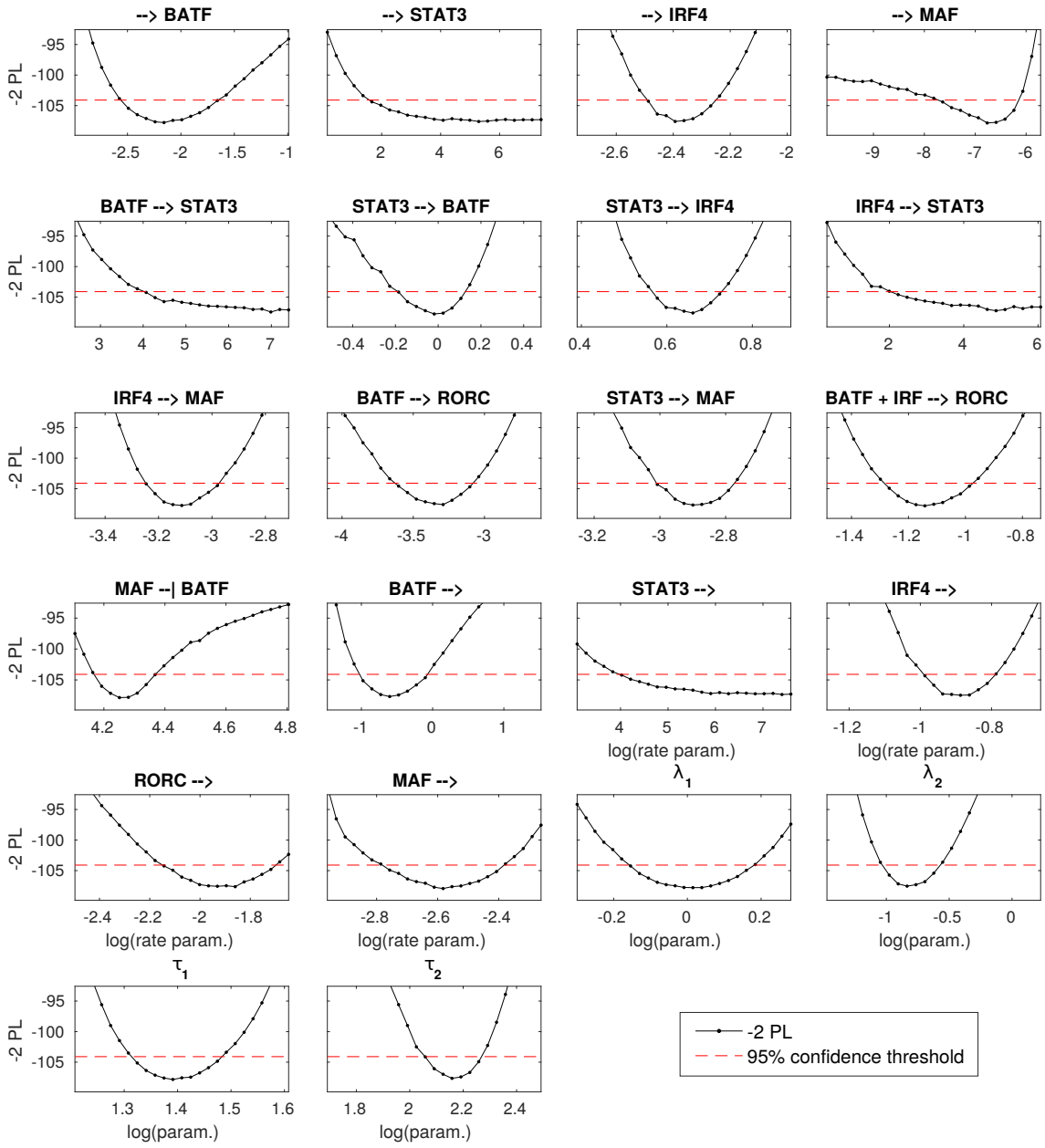


Figure 19: Profile likelihood plots for the parameters of the best found model. The panel titles show the 18 model mechanisms and black lines are the profile likelihood curves for the corresponding reaction rates. Also the four latent process parameters have been included in the analysis. The 95 % confidence interval for a given parameter is the interval where the profile likelihood curve is below the threshold. The parameter profiles indicate that rates of the mechanisms that affect STAT3 are non-identifiable.

7 Discussion and conclusion

This thesis introduces a general framework which can be used to make inferences about the structure of mechanistic ODE models. The approach is especially suitable for inference problems in which the model space is rather large and exhaustive evaluation of all models is out of reach. The strategy that we propose relies on well-established MCMC techniques and provides an efficient means to obtain an approximation of the posterior distribution over alternative model structures. This approximation can then be used to obtain probabilistic predictions about the ODE model structure. The results indicate that algorithm is useful and can be applied to realistically sized problems.

The good performance of our MCMC algorithm is due to the suitable proposal distribution, which allows moving between models that have a similar structure. A sensible proposal distribution enables efficient exploration of the model space as well as fast convergence to the high probability region of the distribution. Since it is possible that the algorithm gets stuck in a local mode, it is of great importance to run several MCMC chains and check if the chains cover the same high probability region(s) or not. Depending on how informative the data are, and how the inference problem is formulated, it can be possible that chains that only partly cover the same high probability regions can also give similar inferences for the ODE model structure. Furthermore, one should note that in discrete space, an MCMC chain needs to visit a single point only once to obtain a rigorous posterior probability for it. Consequently, it is obvious that search strategy provides us always with much richer information when compared with a point estimate that is obtained through a deterministic greedy search. Furthermore, information from several chains is always richer than from a single chain, and using multiple independent chains allows parallel exploration of different parts of the model space. In the approach proposed by this thesis, it is possible to combine the information of independent chains, which generates an opportunity to gain speedup through massive parallelization.

In this thesis, the employed MCMC method was a Metropolis sampler, which has a symmetric proposal distribution and allows rejection of some moves. In different applications, other types of MCMC methods can be constructed based on the desired properties and used proposal distribution. For instance, population-based MCMC

sampling [36] can be used to improve the mixing performance of the sampler or multiple-try Metropolis algorithm [43] can be used to make the sampler explore the space more efficiently. The experiments presented in this thesis assume normally distributed measurement noise, but the noise model can also be altered depending on the application. Furthermore, the prior distribution over the alternative models can be specified by a more informative manner, if such knowledge is available. Also the inference framework can be flexibly implemented in various forms. For example, the BIC approximation for the marginal likelihood can be replaced with a more accurate approximation such as the power posterior estimator [17]. Thus, our framework provides an ODE modeller with a flexible and practical tool which makes it easier to carry out general data-driven mechanistic modeling studies.

The application in which the method is applied in this study is to infer the structure of the gene regulatory network that drives differentation of Th17 cells. The results show that models with three phases in the cellular differentiation are most likely. Alternative models that capture the dynamics of the network are formulated as LEM models [34], which allows us to model a dynamically evolving regulatory network. The latent effect of these models is modeled using sigmoidal functions that are continuous and differentiable during the time course under investigation. The used parameter bounds for the latent process parameters are flexible, could in some applications cause restrictive problems. This could be avoided using an latent process defined using splines, if the phase changes of the differentiation are assumed to be trigged by a discrete event. However, one must take into account the peculiarities of gradient-based parameter optimization for models with such discontinuity [19].

References

[1] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, pages 716–723, Dec. 1974.

[2] B. Alberts, A. Johnson, J. Lewis, D. Morgan, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell*. Garland Science, 6th edition, 2014.

[3] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50:5–43, 2003.

[4] B. Ballnus, S. Hug, K. Hatz, L. Görlitz, J. Hasenauer, and F. J. Theis. Comprehensive benchmarking of markov chain monte carlo methods for dynamical systems. *BMC Systems Biology*, 11(1):63, 2017.

[5] R. Bellman and K. Åström. On structural identifiability. *Mathematical Biosciences*, 7:329–339, 1970.

[6] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society*, 35:99–109, 1943.

[7] G. E. P. Box and G. C. Tiao. *Bayesian Inference in Statistical Analysis*. John Wiley & Sons, 1973.

[8] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, Ltd, 2nd edition, 2008.

[9] G. D. Byrne and A. C. Hindmarsh. A polyalgorithm for the numerical solution of ordinary differential equations. *ACM Transactions on Mathematical Software*, 1(1):71–96, 1975.

[10] B. Calderhead and M. A. Girolami. Estimating Bayes factors via thermodynamic integration and population MCMC. *Computational Statistics and Data Analysis*, 53(12):4028–4045, 2009.

[11] Y. H. Chan, J. Intosalmi, S. Rautio, and H. Lähdesmäki. A subpopulation model to analyze heterogeneous cell differentiation dynamics. *Bioinformatics*, 32(21):3306–3313, 2016.

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

[13] C. F. Curtiss and J. O. Hirschfelder. Integration of stiff equations. *Proceedings of the National Academy of Sciences of the United States of America*, 38(3):235–243, 1952.

[14] G. Dahlquist. Convergence and stability in the numerical integration of ordinary differential equations. *Mathematica Scandinavica*, 4:33–53, 1956.

[15] G. Dahlquist. Stability and error bounds in the numerical integration of ordinary differential equations. *Trans. of the R. Inst. Technol. Stockholm*, 130:87, 1959.

[16] G. Dahlquist. A special stability problem for linear multistep methods. *BIT Numerical Mathematics*, 3(1):27–43, Mar 1963.

[17] N. Friel and A. N. Pettitt. Marginal likelihood estimation via power posteriors. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(3):589–607, 2008.

[18] N. Friel and J. Wyse. Estimating the evidence – a review. *Statistica Neerlandica*, 66(3):288–308, 2012.

[19] F. Fröhlich, F. J. Theis, J. O. Rädler, and J. Hasenauer. Parameter estimation for dynamical systems with discrete events and logical operations. *Bioinformatics*, 33(7):1049–1056, 2017.

[20] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 3rd edition, 2013.

[21] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(6):721–741, 1984.

[22] C. J. Geyer. Introduction to Markov chain Monte Carlo. In S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, editors, *Handbook of Markov Chain Monte Carlo*, chapter 1. Chapman & Hall/CRC, 2011.

[23] D. T. Gillespie. A rigorous derivation of the chemical master equation. *Physica A: Statistical Mechanics and its Applications*, 188(1-3):404–425, 1992.

[24] D. T. Gillespie. Stochastic simulation of chemical kinetics. *Annual Review of Physical Chemistry*, 58(1):35–55, 2007.

[25] M. A. Girolami. Bayesian inference for differential equations. *Theoretical Computer Science*, 408(1):4–16, 2008.

[26] D. F. Griffiths and D. J. Higham. *Numerical Methods for Ordinary Differential Equations: Initial Value Problems*. London: Springer, 1st edition, 2010.

[27] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer-Verlag, Berlin, 2nd edition, 1996.

[28] L. E. Harrington, R. D. Hatton, P. R. Mangan, H. Turner, T. L. Murphy, K. M. Murphy, and C. T. Weaver. Interleukin 17-producing CD4+ effector T cells develop via a lineage distinct from the T helper type 1 and 2 lineages. *Nature Immunology*, 6(11):1123–1132, 2005.

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

[30] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software*, 31(3):363–396, 2005.

[31] B. P. Ingalls. *Mathematical Modeling in Systems Biology: An Introduction*. MIT Press, 2013.

[32] H. F. Inman and E. L. Bradley Jr. The overlapping coefficient as a measure of agreement between probability distributions and point estimation of the overlap of two normal densities. *Comm. Stat. Theor. Meth.*, 18(10):3851–3874, 1989.

[33] J. Intosalmi, H. Ahlfors, S. Rautio, H. Mannerström, Z. Chen, R. Lahesmaa, B. Stockinger, and H. Lähdesmäki. Analyzing Th17 cell differentiation dynamics using a novel integrative modeling framework for time-course RNA sequencing data. *BMC Systems Biology*, 9(81), 2015.

[34] J. Intosalmi, K. Nousiainen, H. Ahlfors, and H. Lähdesmäki. Data-driven mechanistic analysis method to reveal dynamically evolving regulatory networks. *Bioinformatics*, 32(12):288–296, 2016.

[35] K. R. Jackson and R. Sacks-Davis. An alternative implementation of variable step-size multistep formulas for stiff ODEs. *ACM Transactions on Mathematical Software*, 6(3):295–318, 1980.

[36] A. Jasra, D. A. Stephens, and C. C. Holmes. On population-based simulation for static inference. *Statistics and Computing*, 17(3):263–279, 2007.

[37] H. Jeffreys. *The Theory of Probability*. Oxford, 3rd edition, 1961.

[38] E. Klipp, W. Liebermeister, C. Wierling, A. Kowald, H. Lehrach, and R. Herwig. *Systems Biology: A Textbook*. Wiley-VCH, 1st edition, 2009.

[39] T. Korn, E. Bettelli, M. Oukka, and V. K. Kuchroo. IL-17 and Th17 Cells. *Annual Review of Immunology*, 27(1):485–517, 2009.

[40] C. Kreutz, A. Raue, D. Kaschek, and J. Timmer. Profile likelihood in systems biology. *FEBS Journal*, 280(11):2564–2571, 2013.

[41] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[42] J. R. Leis and M. A. Kramer. The simultaneous solution and sensitivity analysis of systems described by ordinary differential equations. *ACM Transactions on Mathematical Software*, 14(1), 1988.

[43] J. S. Liu, F. Liang, and W. H. Wong. The multiple-try method and local optimization in Metropolis sampling. *Journal of the American Statistical Association*, 95(449):121–134, 2000.

[44] C. Maier, C. Loos, and J. Hasenauer. Robust parameter estimation for dynamical systems from outlier-corrupted data. *Bioinformatics*, 33:718—-725, 2017.

[45] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[46] C. G. Moles, P. Mendes, and J. R. Banga. Parameter estimation in biochemical pathways: A comparison of global optimization methods. 13:2467–2474, 2003.

[47] H. Park, Z. Li, X. O. Yang, S. H. Chang, R. Nurieva, Y.-H. Wang, Y. Wang, L. Hood, Z. Zhu, Q. Tian, and C. Dong. A distinct lineage of CD4 T cells regulates tissue inflammation by producing interleukin 17. *Nature Immunology*, 6(11):1133–1141, 2005.

[48] A. Raue, J. Karlsson, M. P. Saccomani, M. Jirstrand, and J. Timmer. Comparison of approaches for parameter identifiability analysis of biological systems. *Bioinformatics*, 30(10):1440–1448, 2014.

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

[50] A. Raue, M. Schilling, J. Bachmann, A. Matteson, M. Schelker, D. Kaschek, S. Hug, C. Kreutz, B. D. Harms, F. J. Theis, U. Klingmüller, and J. Timmer. Lessons learned from quantitative dynamical modeling in systems biology. *PLOS ONE*, 8(9):1–17, 2013.

[51] A. Raue, B. Steiert, M. Schelker, C. Kreutz, T. Maiwald, H. Hass, J. Vanlier, C. Tönsing, L. Adlung, R. Engesser, W. Mader, T. Heinemann, J. Hasenauer, M. Schilling, T. Höfer, E. Klipp, F. Theis, U. Klingmüller, B. Schöberl, and J. Timmer. Data2Dynamics: a modeling environment tailored to parameter estimation in dynamical systems. *Bioinformatics*, 31(21):3558–3560, 2015.

[52] B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1st edition, 1996.

[53] C. P. Robert. *The Bayesian Choice*. Springer texts in statistics. Springer, 2007.

[54] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer texts in statistics. Springer, 2nd edition, 2004.

[55] H. H. Robertson. The solution of a set of reaction rate equations. *Academ Press*, pages 178–182, 1966.

[56] J. S. Rosenthal. Optimal proposal distributions and adaptive MCMC. In S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, editors, *Handbook of Markov Chain Monte Carlo*, chapter 4. Chapman & Hall/CRC, 2011.

[57] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.

[58] E. Süli and D. F. Mayers. *An Introduction to Numerical Analysis*. Cambridge University Press, 1st edition, 2003.

[59] L. A. Tesmer, S. K. Lundy, S. Sarkar, and D. A. Fox. Th17 cells in human disease. *Immunology Review*, 223:87–113, 2008.

[60] The MathWorks, Inc. *Optimization Toolbox ™ User's Guide, Matlab R2017a*. 2017.

[61] V. Vyshemirsky and M. A. Girolami. Bayesian ranking of biochemical system models. *Bioinformatics*, 24(6):833–839, 2008.

[62] C. T. Weaver, C. O. Elson, L. A. Fouser, and J. K. Kolls. The th17 pathway and inflammatory diseases of the intestines, lungs and skin. *Annual review of pathology*, 8:477–512, 2013.

[63] M. S. Weitzman. Measure of the overlap of income distribution of white and negro families in the United States. Technical Report 22, U.S. Department of Communications, Bureaus of the Census, Washington DC, 1970.

[64] T.-R. Xu, V. Vyshemirsky, A. Gormand, A. von Kriegsheim, M. Girolami, G. S. Baillie, D. Ketley, A. J. Dunlop, G. Milligan, M. D. Houslay, and W. Kolch. Inferring signaling pathway topologies from multiple perturbation measurements of specific biochemical species. *Science Signaling*, 3(113):ra20, 2010.

[65] N. Yosef, A. K. Shalek, J. T. Gaublomme, H. Jin, Y. Lee, A. Awasthi, C. Wu, K. Karwacz, S. Xiao, M. Jorgolli, D. Gennert, R. Satija, A. Shakya, D. Y. Lu, J. J. Trombetta, M. R. Pillai, P. J. Ratcliffe, M. L. Coleman, M. Bix, D. Tantin, H. Park, V. K. Kuchroo, and A. Regev. Dynamic regulatory network controlling TH17 cell differentiation. *Nature*, 496(7446):461–468, 2013.

[66] J. Zhu, H. Yamane, and W. E. Paul. Differentiation of effector CD4 T cell populations (*). *Annual review of immunology*, 28:445–89, 2010.