Parameter estimation for biochemical reaction networks using Wasserstein distances

Kaan Öcal

Master of Science

School of Informatics
University of Edinburgh

2019

Abstract

Mathematical modelling of biological processes is an essential aspect of contemporary research in fields as diverse as molecular biology, population biology and ecology. Stochastic models in particular are able to accurately describe many biological phenomena such as gene expression, cell fate decision and the spread of diseases in a population, which are known to be inherently non-deterministic. Despite their increased accuracy the use of stochastic models in practice is hampered by their complexity, with efficient parameter inference for such models currently being a particularly active topic of research. In this thesis we present a method for estimating parameters for a general class of stochastic reaction networks based on experimentally obtained population snapshot data. We obtain parameter estimates by minimizing a Wasserstein distance between the observed distribution over states and the distribution predicted by the model. Since the evaluation of this distance requires expensive simulations we use a Gaussian process to learn the distance for all parameters and apply Bayesian optimization to efficiently minimize it. The effectiveness of our method is demonstrated on three biologically relevant examples from the literature: a classical gene expression network, a genetic feedback loop and a spatial dimerization system. Our approach only requires access to a simulator and can be used where analytical descriptions of the model are unavailable.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Kaan Öcal*)

Acknowledgements

I would like to thank Steven Kleinegesse and Michalis Michaelides for their valuable advice on the proper application and implementation of Bayesian optimization, and Patric Fulop for sharing his insights on computational optimal transport which have been essential for this work. I would especially like to thank my supervisors, Guido Sanguinetti and Ramon Grima, for their support and guidance through every stage of this project.

This work was supported in part by the EPSRC Centre for Doctoral Training in Data Science, funded by the UK Engineering and Physical Sciences Research Council (grant EP/L016427/1) and the University of Edinburgh, and the German National Academic Foundation.

Table of Contents

Chapter 1

Introduction

Quantitative modelling plays an increasingly prominent role in modern biological research. Of particular interest is the study of biological reaction networks [20] at the molecular level, including those involved in gene expression, cellular signalling and metabolism. These processes are among the essential building blocks of organic life and a detailed understanding of them is therefore necessary to further advance our knowledge about biological systems in general. Various mathematical models such as deterministic rate equations, reaction-diffusion equations, the Chemical Master Equation [20] and Brownian dynamics [46] have been developed in order to capture important properties of these networks and gain insight into their function, and their study comprises a fundamental part of contemporary systems biology.

A prerequisite to applying mathematical models in the sciences is fitting parameters to data, whereby empirical observations are used to calibrate the models in order to obtain physical insight and predictive power. In biology such parameters can be e.g. the rate of a chemical reaction, the diffusion constant of a signal protein or the affinity of a receptor to a ligand. It is rarely possible to measure these quantities directly, necessitating the use of sophisticated statistical inference methods to fit models based on quantities that can be observed. The complexity and sensitivity of biological systems, coupled with imperfect experimental methods, often place hard limits on what can be recorded in an experiment, frequently requiring the use of specialized mathematical tools for parameter estimation depending on the model and the type of data available.

Our contribution in this thesis is a method for estimating parameters in stochastic models of biochemical reaction networks based on population snapshot data where concentrations of chemical species are measured at a fixed time across a population.

Stochastic models of reaction systems describe a system in terms of probability distributions over particle numbers instead of deterministic quantities, and it is possible to estimate parameters in such models by matching the probability distribution predicted by the model to the experimentally observed distribution in a population. Thanks to flow cytometry [1] and related experimental methods it is possible to measure cell-by-cell molecule numbers for RNA, proteins and other substances for large numbers of cells at a time, yielding accurate estimates of such probability distributions which can be used for fitting and evaluating these models.

In order to compare and match the probability distributions we propose to use Wasserstein distances, introducing the Wasserstein loss as a quantity to be minimized for parameter estimation. Wasserstein distances are a flexible class of distance metrics for general probability distributions and have recently gained interest in the Machine Learning community [11, 26, 32], notably in the wake of Wasserstein GANs [2]. In the context of this work they provide a principled method for assessing the discrepancy between the probability distributions returned by our models, rendering it suitable not only for parameter estimation but also for tasks such as model comparison, which will be briefly discussed in this work.

In this thesis we will mainly rely on the well-known Chemical Master Equation [20] to describe stochastic reaction networks, but the methods we develop are applicable to other simulator-based models such as the Reaction-Diffusion Master Equation and Brownian dynamics [46] which are of interest to the scientific community. The probability distributions described by these models can occasionally be computed directly, but for most non-trivial examples such computations are infeasible and one has to obtain Monte Carlo approximations using direct simulations. Such sampling-based approaches typically result in black-box models, that is, they do not give the functional dependence of the output distributions on the input parameters, which makes it difficult to perform parameter inference or sensitivity analysis without the use of specialized mathematical methods. In order to combat this difficulty we propose to use Bayesian optimization [43], a state-of-the-art method for black-box function optimization in low to moderate numbers of dimensions. With this approach we obtain an effective method for parameter estimation requiring a modest number of simulations compared to classical approaches and that is suited for simulator models such as Brownian dynamics for which efficient inference methods are currently unavailable [9].

This thesis is organized as follows. Chapter 2 will provide the background necessary to describe biochemical reaction networks and the Chemical Master Equation,

as well as Wasserstein distances and Bayesian optimization. In Chapter 3 we will describe our goals for this project and review currently available methods for parameter inference in stochastic biochemical reaction networks. Chapters 4 and 5 will describe our framework from a theoretical and an implementation point of view. Chapter 6 will present an analysis of the results obtained using our method on several models from the literature, and finally Chapter 7 will evaluate the results, discuss limitations of our approach and give suggestions and outlines for future work.

Except for Sections 3.2 and 6.4 and the appendix most of the contents of this thesis have been adapted from [31], with a majority of the sections of the paper being included verbatim or with slight modifications.

Chapter 2

Background

2.1 Biochemical Reaction Networks

A biochemical reaction network consists of species S_j $(j = 1, \ldots, s)$ and reactions R_i $(i = 1, \ldots, r)$ of the form

$$a_{i,1}S_1 + \ldots + a_{i,s}S_s \longrightarrow b_{i,1}S_1 + \ldots + b_{i,s}S_s \tag{2.1}$$

where $\boldsymbol{a}_i := (a_{i,1}, \ldots, a_{i,s})$ and $\boldsymbol{b}_i := (b_{i,1}, \ldots, b_{i,s})$ are vectors of nonnegative integers, the stoichiometric coefficients of the reactions. The species can describe molecules of interest such as proteins, genes, enzymes or metabolites and the reactions represent interactions between these, e.g. the binding of a protein to a gene or the conversion of a metabolite by an enzyme. Reactions are classified as zero-molecular, unimolecular, bimolecular etc. depending on the number of educt particles involved, which also defines the order of the reaction (the number of product particles is not relevant for the classification).

It will be illustrative to consider the example of a simple gene transcription network depicted in Fig. 2.1, consisting of a gene G, its associated mRNA M and the protein P which is coded for by the gene. According to the central dogma of molecular biology the gene is transcribed to produce mRNA molecules which are then in turn used as templates for the protein during transcription. Both the mRNA and the protein itself are subject to degradation with a limited half life; one thus obtains the four reactions listed in Fig. 2.1.

Reaction networks as defined above can be used to describe a variety of biochemical processes of interest such as gene expression and regulation, cellular signalling and protein polymerization. They can also be used for describing population models in

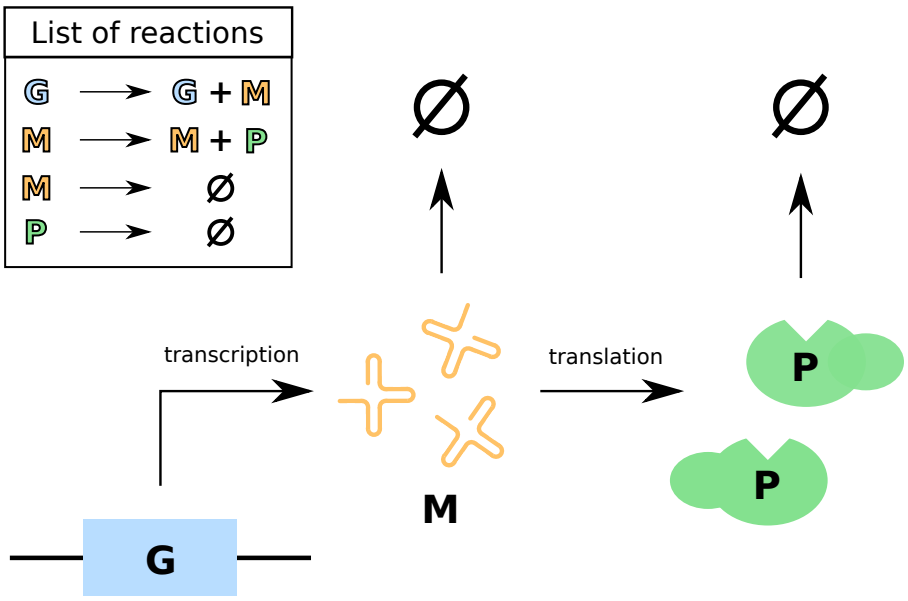


Figure 2.1: Illustration of the two-stage gene expression model. The gene (**G**) consists of a string of DNA that is transcribed into messenger RNA (**M**), which in turn is used to produce proteins (**P**). Both the mRNA and the protein get degraded after a sufficiently long time.

ecology and epidemiology, including the well-known Lotka-Volterra and SIR models. While our methods are also applicable to these models we will exclusively focus on biomolecular reaction networks in this work.

At this point we have only been concerned with an abstract description of a reaction network which is incomplete in that it only describes its qualitative aspects, namely which reactants and reactions are involved. Equally important in the study of a reaction network is to define its dynamics and the mechanisms according to which the above reactions take place. Common approaches to studying such reaction networks assume that the reactant numbers or concentrations evolve according to a set of deterministic ODEs, but it has by now been well established that many reaction networks in biology are inherently stochastic [5, 8, 21] and that deterministic approaches can fail to capture properties essential for our understanding of these systems [25, 34]. We will therefore turn our attention to the Chemical Master Equation, a commonly used model for stochastic reaction networks which is able to deal with many of the phenomena caused by the inherent stochasticity of such systems in nature.

2.1.1 The Chemical Master Equation

This section aims to provide a brief review of the Chemical Master Equation approach to modelling stochastic reaction networks in biology, referring to [20, 40] for a readable and comprehensive treatment of the theory.

We model a given reaction network as a Markov chain whose states are given by vectors $\boldsymbol{n} := (n_1, \ldots, n_s) \in \mathbb{N}^s$ defining the number of particles of each species present at a given time. The positions of the particles are not taken into consideration, and it is assumed that the particles are uniformly distributed in the reaction volume at all times. The transitions of the Markov chain correspond to the possible reactions changing the state vector, with the rate of reaction R_i determined by the state-dependent propensity function $\rho_i(\boldsymbol{n})$. The forward Kolmogorov equation for this Markov chain is called the Chemical Master Equation and reads:

$$\frac{\partial}{\partial t} P(\boldsymbol{n}, t) = \sum_{i=1}^{r} \left[\rho_i (\boldsymbol{n} - \boldsymbol{S}_i) P (\boldsymbol{n} - \boldsymbol{S}_i, t) - \rho_i(\boldsymbol{n}) P(\boldsymbol{n}, t) \right] \tag{2.2}$$

Here $\boldsymbol{S}_i := \boldsymbol{b}_i - \boldsymbol{a}_i$ describes the net change in reactant numbers during an occurrence of reaction R_i. The Chemical Master Equation defines the evolution of the probability distribution over states (i.e. particle counts) in time.

The form of the transition functions $\rho_j(\vec{n})$ depends on the specific reaction, but the most commonly used transition functions are given by the mass-action law,

$$\rho_i(\boldsymbol{n}) := \lambda_i \binom{n_1}{a_{i,1}} \ldots \binom{n_s}{a_{i,s}} \tag{2.3}$$

for certain rate constants $\lambda_i > 0$. While our approach can handle general transition functions, in what follows we restrict ourselves to mass-action propensities of the form (2.3); with this setup the task of inferring parameters for the CME reduces to finding the appropriate rate constants λ_i.

An extension of the Chemical Master Equation incorporating spatial information, the so-called Reaction-Diffusion Master Equation (RDME), is described in the appendix. The RDME keeps track of the locations of reactants in a grid and is useful in cases where spatial aspects such as diffusion play a role, e.g. in the study of cell signalling networks. We will use the RDME instead of the CME in Section 6.4 where we apply the framework developed in this thesis to analyse properties of the RDME. A rather different class of models used to simulate stochastic reaction networks in a spatial setting, Brownian dynamics (BD) simulations, will also be used in that section (see the appendix for a description).

2.1.2 Population Snapshot Data

The main type of data one works with in the context of biological reaction systems is cell-by-cell particle counts. There are various methods for obtaining these particle counts, two important ones being fluorescence microscopy and flow cytometry. Fluorescence microscopy allows scientists to "tag" molecules found in cells using fluorescent signal molecules and to observe the number and locations of the tagged molecules by measuring the intensity of the light emitted by the latter. Using specialized signal molecules binding to different targets it is often possible to measure the presence of up to two to three types of molecules simultaneously, yielding empirical joint distributions over particle numbers. Flow cytometry works by directing a large population of cells through an optical measuring device which can automatically measure abundances of selected molecules such as RNA or proteins for large numbers of cells at a time, although it is usually only possible to measure the abundance of one chemical species at a time.

In stochastic dynamical systems such as the reaction networks considered in this thesis, measurements will always be subject to non-deterministic fluctuations. This implies that particle counts cannot be reliably observed, but *distributions* over particle counts, such as those computed by the CME, can. These probability distributions will be the data on which our inference method is based.

2.2 Wasserstein distances

We perform parameter estimation based on population snapshot data by minimizing the discrepancy between the observed distribution over particle numbers and the distribution returned by the simulator. In this section we motivate and describe our choice of discrepancy measure, namely Wasserstein distances, and refer to [33, 53] for a more detailed overview of the discussed topics.

Stochastic models for biochemical reaction networks usually describe a system in terms of probability distributions over particle counts, i.e. probability distributions on $\mathbb{N}^s$, where s is the number of species in the reaction network considered. These probability distributions can sometimes be computed explicitly, e.g. by solving the Chemical Master Equation, but in most cases they need to be approximated empirically using Monte Carlo methods. Due to the finite number of samples used these empirical histograms will generally be finitely supported and subject to fluctuations, which will

limit the choice of discrepancy metrics suitable for inference as we shall see shortly.

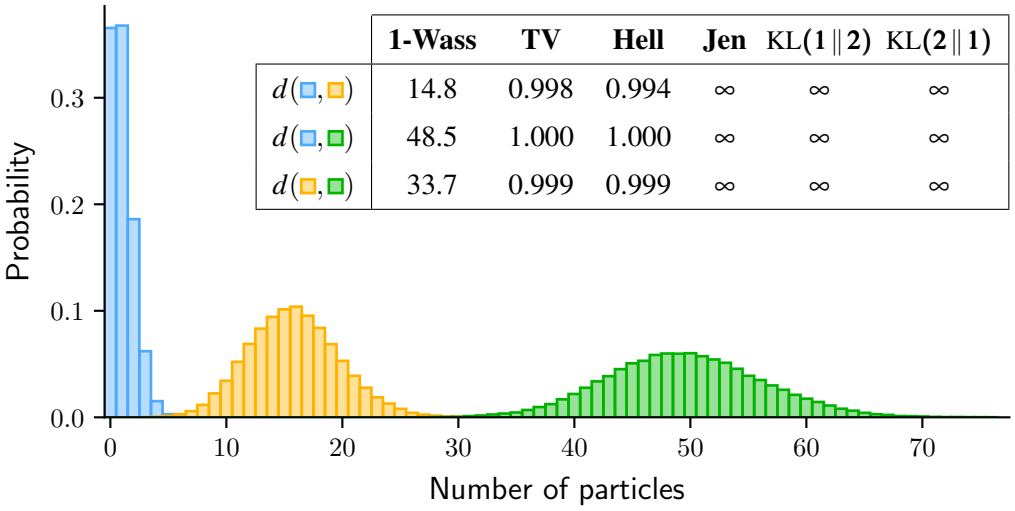
There exists a variety of methods for comparing general probability distributions which we can roughly divide into summary statistic-based and direct methods:

Summary statistic-based methods usually consider finite-dimensional statistics such as a set of moments of the given probability distributions and try to compare the distributions by measuring the differences in these summary statistics.

Direct methods consider discrepancy measures on the level of distributions such as the Kullback-Leibler divergence. Such discrepancy measures play an important role in statistics and machine learning and enjoy a rich theoretical foundation. The Jensen divergence, the Total Variation (TV) distance and the Hellinger distance are classical examples of such discrepancy measures, while we will focus on the less common class of Wasserstein distances [33, 53] in this work.

Summary statistic-based methods tend to be simple to implement and computationally efficient, especially for parameterized distributions, but in the context of this project they suffers from issues that can potentially reduce their effectiveness. There is a fundamental ambiguity in describing a general probability distribution by finitely many summary statistics; given any finite set of prescribed moments there is usually a large number of probability distributions with the the same value for these moments (cf. the so-called Moment Problem). In practice only a small subset of these probability distributions will be candidate distributions, namely these that can be produced by the model, and taking a sufficiently large set of moments will usually identify the distribution uniquely. Unless one is dealing with parametric families of distributions however it is not *a priori* clear which set of moments will suffice for this, and it is generally difficult to determine if two probability distributions with matching moments agree as a whole without performing direct (non-summary statistic-based) comparisons. In cases where e.g. certain parameters in the model cannot be uniquely identified this can lead to problems since it will not be immediately obvious if this phenomenon is due to using an insufficient set of summary statistics or due to the model parameters being inherently underconstrained. Furthermore many classes of distributions of particular interest in systems biology exhibit features such as multimodality which makes it difficult to characterize them by their moments.

Direct methods can bypass the above difficulties but their use is complicated by different problems. With the exception of Wasserstein distances all mentioned discrepancy measures are purely overlap-based and do not take into account the geom-



	1-Wass	TV	Hell	Jen	KL(1 ‖ 2)	KL(2 ‖ 1)
$d(\square,\square)$	14.8	0.998	0.994	∞	∞	∞
$d(\square,\square)$	48.5	1.000	1.000	∞	∞	∞
$d(\square,\square)$	33.7	0.999	0.999	∞	∞	∞

Figure 2.2: Common discrepancy measures for probability distributions do not provide usable distance metrics for simulated data. The histograms show empirical estimates of the steady-state distributions for a simple birth-death process with three different ratios r of birth and death rates ($r_{\square} = 1$, $r_{\square} = 16$, $r_{\square} = 50$). The table compares their 1-Wasserstein (1-Wass), total variation (TV) and Hellinger (Hell) distances, their Jensen divergences (Jen) and their Kullback-Leibler divergences (KL). Even though the two outer histograms are significantly further apart than the neighbouring pairs, the total variation and Hellinger distances in all cases differ by less than 1%, and the Kullback-Leibler and Jensen divergences between any two of these histograms are infinite. The 1-Wasserstein distance on the other hand captures an intuitive notion of distance between these histograms.

etry of the underlying spaces, leading to counterintuitive behaviour in some cases (cf. Fig. 2.2). They are effective at comparing parametrized families of distributions with infinite support but often run into difficulties when the two compared distributions live in different regions. Our models will output empirically estimated histograms in which such a scenario will be quite common, drastically limiting the usefulness of these discrepancy measures. For this reason we have focused on Wasserstein distances in this project; the remainder of this section will be concerned with defining these distances.

Consider two normalized histograms P, Q over $\mathbb{N}^s$ representing probability distributions; the value of the histogram P at $\vec{i} = (i_1, \ldots, i_s) \in \mathbb{N}^s$ is denoted $P_{\vec{i}}$. A transport plan T between P and Q is a histogram on $\mathbb{N}^s \times \mathbb{N}^s$ whose first and second marginals

are P and Q, respectively,

$$\sum_{\vec{j}} T_{\vec{i},\vec{j}} = P_{\vec{i}} \qquad \sum_{\vec{i}} T_{\vec{i},\vec{j}} = Q_{\vec{j}} \tag{2.4}$$

The value $T_{\vec{i},\vec{j}}$ can be viewed as the amount of probability mass that has to be moved from $\vec{i}$ to $\vec{j}$ in order to convert the histogram P into Q; Eq. (2.4) then represents the conservation of probability mass during this process. The simplest transport plan between P and Q is the independent coupling given by

$$(P \otimes Q)_{\vec{i},\vec{j}} = P_{\vec{i}} \cdot Q_{\vec{j}} \tag{2.5}$$

which specifies that the probability mass in every bin of P is to be distributed evenly across Q. We denote the space of transport plans between P and Q by $U(P,Q)$.

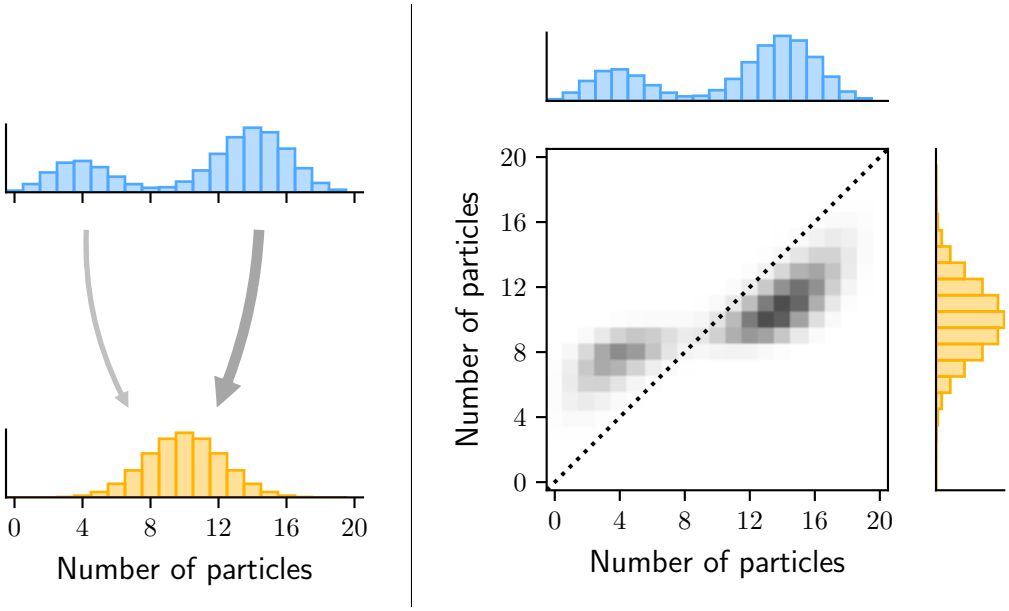


Figure 2.3: **(a)** Optimal transport distances between histograms measure how much mass has to be moved in order to convert one histogram into the other. **(b)** Illustration of a transport plan between the two histograms in (a). The joint histogram shows the amount of mass transported between different locations in the histograms. Mass on the diagonal (dotted line) is not moved during transport.

Optimal transport maps are defined by assigning to each move a certain cost. We define a cost function C to be a nonnegative function on $\mathbb{N}^s \times \mathbb{N}^s$, where $C_{\vec{i},\vec{j}}$ represents the cost involved in transporting a unit of probability mass from $\vec{i}$ to $\vec{j}$. While this is

not necessary for the theory we will assume that the cost function is a distance metric on the ground space $\mathbb{N}^s$. The optimal transport problem with cost function C reads

$$\mathcal{W}_C(P,Q) := \inf_{T \in U(P,Q)} \langle C, T \rangle = \inf \left\{ \sum_{\vec{i},\vec{j}} C_{\vec{i},\vec{j}} T_{\vec{i},\vec{j}} \colon T_{\vec{i},\vec{j}} \geq 0, \right. \\ \left. \sum_{\vec{j}} T_{\vec{i},\vec{j}} = P_{\vec{i}}, \sum_{\vec{i}} T_{\vec{i},\vec{j}} = Q_{\vec{j}} \right\} \tag{2.6}$$

One can check that $\mathcal{W}_C$ defines a metric on the space of probability distributions on $\mathbb{N}^s$, called the (1-)Wasserstein distance with cost function C. More generally one can verify that the p-Wasserstein distance

$$\mathcal{W}_{C,p}(P,Q) := \mathcal{W}_{C^p}(P,Q)^{1/p} \tag{2.7}$$

defines a metric on the space of probability distributions on $\mathbb{N}^s$ for all $p \geq 1$.

Wasserstein distances are always finite and well-defined as long as the appropriate moments of the distributions involved exist, which is always the case for finitely supported histograms. Unlike the other direct discrepancy measures mentioned above, Wasserstein distances do not solely rely on overlap and are robust to the small fluctuations typically involved in empirically approximating probability distributions. For this reason we will use Wasserstein distances for a suitable ground metric C as our discrepancy measure to compare probability distributions.

2.3 Gaussian process Bayesian optimization

The dependence of the Wasserstein distance between the experimentally observed distribution and the distribution returned by the chosen model on the parameters of the model is generally not available in closed form and can only be evaluated by running expensive simulations. We are thus faced with the task of minimizing a function that is noisy (due to finite sample sizes), costly to evaluate and about which no additional information such as knowledge of gradients is given. In order to do this efficiently we rely on Bayesian optimization, a method for efficiently optimizing expensive black-box functions in low to moderate dimensions based on a Gaussian process surrogate of the target function. See [35] for a comprehensive reference on Gaussian processes and [43] for an overview of Bayesian optimization going beyond the description in this paper.

In order to apply Bayesian optimization to our problem we start by placing a Gaussian process prior on the loss function $L(\mathbf{x})$ that measures the discrepancy with the observed data:

$$\hat{L} \sim \mathrm{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{2.8}$$

with mean function $\mu(\mathbf{x})$ and covariance kernel $k(\mathbf{x}, \mathbf{x}')$. Thus $\hat{L}$ is a statistical model of the true function L. We assume that we can use a simulator to compute noisy observations of $L(\mathbf{x}_i)$:

$$\tilde{L}(\mathbf{x}_i) = L(\mathbf{x}_i) + \varepsilon_i \tag{2.9}$$

at any given point $\boldsymbol{x}_i$, where the ε_i are measurement noise. We assume that the ε_i are iid. normal random variables with mean zero. With this setup our Gaussian process $\hat{L}$ can be updated by obtaining data points $\mathcal{D}_i = \{\mathbf{x}_i, \tilde{L}(\mathbf{x}_i)\}$ for different $\mathbf{x}_i$ and computing the posterior $\hat{L} \mid \mathcal{D}$.

Our goal is to minimize $L(\mathbf{x})$ with as few evaluations of $\tilde{L}(\mathbf{x})$ as possible. Bayesian optimization consists of a procedure for sequentially choosing the points $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$ at which $\tilde{L}(\mathbf{x})$ is to be evaluated in order to decrease the uncertainty about the location of the optimum, based on the Gaussian process $\hat{L}$. This is done by considering a so-called acquisition function $\alpha(\mathbf{x}; L \mid \mathcal{D})$ depending on the collected observations $\mathcal{D}$ and choosing the next point to evaluate as

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x} \in \mathcal{X}} \alpha(\mathbf{x}; L \mid \mathcal{D}_{1:n}) \tag{2.10}$$

The acquisition function returns a point $\mathbf{x}_{n+1}$ such that computing $\tilde{L}(\mathbf{x}_{n+1})$ yields additional knowledge about the minimum of $L(\mathbf{x})$, e.g. by choosing a point which is likely to be near the true minimum. It should be simpler to evaluate and optimize than the target function so that one can use standard optimization methods for finding $\mathbf{x}_{n+1}$ with little overhead. After finding $\mathbf{x}_{n+1}$ and running the simulator to compute $\tilde{L}(\mathbf{x}_{n+1})$ one updates the Gaussian process $\hat{L}$ with the data $\mathcal{D}_{n+1} = \{\mathbf{x}_{n+1}, \tilde{L}(\mathbf{x}_{n+1})\}$ and repeats this procedure until the true optimum of $L(\mathbf{x})$ is found. An illustration of Bayesian optimization can be seen in Fig. 2.4.

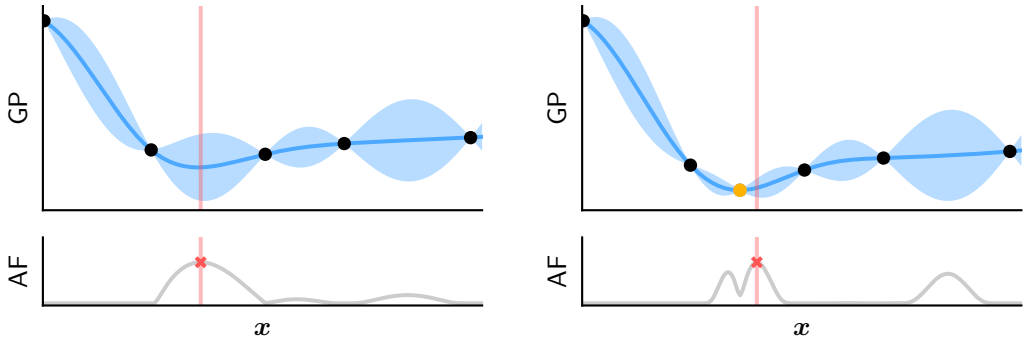
A common choice for the acquisition function α is Expected Improvement, defined by the following formula:

$$\alpha_{\mathrm{EI}}(\mathbf{x}; L \mid \mathcal{D}) := \mathbb{E}_{L|\mathcal{D}} \left[\left(\min_{\mathbf{x}_i \in \mathcal{D}} \tilde{L}(\mathbf{x}_i) - \hat{L}(\mathbf{x}) - \beta \right)^+ \right] \tag{2.11}$$

Here $\beta \geq 0$ is a small "jitter" parameter used to reduce the time spent in local optima and increase exploration. With this acquisition function the predicted optimum of $L(\mathbf{x})$ is typically computed as:

$$\mathbf{x}^* := \arg\min_{\mathbf{x}_i \in \mathcal{D}} \tilde{L}(\mathbf{x}_i) \tag{2.12}$$

Since Eq. (2.11) can be computed in closed form the expected improvement at a point $\mathbf{x}$ can be evaluated quite cheaply, and gradients can be computed at little additional cost. It is known that Bayesian optimization using this acquisition function is guaranteed to find the optimum of the target function L under some mild assumptions on L and the Gaussian process prior [52][1]. This combined with its simplicity and empirical performance properties make Expected Improvement a popular choice of acquisition function in Bayesian optimization. Other common acquisition functions are Upper Confidence Bound, Probability of Improvement and Knowledge Gradient [43], which we shall not consider here.



(a) Before update **(b)** After update

Figure 2.4: Illustration of Bayesian optimization in one dimension. Plotted are a Gaussian process (GP) and its acquisition function (AF) before and after an update step. The shaded area represents two standard deviations around the mean. Each round consists of computing the loss function at the point maximizing the acquisition function (vertical line) and updating the GP with the computed value. After the update step the acquisition function changes to reflect the information gained in the process, and a new point is chosen for the next round.

[1] We point out that our Gaussian process hyperparameters will not be fixed during optimization and that the cited theoretical guarantees do not necessarily apply in our case.

Chapter 3

Objectives

3.1 Prior work

Bayesian optimization has been applied to several chemical and biological problems in recent years. The paper [50] considers Bayesian optimization for parameter inference in models of molecular assembly; the authors do not consider spatial models and fit data by minimizing the total discrepancy between particle concentrations at different time points using a multi-output Gaussian Process as a surrogate. In [16] the authors optimize codon choice in the design of synthetic genes, and [51] presents a method for efficient Bayesian optimization for deterministic models of biochemical reaction networks based on solving deterministic rate equations and minimizing the difference between the computed solution and observed values at different time points. This is in contrast to our application of Bayesian optimization to stochastic models, which will require constructing a suitable loss function to measure the discrepancy between probability distributions (see Section 4.2).

Parameter inference for stochastic biochemical reaction networks typically focuses on the Chemical Master Equation as it is quite well-understood from a theoretical point of view. Since the CME usually describes an infinite system of coupled ODEs, solving it numerically is not generally possible and in practice one often relies on various approximations that have been developed to this day [20, 40]. The diffusion approximation and the system size expansion [20] are two well-known examples and frequently used for inference, especially in the Bayesian setting due to their mathematical tractability [14, 15, 22]. However, in the presence of bimolecular reactions such as enzyme-substrate or protein-DNA interactions these approximations are unable to deal with the low copy numbers frequently found in biological systems, rendering

them unsuitable for modelling or inference in reaction systems involving species such as e.g. individual genes or mRNA, which is often present in copy numbers of less than 20 per cell [23].

Other methods for parameter inference rely on fitting moments of the particle number distributions returned by the CME to experimental data [6, 12, 36, 54]. Moments can often be computed or approximated by solving a set of coupled ODEs which can be derived from the CME [20, 40], bypassing expensive simulations of the system in question. For reaction systems with only zeroth or first-order reactions the moments can be computed explicitly at all times, and such computations can be carried out very efficiently on a computer. For reaction systems with higher-order reactions, however, the equations for the moments of a given order involve moments of strictly higher orders and the resulting infinite moment hierarchy cannot be solved in general. Moment closure approximations, where selected higher-order moments are expressed as functions of lower-order moments in order to truncate the hierarchy, are a popular method for estimating moments in this case, but they modify the original moment equations and are therefore not guaranteed to yield accurate results; validity conditions for general moment closure schemes are currently not well-understood [37–39]. Given the wide variety of moment closure schemes it is not generally clear *a priori* which, if any, will prove suitable for a given reaction system, and the right method is usually chosen empirically based on performance [4]. In addition, moment closure typically results in a set of coupled nonlinear equations which can have multiple different solutions, further complicating their use in parameter inference. We will provide an example of a genetic feedback loop based on [4] for which many commonly used moment approximations break down or provide inaccurate moment estimates, rendering their use for parameter estimation impractical.

In addition to the bias typically introduced by using moment closure approximations for higher-order reaction networks, as pointed out in [28] moment-based methods are not always suitable for inference since they can result in inaccurate parameter fits and reduced predictive power compared to alternative approaches. For this reason [30] and [28] perform parameter inference by maximizing the likelihood of the observed data instead of comparing moments. This requires computing the likelihood function by solving the CME, which is not generally possible directly due to the infinite state space. Approximate likelihoods can be computed by solving a finite-dimensional approximation of the CME, the so-called Finite State Projection (FSP) [27], which is the approach taken in [30] and [28], but due to combinatorial explosion in the state space

this approach scales poorly for larger reaction systems. Our method does not require the likelihood function and instead relies on empirically approximating the steady state distribution using simulations, rendering it more scalable and more flexible than current likelihood-based approaches where population snapshot data is available.

Parameter estimation for spatial stochastic reaction networks is more challenging than for the CME. Approximation and inference method for the RDME are often based on the ones described above as the RDME can always be reformulated in terms of the CME. However, due to the additional spatial fluctuations many approximations such as moment equations will tend to fare worse for the RDME and specialized methods for the RDME would be desirable. Inference for Brownian dynamics simulations is frequently based on deterministic models such as rate equations [9] due to the complexity of the former, in spite of the fact that these two models often behave very differently.

3.2 Goals

There is currently no efficient direct method for performing parameter inference for general stochastic biochemical reaction networks based on population snapshot data. The inference methods presented above for the Chemical Master Equation, which is the simplest commonly used model for stochastic reaction networks, either rely on analytical approximations like the Diffusion Equation and moment closure schemes, adding a bias to the inference which is difficult to quantify, or they involve numerically approximating the CME via the Finite State Projection which drastically limits the size and complexity of reaction networks that can be studied.

Our goal is therefore to present an inference method which addresses the above points, i.e. satisfies the following criteria:

- It should not rely on approximations to the CME (unbiasedness)
- It should not require numerically solving or approximating the CME to compute likelihoods (tractability)
- It should not require a large number of simulations (efficiency)

We will address the above points in this work, constructing an inference method which satisfies all three criteria and demonstrate its effectiveness on examples taken from the literature. As our method does not directly depend on the CME formalism it can be applied to various other models of biological interest, including the RDME or even

Brownian dynamics, for which the methods discussed in Section 3.1 are inapplicable and which lacks specialized inference methods at the time of writing. We hope that our contribution will enable the scientific community to more efficiently tackle inference problems of the sort discussed in this thesis.

Chapter 4

Design

4.1 Population Snapshot Data

The goal of the project is to perform parameter inference for the Chemical Master Equation using population snapshot data, i.e. measurements of particle numbers in an independent population of cells. For simplicity we choose to focus on the steady-state distributions over particle counts in this project. Steady states are easier to observe in practice and timescale independent compared to the (often rapid) initial transients; observing and comparing these transients also necessitates knowledge of the initial state of the system which is not always given in biological applications.

If one considers particle numbers in a system with s species then the state space is equal to $\mathbb{N}^s$, a countable discrete set. In applications a probability distribution over $\mathbb{N}^s$ can be approximately represented by restricting it to a finite set of the form $\{0,\ldots,L\}^d \subseteq \mathbb{N}^s$ and normalizing; for sufficiently large L this restriction will approximate any given probability distribution over $\mathbb{N}^s$ arbitrarily closely. In practice the error incurred by empirically estimating the probability distributions involved dominates this truncation error even for moderate L.

Steady-state distributions for biochemical reaction networks can generally be obtained by observing the system in question for a sufficiently long time and computing time-averages[1]. For long times most observations will be taken when the system is close to its steady state irrespective of the initial conditions, leading to the convergence of the time-average to the true steady state distribution. Based on this observation we compute steady state distributions by simulating a single copy of the reaction system using the Stochastic Simulation Algorithm [13] until convergence, which we test for

[1] Assuming the system is ergodic, which is frequently the case in practice.

periodically using the p-Wasserstein distance for some $p \geq 1$. We remark that the exact choice of p is irrelevant in this case as the Wasserstein distances all metrize weak convergence for the distributions we will consider [53].

We remark that the steady state distribution of a reaction system does not change if all transition rates are rescaled by a common factor $c > 0$. Thus by observing the steady state one can only identify the rate constants up to a common scaling factor, which can be fixed if any one reaction rate is known. It is frequently possible to measure the degradation rate of reaction species experimentally, which removes this ambiguity - such an approach to inference is taken for example in [41]. In the remainder of this paper we will always assume that one reaction rate is given and estimate the remaining rate constants.

4.2 Constructing a Loss Function using Wasserstein Distances

Both the observed data and the simulator output come in the form of a histogram over the space $\mathbb{N}^s$ where s is the number of species in the model. Frequently one will only measure some of the species present in the system, in which case the observed data is a lower-dimensional marginal of said histogram, and in some cases it may not be possible to distinguish between several types of reactants (e.g. an activated vs. an inactivated gene), further decreasing its dimensionality. This leads to reduced histograms over $\mathbb{N}^{s'}$, with $s' \leq s$. In this section we will assume for simplicity that $s' = s$; this will not affect the rest of the discussion.

The loss function is a measure of the discrepancy between the observed histogram and the histogram output by the simulator for a specific parameter setting. We will consider Wasserstein distances between these histograms as our discrepancy measure, and therefore our loss function will be (a modification of) a suitable Wasserstein distance as described in this section.

As remarked in Section 2.2, defining Wasserstein distances requires a choice of a metric on the base space $\mathbb{N}^s$. The simplest metric is the discrete metric given by $C_{\vec{i},\vec{j}} = \mathbb{1}_{\vec{i} \neq \vec{j}}$, which assigns a pairwise distance of 1 to all points in $\mathbb{N}^s$; the associated p-Wasserstein metric equals the p-th root of the TV distance. A more interesting class

of metrics is given by the weighted ℓ^q-metrics defined by

$$d_{\ell^q}^{(\mathbf{w})}(\vec{i},\vec{j}) = \left(\sum_{k=1}^{s} w_k^q |i_k - j_k|^q\right)^{\frac{1}{q}} \tag{4.1}$$

for $q \geq 1$ and positive weights w_k. These metrics are all equivalent and yield equivalent classes of p-Wasserstein metrics for fixed p. We will use the notation $\mathcal{W}_p(P,Q)$ for the p-Wasserstein distance when the ground metric is understood.

Our reason for introducing weights in Eq. (4.1) is practical. One often finds that different reactant species have different abundances in a given reaction system, e.g. the number of protein molecules in cells is often significantly higher than that of the mRNA molecules they are translated from. Abundant species tend to display larger absolute fluctuations and optimising the Wasserstein distance with an unweighted ground metric will generally try to fit the distributions of abundant species first as this yields the largest immediate decrease in the observed loss. In order to prevent this we rescale the transportation cost for each species by a positive weight w_k, $k = 1,\ldots,s$, which should be chosen so that similar relative changes in the numbers of each species have an comparable effect on the Wasserstein distance. If this is the case the optimization routine will try to fit the distributions of all species simultaneously, improving the speed of convergence. In our experiments we set

$$w_k := \mathbb{E}_{\text{obs}}[n_k]^{-1}$$

where n_k is the number of reactants of species n_k and the mean is taken with respect to the observed (input) data.

The weighted p-Wasserstein distance still has the disadvantage of spanning several orders of magnitude in typical setups. As we will perform inference on parameters in log space (see Section 4.3) a moderate change in input values can result in an order of magnitude change in typical particle numbers and therefore in the loss. Given that our Gaussian Process will have to fit a function spanning a large range of values it will be difficult to accurately fit the region of interest around the expected optimum, where the loss will be close to 0. In order to counteract this we perform the transformation $x \mapsto \ln(1+x)$ to the Wasserstein distance; this is a monotone map that satisfies $\ln(1+x) \approx x$ for small x, i.e. does not significantly affect loss values close to the optimum while it reduces the range of the loss function far away from it. We will take this quantity rather than the raw Wasserstein distance as our loss function in what follows.

4.3 Bayesian Optimization of the Loss Function

At this stage we are given experimentally observed data P (an experimentally measured histogram over particle numbers) and a parametrized simulator model $Q(\mathbf{x})$ whose output is a histogram depending on the parameters $\mathbf{x}$. In our case the function $Q(\mathbf{x})$ is the empirically approximated steady state distribution of the system with parameters $\mathbf{x}$. Our goal is to find the choice of $\mathbf{x}$ minimizing our Wasserstein loss

$$L(\mathbf{x}) := \ln\left(1 + \mathcal{W}_p(Q(\mathbf{x}), P)\right) \tag{4.2}$$

We can approximate L for any set of parameters by simulating the Chemical Master Equation for sufficiently long times and computing the Wasserstein distance based on the estimated steady-state distribution. We will minimize the loss using Bayesian optimization; in this section we describe the details of our setup.

Assuming our task is to infer d different parameters we start by choosing a (bounded) search space $\mathcal{X} \subseteq \mathbb{R}^d$. Since reaction rates for the CME are positive and often range over orders of magnitude we use the log reaction rates for inference - in general this will depend on the type of parameters one wishes to estimate. We choose a reasonably large region in which the true parameters are expected to be found; if this is not the case after optimization one can enlarge the search space and continue optimization until the optimum is found. We then sample m points $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathcal{X}$ spread across the search space and evaluate $\tilde{L}(\mathbf{x}_1), \ldots, \tilde{L}(\mathbf{x}_m)$ in order to pre-train the GP. The choice of m usually depends on the dimension and the expected roughness of the loss landscape. We sample the points using Latin hypercubes in order to achieve uniform coverage of the parameter space.

The mean of the Gaussian process $\hat{L}$ is set to a constant equal to the mean of the $\tilde{L}(\mathbf{x}_i)$. For the covariance kernel we initially choose a squared exponential function of the form

$$k(\mathbf{x}, \mathbf{x}') = \sigma_y^2 \exp\left[-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \Lambda (\mathbf{x} - \mathbf{x}')\right] \tag{4.3}$$

The hyperparameters for this setup are the marginal variance σ_y^2 and the precision matrix Λ, restricted to be diagonal for simplicity. We fit the kernel hyperparameters by maximizing the marginal likelihood of the data $\mathcal{D}_{1:m}$, a common procedure for determining hyperparameters for Gaussian process regression [35].

Bayesian optimization now consists of repeatedly optimizing the acquisition function, computing the loss function $L(\mathbf{x}^*)$ at the optimum $\mathbf{x}^*$ and updating the Gaussian

process $\hat{L}$ with this information. In order to improve the fit of the Gaussian process we periodically refit the kernel after sampling enough new points. The resulting procedure is summarized in Algorithm 1.

Algorithm 1 Bayesian optimization-based parameter inference

Input: P_{obs} - observed histogram
Options: $N > 0$ - number of rounds before refitting GP hyperparameters
$m > 0$ - number of pre-training samples
$\varepsilon > 0$ - tolerance
Output: $\mathbf{x}^*$ - parameter estimate

---

sample $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathcal{X}$
for all $i = 1, \ldots, m$ **do**
  compute $\tilde{L}(\mathbf{x}_i)$ by running simulator
  $\mathcal{D}_i \leftarrow \{\mathbf{x}_i, \tilde{L}(\mathbf{x}_i)\}$
end for
fit mean and kernel hyperparameters of $\hat{L}$ **to** $\mathcal{D}_{1:m}$
$n \leftarrow m$
loop
  maximize $\alpha(\mathbf{x}; \hat{L} \mid \mathcal{D}_{1:n})$
  $\mathbf{x}_{n+1} \leftarrow \arg\max_{\mathbf{x}} \alpha(\mathbf{x}; \hat{L} \mid \mathcal{D}_{1:n})$
  compute $\tilde{L}(\mathbf{x}_{n+1})$ by running simulator
  if $\tilde{L}(\mathbf{x}_{n+1}) < \varepsilon$ **then**
    $\mathbf{x}^* \leftarrow \mathbf{x}_{n+1}$
    break
  update $\hat{L}$ **with** $\mathcal{D}_{n+1} = \{\mathbf{x}_{n+1}, \tilde{L}(\mathbf{x}_{n+1})\}$
  $n \leftarrow n + 1$
  if $n - m = 0 \pmod{N}$ **then**
    refit kernel hyperparameters of $\hat{L}$ **to** $\mathcal{D}_{1:n}$
end loop
return $\mathbf{x}^*$

4.3.1 Non-stationary Bayesian Optimization

One issue with the squared exponential kernel commonly used in Gaussian process regression is that it is stationary, that is, the covariance $k(\mathbf{x},\mathbf{x}')$ only depends on the relative difference $\mathbf{x}-\mathbf{x}'$. This makes it unsuitable for modelling functions which have different levels of roughness in different parts of parameter space. The loss functions we encountered often displayed a minimum located in a narrow valley surrounded by a large plateau where the loss showed little variation. A Gaussian process with a stationary kernel will either tend to choose very short length scales in order to fit the valley accurately, resulting in a lot of unnecessary uncertainty far away from the minimum and an inefficient optimization procedure due to overexploration, or it will pick large length scales to fit the plateau and treat the observations around the valley as statistical outliers, rendering the optimization routine unable to find the minimum.

Following [24] we thus consider a weighted superposition of two independent Gaussian processes, $f = w_g f_g + w_l f_l$ with

$$f_g \approx \mathrm{GP}(0, k_g(\mathbf{x},\mathbf{x}')) \qquad\qquad f_l \approx \mathrm{GP}(0, k_l(\mathbf{x},\mathbf{x}')) \tag{4.4}$$

and weight functions $w_g(\mathbf{x})$, $w_l(\mathbf{x})$ to be determined later. This enables us to decompose the Gaussian process into a global component $w_g f_g$ modelling the smooth large-scale behaviour of the loss function and a local component $w_l f_l$ that can fit the function accurately at the minimum. We choose squared exponential kernels $k_g(\mathbf{x},\mathbf{x}')$ and $k_l(\mathbf{x},\mathbf{x}')$ for these two Gaussian process components. The weights are parametrized as

$$w_g(\mathbf{x}) = \sqrt{\frac{1}{1+\nu(\mathbf{x})}} \qquad\qquad w_l(\mathbf{x}) = \sqrt{\frac{\nu(\mathbf{x})}{1+\nu(\mathbf{x})}} \tag{4.5}$$

for a nonnegative function $\nu(\mathbf{x})$. We set $\nu(\mathbf{x})$ to be a squared exponential basis function of the form

$$\nu(\mathbf{x}) = \exp\left[-\frac{1}{2}(\mathbf{x}-\mathbf{x}_\nu)^T \Lambda_\nu (\mathbf{x}-\mathbf{x}_\nu)\right] \tag{4.6}$$

for Λ_ν a symmetric positive-definite matrix, chosen to be diagonal in our case, and an anchor point $\mathbf{x}_\nu$.

The kernel of the total Gaussian process $f = w_g f_g + w_l f_l$ can be computed to be

$$k(\mathbf{x},\mathbf{x}') = w_g(\mathbf{x}) w_g(\mathbf{x}') k_g(\mathbf{x},\mathbf{x}') + w_l(\mathbf{x}) w_l(\mathbf{x}') k_l(\mathbf{x},\mathbf{x}') \tag{4.7}$$

As before we fit the hyperparameters by maximum likelihood estimation, constraining $\mathbf{x}_\nu$ to the location of the current best observation each time the kernel is refit. This is consistent with our observation that the loss function typically exhibits the largest amount of variation around the minimum.

Chapter 5

Implementation

5.1 Simulations

All simulators used for this project were written in Python using the `numpy` and `scipy` libraries. We tested the correctness of the simulators by applying them to various examples of reaction networks from the literature and verifying that our results reproduced the reference data in all cases.

In order to compute the steady state distribution of a network we run one instance of the system for a long time and compute time averages until convergence. We check for the latter by computing Wasserstein distances between the time averages at time points nT, $n = 1, 2, \ldots$, where T is a chosen epoch length. The epoch length is determined heuristically such that simulating the system for a few epochs results in convergence for typical parameter settings. Simulations are stopped when the distance at two consecutive time points becomes less than 0.02. In order to avoid wasting computation time for parameter settings yielding very bad fits to the observed data we also stop simulations if the distance at two consecutive time points becomes less than 2% of the approximate Wasserstein distance to the observed data. The numbers were chosen empirically based on a trade-off between accuracy and computation time. As computing joint Wasserstein distances in multiple dimensions can be expensive, in our simulations we computed the sum of the Wasserstein distances of the marginals when checking for convergence; we found that this did not measurably affect results at the chosen tolerances.

5.2 Wasserstein distances

In this section we discuss our method for computing Wasserstein distances. Starting with two empirically estimated histograms P, Q over $\mathbb{N}^s$ we note that both P and Q will always have compact support, i.e. there exists $L \geq 0$ such that $P_{\vec{i}}, Q_{\vec{i}} = 0$ if $i_k \geq L$ for any k. In this case one can view both P and Q as elements of a L^s-dimensional vector space and T as a $L^s \times L^s$ matrix.

Eq. (2.6) is a linear optimization problem involving L^{2s} variables and $L^{2s} + 2L^s$ constraints, rendering naive attempts at optimization impracticable for realistic histogram sizes. Our solver is based on the Sinkhorn Algorithm [7] which computes the optimum of a relaxed version of Eq. (2.6):

$$d_c^{(\varepsilon)}(P,Q) = \inf_{T \in U(P,Q)} \langle C, T \rangle - \varepsilon H(T) \tag{5.1}$$

where the regularizer $H(T)$ is defined as

$$H(T) = -\sum_{\vec{i},\vec{j}} T_{\vec{i},\vec{j}} \log T_{\vec{i},\vec{j}}$$

with the (unusual) convention that $H(T) = -\infty$ if one of the entries of T is 0 or negative. The additional term $H(T)$, equal to the entropy for strictly positive transport plans, is a regularizer that modifies the objective in order to make the solution more tractable; one can show that the solutions to Eq. (5.1) converge to the solution of the unregularized problem as $\varepsilon \to 0$. Since the solution of 5.1 is always a strictly positive transport plan and $H(T)$ is bounded by $2s \log L$ for a strictly positive $L^s \times L^s$-matrix T the introduction of the regularizer does not introduce significant approximation errors for small ε.

The Sinkhorn algorithm is an iterative solver for Eq. (5.1), a description of which can be found in [7]. One drawback of this algorithm is that the number of iterations required for convergence increases as $\varepsilon \to 0$; in order to compute the solution to Eq. (5.1) for small ε we therefore use an annealing procedure starting with a large value of ε (typically $\varepsilon = 10$) and multiplying it by an annealing factor $\delta < 1$ at each step. We run each step until the two margin constraints (2.4) are satisfied to a specified tolerance ε' in the ℓ^1-norm. To improve convergence speed for small ε we use the overrelaxation method presented in [49].

Due to the large numerical range encountered in the Sinkhorn algorithm all computations are performed in log-space. In addition, when computing the p-Wasserstein

distance with a weighted ℓ^q ground metric for $p = q$ the cost matrix decomposes as a sum of terms, one for each histogram dimension,

$$C^p_{\vec{i},\vec{j}} = \sum_{k=1}^{s} w_k^p |i_s - j_s|^p \tag{5.2}$$

which permits vectorization of the relevant matrix-vector products in the Sinkhorn algorithm. For this reason and since the exact choice of q does not matter (see Section 2.2) we will generally set $p = q$ for efficiency.

We remark that Wasserstein distances in one dimension can be evaluated using a simpler and more straightforward algorithm: if F and G are the cumulative distribution functions of two probability distributions f and g on $\mathbb{R}$, respectively, then one can prove [53] that

$$\mathcal{W}_p(f,g) = w_1 \left(\int_0^1 \left| F^{-1}(x) - G^{-1}(x) \right|^p \mathrm{d}x \right)^{1/p} \tag{5.3}$$

For discrete histograms this integral can be computed exactly with little overhead, rendering Wasserstein distances in one dimension especially convenient from a computational point of view.

The Sinkhorn algorithm as described above can become computationally very demanding for large histogram sizes, especially in terms of memory usage; we therefore briefly discuss various remedies for this. The iterations consist mostly of repeated matrix-vector operations and can be easily implemented on GPUs, potentially resulting in noticeable speed gains compared to using a CPU. For systems with large particle numbers one can coarse-grain the histogram by binning particle numbers and approximating the Wasserstein distances using the coarsened histograms. Finally one can compute Wasserstein distances between lower-dimensional marginals separately and minimize the sum of the distances. While this method may potentially lose information about correlations between different species we have observed it to yield near identical results for the three-stage gene expression model (cf. Section 6.2). Since it is difficult to experimentally measure joint distributions of several distinct species, fitting marginalized distributions is a reasonable approach in these situations and we found it to perform well in practice.

5.3 Bayesian optimization

As for the simulators we used Python, `numpy` and `scipy` to implement a framework for Gaussian process-based Bayesian Optimization. We used the `scipy.optimize`

package to optimize the acquisition function at every round. The acquisition function is in general not convex, which can lead to the library optimization routine getting stuck in local minima. To mediate this we ran each optimization several times with randomly sampled initial conditions. We verified the correctness of the results post hoc by re-running the optimization at each round with a significantly increased number of restarts and checking that the found optimum agreed with the one originally proposed; with the chosen settings we found this to be the case for virtually all rounds.

The Gaussian process observation noise in Eq. (2.9) is set to 0.03. This value was chosen as it is on the order of magnitude of the typical observation noise expected with the tolerances given in Section 5.1 and was found to work well for all experiments in this thesis. It is possible to optimize the observation noise along with the remaining hyperparameters of the Gaussian process, but we did not record any significant differences in performance between the two approaches.

The jitter parameter β in the Expected Improvement acquisition function in Eq. (2.11) was set to 0.01, a typical value frequently encountered in the literature and e.g. the default in the Python package `scikit-learn` at the time of writing.

Chapter 6

Evaluation

6.1 Experiments

In this section we apply our method in order to infer parameters for two different biochemical reaction networks using the CME: the classical three-stage gene expression model [42] and a bursty positive feedback loop [17]. In addition we use our method to investigate a spatial model of a reaction network in Section 6.4 using the RDME mentioned in Section 2.1.1.

In all our experiments we chose the 1-Wasserstein distance with a weighted ℓ^1 ground metric on $\mathbb{N}^s$. Computing p-Wasserstein distances for $p > 1$ using the Sinkhorn is more challenging as the increased numerical range of the cost function can lead to numerical instabilities. As described in Section 4.2 the weight for each species is chosen to be inversely proportional to the mean particle number in the reference distribution, $w_k \propto \mathbb{E}_{\text{obs}}[n_k]^{-1}$.

6.2 Three-stage gene expression model

Our first experiment involves identifying the parameters in the three-stage model of gene expression found in [42] and described by the following reactions:

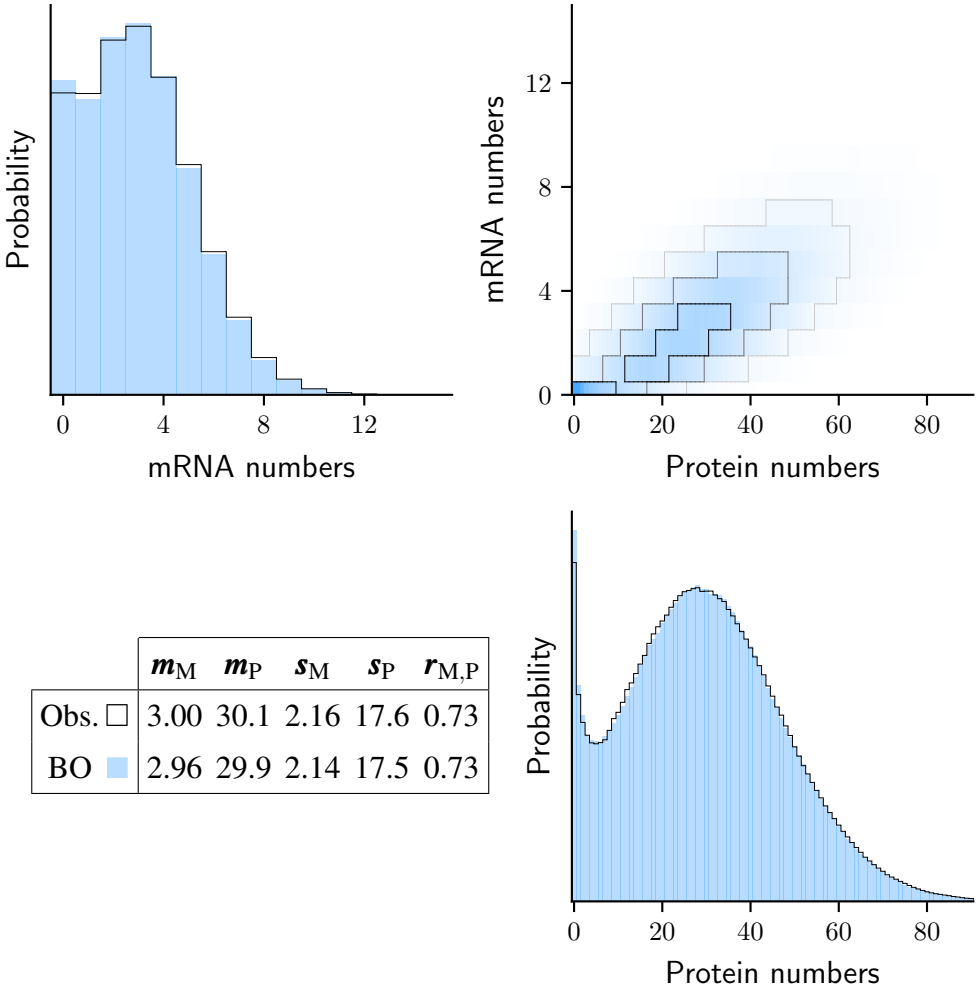
$$G \xrightarrow{\rho_m} G + M \qquad M \xrightarrow{\rho_p} M + P \qquad M \xrightarrow{\delta_m} \emptyset \qquad (6.1)$$
$$G \underset{\sigma_a}{\overset{\sigma_d}{\rightleftharpoons}} G^* \qquad P \xrightarrow{1} \emptyset$$

This model consists of four reactant species: a gene in an activated (G) and inactivated (G^*) form, mRNA (M) and protein (P). It differs from the example considered in Section 2.1 by the addition of the inactivated state G^* of the gene which is not able to

produce mRNA. The protein degradation rate is set to 1, whereas the remaining five rate constants are to be estimated from the measured joint distribution over mRNA and proteins.

We fixed ground truth values for all parameters were (taken from Fig. 3 in [42] with $\gamma = 1$) and obtained a synthetic reference distribution using the SSA. We then applied our method to recover the parameter values based on the observed distribution. The search range for the parameters was set to cover two orders of magnitude per dimension and included the ground truth values; the results can be seen in Fig. 6.1. We re-ran the same experiment comparing marginal mRNA and protein numbers (without using the joint distribution) and obtained very similar results, suggesting that it is not always necessary to measure joint distributions to perform parameter inference for the CME if the marginals are fit precisely.

The three-stage gene expression model is a typical example of a linear reaction system since it does not include any bimolecular reactions. It is therefore possible to compute the steady state moments of protein and mRNA numbers exactly without having to resort to moment closure approximations; parameters can then be estimated from population snapshot data by standard optimization methods. This method is significantly faster than our approach as it does not require running simulations, but the three-stage gene expression model demonstrates that our method works for nontrivial reaction systems.



	m_M	m_P	s_M	s_P	$r_{M,P}$
Obs. □	3.00	30.1	2.16	17.6	0.73
BO ■	2.96	29.9	2.14	17.5	0.73

(a)

	ρ_m	σ_d	σ_a	ρ_p	δ_m
Range	0.1–10	0.01–1	0.01–1	1–100	0.1–10
GT □	4.00	0.20	0.60	10.0	1.00
BO ■	4.13	0.18	0.56	10.1	1.06

(b)

Figure 6.1: Results for the three-stage gene expression model in Eq. (6.1). **(a)**: Steady steady state distribution over mRNA and protein numbers for the observed data (contours) and the parameters estimated using Bayesian optimization (shaded). The table shows the means ($\boldsymbol{m}$) and standard deviations ($\boldsymbol{s}$) of mRNA and protein numbers as well as their Pearson correlation coefficient ($\boldsymbol{r}$). Both the shape and the moments of the observed distribution are matched by our method. **(b)**: Ground truth and estimated parameters for the observed data and the chosen search ranges. The results were obtained after 362 rounds starting with 300 initial samples, where the GP kernel was refit every 75 rounds during optimization.

6.3 Bursty feedback loop

In our second experiment we considered a genetic feedback loop described by the following list of reactions, taken from [17] and [4]:

$$G_u \xrightarrow{\rho_u} G_u + kP \qquad G_b \xrightarrow{\rho_b} G_b + kP \qquad k \sim \text{Geom}(b) \tag{6.2}$$
$$G_u + P \underset{\sigma_u}{\overset{\sigma_b}{\rightleftharpoons}} G_b \qquad P \xrightarrow{1} \emptyset$$

The species in this model are a gene in the bound (G_b) and unbound (G_u) state as well as a protein (P) produced by the gene. This system describes a protein which can bind to its gene and hence influence its own transcription rate. The intermediate steps involving mRNA are not considered explicitly in the above example, instead they are reflected in the fact that the number of proteins produced at each translation event follows a geometric distribution with mean b,

$$p(k) = \frac{b^k}{(1+b)^{k+1}} \qquad (k \geq 0)$$

which is an approximation of mRNA-mediated protein production when the lifetime of mRNA is very short compared to the mean protein lifetime [10].

We will compare our approach to moment-based inference methods for this problem. Due to the bimolecular reaction $G_u + P \to G_b$ the moment equations for this system are not directly solvable: computing the evolution of any given moment requires knowledge of some higher-order moments, leading to the infinite hierarchy of moment equations mentioned in Section 3.1. One therefore has to use a moment closure scheme to arrive at a low-dimensional set of equations that approximate the first few moments.

The system (6.2) displays different types of behaviour depending on whether $\rho_u > \rho_b$ (negative feedback) or $\rho_u < \rho_b$ (positive feedback). In [4] the authors compare different moment closure schemes as well as the Linear Mapping Approximation [3] and show that for the negative feedback loop it is possible to efficiently obtain accurate parameter estimates using these approximations. In this section we will focus on the positive feedback case, which we found to be challenging for the approach presented in [4].

Positive feedback in this system can result in strong sensitivity to parameter values (see Fig. 6.2). We chose parameters that resulted in the gene spending non-negligible amounts of time in both the bound and the unbound state. This regime (which we shall

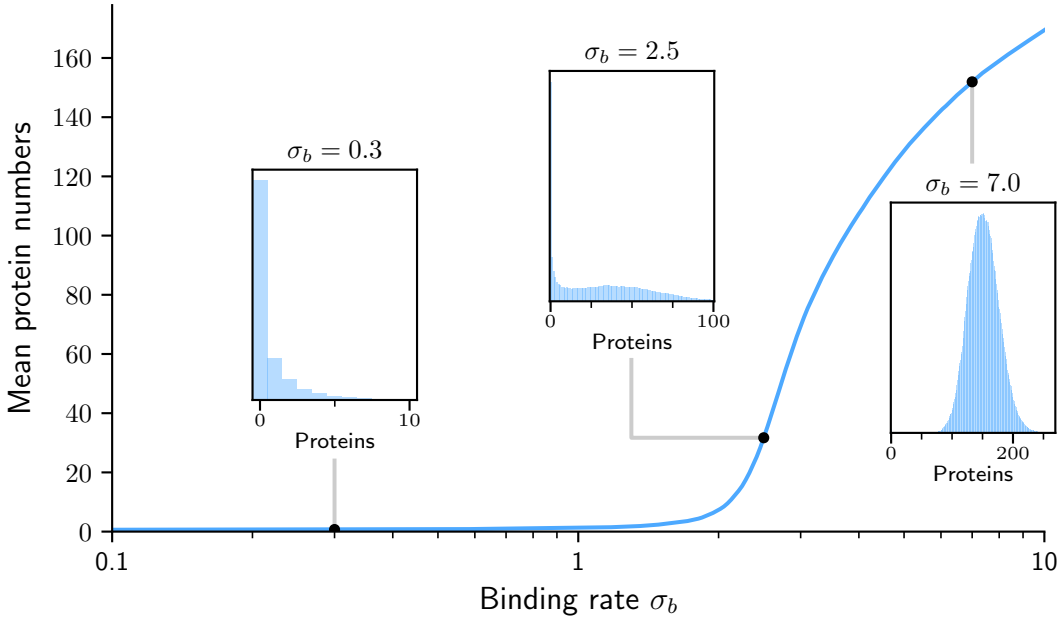


Figure 6.2: Mean protein numbers plotted against the binding rate σ_b for the bursty positive feedback loop. There is a sharp increase around $\sigma_b \approx 2$ as the system switches from being in the inactivated state most of the time to the activated state. The steady state distributions differ qualitatively depending on σ_b, with the intermediate region characterized by bimodal protein number distributions. The values of the remaining parameters are $\rho_u = 0.3$, $\sigma_u = 400$, $\rho_b = 105$, $b = 2$.

call the intermediate regime) is characterized by a bimodal steady state distribution over protein numbers which changes rapidly with the parameters σ_b and ρ_b.

In order to test how well moment closure methods can approximate the positive feedback loop we applied six different moment closure schemes from [4]: conditional derivative matching and conditional Gaussian [48], both conditioned on either the bound or the unbound states of the gene (denoted CDM1 and CDM2, resp. CG1 and CG2), as well as unconditional Gaussian (Gauss) and derivative matching (DM, [44]). We also considered the Linear Mapping Approximation (LMA, [3]) as it yields a set of moment equations that can be solved directly, similar to classical moment closure schemes. In addition we tested the conditional negative binomial approximation, again conditioned on both the bound and the unbound state of the gene, which we respectively denote CNB1 and CNB2. For more details on these moment closure schemes and how they are applied to estimate moments we refer to the cited papers.

We found that none of the nine methods tested were able to accurately predict mean

and standard deviation of the protein number distribution for our system (Fig. 6.3). While many were able to model the system outside the intermediate regime, the presence of large fluctuations in that regime significantly decreased the accuracy of the methods. The closed moment equations are nonlinear and usually admit multiple solutions, yielding complex, negative or outlandishly high predictions for moments. This is similar to the scenario tested in [39] where the considered moment closure schemes failed to yield unique solutions in general. We found that with the exception of the LMA all tested moment closure methods broke down in different parts of the intermediate regime, and while the LMA itself always yields an interpretable solution it often returns inaccurate predictions of moments (cf. Fig. 6.3).

Given that none of the moment closure methods used for the negative feedback case capture the intermediate regime in the positive feedback case one has to rely on alternative inference methods for our problem. The complicated dependence of the steady state on the parameters makes this task challenging in general. In the intermediate regime very small changes in σ_b or ρ_b will typically lead to large changes in the steady state distribution, while in the regime where the gene is mostly unbound the system will virtually be independent of σ_b (cf. Fig. 6.2). Hence the loss landscape looks very different at different points in parameter space, causing problems for both global and local optimization approaches. A grid search for σ_b for example would need to sample values at very short intervals in order to find the correct value in the intermediate regime, while a local optimization routine would likely get stuck if initialized in a region where the loss function is flat.

We tested our method on this problem by performing joint inference over σ_b, ρ_u and σ_u in the intermediate regime, based on observing the (marginal) protein number distribution. The results are shown in Fig. 6.4. We recovered a set of parameters that yield a steady state protein distribution closely matching the input data, even though the estimated parameters themselves do not agree well with the ground truth. This suggests that the positive feedback loop suffers from parameter non-identifiability even in the intermediate regime; we remark that changing only σ_b from its ground truth value does not yield a similar steady state distribution as can be seen in Fig. 6.2.

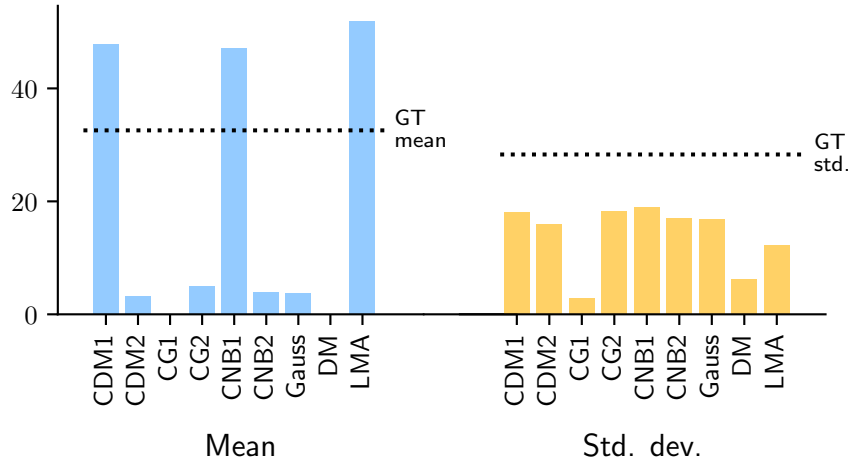
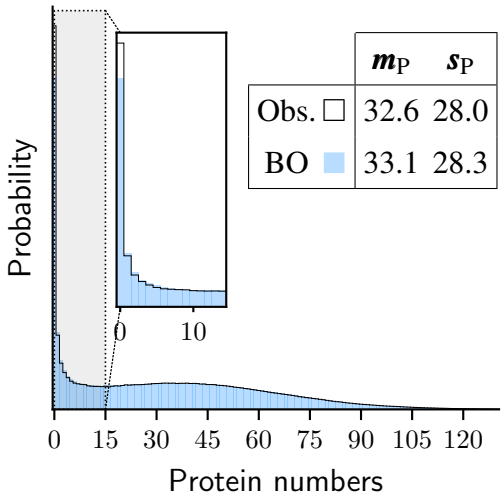


Figure 6.3: Eight different moment closure schemes and the LMA applied to the positive feedback loop, using ground truth the parameters given in Fig. 6.4. CG1 and DM failed to yield a solution predicting a positive mean. Even the best approximation (CNB1) is more than 30% off in its estimate of both the mean and the standard deviation of protein numbers.



	m_P	s_P
Obs. □	32.6	28.0
BO ■	33.1	28.3

(a)

	ρ_u	σ_u	σ_b
Range	0.15–15	10-1000	0.1–10
GT □	0.30	400	2.50
BO ■	0.45	355	2.22

(b)

Figure 6.4: Results for the bursty positive feedback loop. **(a)**: Steady state distribution over protein numbers for the observed data (contours) and the estimated parameters (shaded). The table gives the mean and standard deviation of the two distributions. **(b)**: Ground truth and estimated parameters for the observed data. The results were obtained after 130 rounds starting with 75 initial samples, where the kernel was refit every 25 rounds during optimization. The remaining parameters are $\rho_b = 105$, $b = 2$ (cf. Fig. 6.2).

6.4 Spatial Dimerization Model

In this experiment we considered inference for an extension of the Chemical Master Equation suitable for modelling spatially extended systems, namely the Reaction-Diffusion Master Equation (RDME). The Chemical Master Equation assumes that the reaction system in question is always spatially homogeneous and well-stirred, i.e. that particles are uniformly distributed in the reaction volume at all times. While this is frequently the case for small reaction volumes or large diffusion coefficients, it is well-known that this assumption does not always hold in practice. The Reaction-Diffusion Master Equation is a modification of the CME that partitions the reaction volume into smaller cells within which the reactions are simulated using the CME, where particles can jump between adjacent cells to simulate diffusion in the reaction volume.

A detailed description of the RDME is given in Appendix A.1. Since the RDME is a Markov chain model like the CME simulations can be performed using the SSA. For a fixed partition of the reaction volume into cells the parameters of the RDME consist of the parameters of the CME as well as the diffusion coefficients of the reactants, which determine the rate of jumps between cells.

While the RDME can yield more accurate result than the CME in scenarios where particles diffuse slowly, the results often depend on the chosen size of the cells. The RDME tries to improve on the CME by dividing the reaction volume into smaller cells in which the CME will be more accurate, but since two particles can only react if they are in the same cell, bimolecular reactions become exceedingly rare when the cell size becomes too small. One can therefore not choose an arbitrarily small cell size; generally speaking the size of each cell should not be smaller than the physical reaction radius of the bimolecular reactions. For reaction networks without bimolecular interactions the RDME yields the same results as the CME since the spatial position of the particles is irrelevant for zeroth and first-order reactions.

To test inference for the RDME we consider a simple example of a reaction network involving bimolecular reactions, namely the dimerization model consisting of a sole species A together with the following two reactions:

$$\emptyset \xrightarrow{1} A \qquad\qquad A + A \xrightarrow{\delta} \emptyset \tag{6.3}$$

We use Brownian dynamics (BD) simulations in order to obtain reference data for this system in two dimensions. Brownian dynamics simulations (described in Appendix A.2) are a class of models that simulate molecules as diffusing point particles

undergoing reactions and interacting with each other, and are thus physically more detailed than either the CME or the RDME.

In order to investigate the effect of the chosen grid size for the RDME we estimated the dimerization rate δ for a variety of diffusion constants and grid sizes using our method. The results can be seen in Fig. 6.5. From the figure we can deduce the effect of introducing spatial heterogeneity by passing from the one-compartment CME to the multi-compartment RDME is less prominent for large diffusion coefficients since in this regime the particles diffuse quickly enough to reach spatial equilibrium between reactions, in which case the dynamics are described accurately by the CME. However for small diffusion coefficients the effective dimerization rate has to be increased with finer grid sizes since particles have to be in one compartment to dimerize and such occurrences become less frequent the smaller the compartments get. The exact dependence of the effective dimerization rate on the number of cells for this simple dimerization model can be read off from Fig. 6.5.

As an analytical solution for the steady-state distribution of the dimerization model has not been found in the presence of more than one compartment in the RDME, there is currently no known formula for the effective dimerization rate; while our approach does not yield a closed-form solution for this problem it is able to compute it numerically. This illustrates how parameter estimation can be used to analyse the behaviour of a given biophysical model for a simple example.

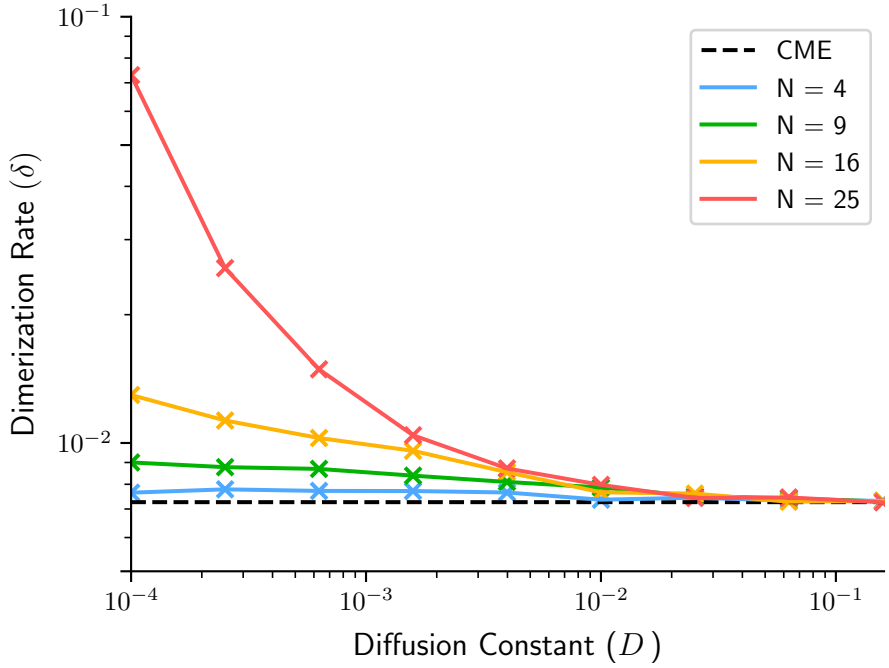


Figure 6.5: The effective dimerization rate in the RDME depends on the grid size and the diffusion coefficient. The figure shows the dimerization rate δ for the RDME version of the simple dimerization model as estimated by our method for different combinations of the grid size N^d and the diffusion constant D in a square reaction container in two dimensions with periodic boundary conditions. The case $N = 1$ is equivalent to the CME (dashed line). The results illustrate how the dynamics described by the RDME differ depending on the spatial discretization scale. For fixed N the RDME is asymptotically equivalent to the CME as $D \to \infty$. At every data point the dimerization rate δ was obtained after 30 rounds starting with 15 initial samples, where the kernel was refit every 15 rounds. The input data were obtained using BD simulations with reaction rate $\lambda = 1.9$ and reaction radius $\rho = 0.1$ for the dimerization reaction (see Appendix A.2).

Chapter 7

Conclusion

7.1 Discussion

In this thesis we developed a general-purpose method for parameter estimation for stochastic biochemical reaction networks. We used Wasserstein distances to quantify the discrepancy between the observed data and the simulator output for different parameters and constructed a probabilistic model of the Wasserstein distance at unexplored parameter settings by training a Gaussian process with these data. Bayesian optimization then enabled us to (i) iteratively choose parameter settings that are likely to be close to the optimum, (ii) evaluate the system at the chosen parameters and (iii) update the model until we found parameters consistent with the observations. While we focused on parameter inference for the Chemical Master Equation and the Reaction-Diffusion Master Equation our approach can be used with any simulator-based model, including e.g. Brownian dynamics.

We tested the presented method on a standard example from the literature, the three-stage gene expression network (cf. Section 6.2), recovering results in close agreement with the ground truth. While the chosen example can be treated efficiently using moment equations due to the linearity of the reactions, this experiment shows that our approach can be applied to examples of real biological interest. Our second example, the positive feedback loop considered in Section 6.3, shows that it can also be used with reaction networks which exhibit high parameter sensitivity and are not well captured by known approximations. Our analysis of nine different moment closure schemes failed to yield accurate results for this example, showcasing the fact that inference methods based on approximations do not work in all cases.

In addition to parameter estimation for the CME, in Section 6.4 we considered a

spatial dimerization system modelled using the RDME. We applied our method to estimate the effective dimerization rate for various combinations of grid sizes and diffusion coefficients, and based on the results we were able to visualize how these settings affect the effective dimerization rate. This shows that our method can be applied robustly to models that are more complicated than the CME and that it can be used to gain understanding about simulator-based models which are not necessarily tractable analytically.

As our method only relies on having access to a simulator and does not require the computation of likelihoods it is suited to biophysical models such as Brownian dynamics which can be sampled from but for which likelihoods are unavailable. Given the fact that many simulator models are expensive to evaluate, Bayesian optimization provides an effective method to perform inference as it tries to minimize the number of simulations needed. There is currently no literature on efficient parameter estimation for Brownian dynamics or related biophysical models of stochastic reaction networks [9], and we hope that the approach presented will provide a first stepping stone in this direction.

7.2 Limitations

While our method can be applied to any simulator-based model for reaction networks it will usually not be as efficient as standard approximation-based inference methods currently in use. Various approximation methods for stochastic reaction networks have successfully been used for parameter estimation in several classes of reaction networks, yielding excellent parameter estimates in close agreement with observations; as these typically involve solving a set of coupled ODEs or SDEs they are significantly easier to evaluate and can be optimized using standard numerical methods. In comparison, computations of steady-state distributions using the SSA are rather expensive and require the use of specialized optimization methods such as Bayesian optimization. Further work on fast and accurate approximation schemes for the CME will remain indispensable in order to improve the ease and efficiency of parameter estimation for general reaction networks in practice.

Bayesian optimization has previously been used for likelihood-free inference e.g. in [18]. One potential limitation of global optimization approaches like Bayesian optimization is that they are often difficult to apply to high-dimensional problems. The number of evaluations needed until convergence usually scales with the dimension of

the parameter space, reducing their usability for problems with many parameters. The effectiveness of Bayesian optimization in particular depends strongly on the possibility of modelling complex high-dimensional functions using a Gaussian process given a limited amount of evaluations, not a trivial task in general. We are positive that continuing research on non-stationary Bayesian optimization methods [24, 47] will enable us to deal with these problems more effectively in the future.

In our approach we used Wasserstein distances to match distributions, one problem with these being that the computations can become expensive for large histogram dimensions. Remedies for this problem were discussed in Section 5.2: implementing the algorithm on GPUs, coarse-graining histograms or marginalizing to obtain lower-dimensional histograms with a subsequent loss of information. Since simultaneous particle count measurements for multiple chemical species are difficult to obtain experimentally at the time of writing we anticipate that the main limitations in this regard will be related to obtaining the data. However, as mentioned in Section 6.2 we were able to recover very good parameter estimates even based on (separately measured) marginal particle counts for the three-stage gene expression model, suggesting that the availability of marginal particle counts does not necessarily restrict the effectiveness of parameter inference. There is a significant amount of ongoing research aimed at improving the efficiency of Wasserstein distance computations in multiple dimensions, and we are positive that the mentioned computational difficulties will be alleviated as further progress is being made.

One important aspect of our approach to parameter estimation is the issue of uniqueness of solutions. We saw in Section 6.3 that the parameter estimates we obtained for the positive feedback loop were not in exact agreement with their ground truth values in spite of the fact that both parameter settings yield near identical steady states. In such a scenario our method will only return one set of parameters consistent with the observations. While one can always continue Bayesian optimization to find the remaining optima, this is not a standard approach and a systematic treatment of the problem of multiple optima will require alternative solutions. Bayesian inference is a promising framework for this and can be applied for moment-based inference e.g. as in [4], but it is not immediately clear how to extend our approach to this setting, a limitation of our current Wasserstein-based approach. We do point out, however, that using Wasserstein distances as opposed to e.g. moments to compare distributions ensures that the results returned by our method are fully consistent with the data and not spurious solutions due to the use of insufficient summary statistics.

7.3 Further Work

In our thesis we performed parameter inference for stochastic reaction networks based on comparing steady-state particle count distributions. A possible extension of this method would lie in considering particle distributions at multiple fixed time-points. As pointed out in Section 4.1, steady state distributions can only determine relative values of reaction rates, and in order to determine absolute reaction rates from scratch it is necessary to consider time series data. Furthermore, it has been shown [29] that measurements at multiple time points can aid in the identification of parameters. We expect that our method can be extended to this kind of data without significant modifications, e.g. by constructing a loss function as the sum of the Wasserstein distances between the distributions at different time points.

An immediate application of our work would be to use experimental data to perform parameter inference on Brownian dynamics models, for which analytical solutions and/or approximation methods are currently few and far in between (cf. [45]). Even simple Brownian dynamics models such as the one described in Section 6.4 are significantly more expensive to evaluate than their RDME equivalents and in the absence of alternative methods, Bayesian optimization is a prime choice for parameter estimation due to its sample efficiency. We believe that research in this area would be of significant interest to the community and hope to be able to pursue this matter in the future.

Bibliography

[1] A. Adan, G. Alizada, Y. Kiraz, Y. Baran, and A. Nalbant, "Flow cytometry: Basic principles and applications," *Crit. Rev. Biotechnol.* 37(2): 163–176, 2017.

[2] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017. arXiv: 1701.07875.

[3] Z. Cao and R. Grima, "Linear mapping approximation of gene regulatory networks with stochastic dynamics," *Nat. Commun.* 9(1): 3305, 2018.

[4] ——, "Accuracy of parameter estimation for auto-regulatory transcriptional feedback loops from noisy data," *J. R. Soc. Interface*, 16(153): 20180967, 2019.

[5] P. J. Choi, L. Cai, K. Frieda, and X. S. Xie, "A stochastic single-molecule event triggers phenotype switching of a bacterial cell," *Science*, 322(5900): 442–446, 2008.

[6] E. Cinquemani, "Identifiability and reconstruction of biochemical reaction networks from population snapshot data," *Processes*, 6(9): 136, 2018.

[7] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," *Advances in Neural Information Processing Systems*, 26: 2292–2300, 2013.

[8] M. B. Elowitz, "Stochastic gene expression in a single cell," *Science*, 297(5584): 1183–1186, 2002.

[9] R. Erban, "From molecular dynamics to Brownian dynamics," *Proc. R. Soc. A*, 470(2167): 20140036, 2014.

[10] N. Friedman, L. Cai, and X. S. Xie, "Linking stochastic dynamics to population distribution: An analytical framework of gene expression," *Phys. Rev. Lett.* 97(16): 168302, 2006.

[11] C. Frogner, C. Zhang, H. Mobahi, M. Araya-Polo, and T. Poggio, "Learning with a Wasserstein loss," 2015. arXiv: 1506.05439.

[12] F. Fröhlich, P. Thomas, A. Kazeroonian, *et al.*, "Inference for stochastic chemical kinetics using moment equations and system size expansion," *PLOS Comput. Bio.* 12(7): e1005030, 2016.

[13] D. T. Gillespie, "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions," *J. Comput. Phys.* 22(4): 403–434, 1976.

[14] A. Golightly and D. J. Wilkinson, "Bayesian sequential inference for stochastic kinetic biochemical network models," *J. Comput. Bio.* 13(3): 838–851, 2006.

[15] ——, "Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo," *Interface Focus*, 1(6): 807–820, 2011.

[16] J. González, J. Longworth, D. C. James, and N. D. Lawrence, "Bayesian optimization for synthetic gene design," 2015. arXiv: 1505.01627.

[17] R. Grima, D. R. Schmidt, and T. J. Newman, "Steady-state fluctuations of a genetic feedback loop: An exact solution," *J. Chem. Phys.* 137(3): 035104, 2012.

[18] M. U. Gutmann and J. Corander, "Bayesian optimization for likelihood-free inference of simulator-based statistical models," *J. Mach. Learn. Res.* 17(125): 1–47, 2016.

[19] S. A. Isaacson, "Relationship between the Reaction–Diffusion Master Equation and particle tracking models," *J. Phys. A*, 41(6): 065003, 2008.

[20] N. van Kampen, *Stochastic Processes in Physics and Chemistry*, 3rd ed. Elsevier, 2007.

[21] D. J. Kiviet, P. Nghe, N. Walker, *et al.*, "Stochasticity of metabolism and growth at the single-cell level," *Nature*, 514(7522): 376–379, 2014.

[22] M. Komorowski, B. Finkenstädt, C. V. Harper, and D. A. Rand, "Bayesian inference of biochemical kinetic parameters using the Linear Noise Approximation," *BMC Bioinformatics*, 10(1): 343, 2009.

[23] S. Marguerat, A. Schmidt, S. Codlin, *et al.*, "Quantitative analysis of fission yeast transcriptomes and proteomes in proliferating and quiescent cells," *Cell*, 151(3): 671–683, 2012.

[24] R. Martinez-Cantin, "Local nonstationarity for efficient Bayesian optimization," 2015. arXiv: 1506.02080.

[25] H. H. McAdams and A. Arkin, "It's a noisy business! Genetic regulation at the nanomolar scale," *Trends in Genetics*, 15(2): 65–69, 1999.

[26] T. A. E. Moselhy and Y. M. Marzouk, "Bayesian inference with optimal maps," *J. of Comput. Phys.* 231(23): 7815–7850, 2012.

[27] B. Munsky and M. Khammash, "The Finite State Projection algorithm for the solution of the Chemical Master Equation," *J. Chem. Phys.* 124(4): 044104, 2006.

[28] B. Munsky, G. Li, Z. R. Fox, D. P. Shepherd, and G. Neuert, "Distribution shapes govern the discovery of predictive models for gene regulation," *PNAS*, 115(29): 7533–7538, 2018.

[29] B. Munsky, B. Trinh, and M. Khammash, "Listening to the noise: Random fluctuations reveal gene network parameters," *Mol. Syst. Biol.* 5: 318, 2009.

[30] G. Neuert, B. Munsky, R. Z. Tan, *et al.*, "Systematic identification of signal-activated stochastic gene regulation," *Science*, 339(6119): 584–587, 2013.

[31] K. Öcal, R. Grima, and G. Sanguinetti, "Parameter estimation for biochemical reaction networks using Wasserstein distances," 2019. arXiv: 1907.07986.

[32] A. Pacchiano, J. Parker-Holder, Y. Tang, *et al.*, "Wasserstein Reinforcement Learning," 2019. arXiv: 1906.04349.

[33] G. Peyré and M. Cuturi, "Computational optimal transport," *Found. Trends Mach. Learn.* 11(5): 355–607, 2019.

[34] R. Ramaswamy, N. González-Segredo, I. F. Sbalzarini, and R. Grima, "Discreteness-induced concentration inversion in mesoscopic chemical systems," *Nat. Commun.* 3: 779, 2012.

[35] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*, ser. Adaptive computation and machine learning. Cambridge, Mass: MIT Press, 2006.

[36] J. Ruess and J. Lygeros, "Moment-based methods for parameter inference and experiment design for stochastic biochemical reaction networks," *ACM Trans. Model. Comput. Simul.* 25(2): 8:1–8:25, 2015.

[37] C. Schilling, S. Bogomolov, T. A. Henzinger, A. Podelski, and J. Ruess, "Adaptive moment closure for parameter inference of biochemical reaction networks," *Biosystems*, Selected papers from the Computational Methods in Systems Biology 2015 conference, 149: 15–25, 2016.

[38] D. Schnoerr, G. Sanguinetti, and R. Grima, "Validity conditions for moment closure approximations in stochastic chemical kinetics," *J. Chem. Phys.* 141(8): 084103, 2014.

[39] ——, "Comparison of different moment-closure approximations for stochastic chemical kinetics," *J. Chem. Phys.* 143(18): 185101, 2015.

[40] ——, "Approximation and inference methods for stochastic biochemical kinetics - a tutorial review," *J. Phys. A*, 50(9): 093001, 2017.

[41] B. Schwanhäusser, D. Busse, N. Li, *et al.*, "Global quantification of mammalian gene expression control," *Nature*, 473: 337–342, 2011.

[42] V. Shahrezaei and P. S. Swain, "Analytical distributions for stochastic gene expression," *PNAS*, 105(45): 17256–17261, 2008.

[43] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proc. IEEE*, 104(1): 148–175, 2016.

[44] A. Singh and J. P. Hespanha, "A derivative matching approach to moment closure for the stochastic logistic model," *Bull. Math. Biol.* 69(6): 1909–1925, 2007.

[45] S. Smith and R. Grima, "An exact solution to Brownian Dynamics of a reversible bimolecular reaction in one dimension," 2016. arXiv: 1605.05557.

[46] ——, "Spatial stochastic intracellular kinetics: A review of modelling approaches," *Bull. Math. Bio.* 81: 2960–3009, 8 2018.

[47] J. Snoek, K. Swersky, R. Zemel, and R. Adams, "Input warping for Bayesian optimization of non-stationary functions," *Proceedings of the 31st International Conference on Machine Learning*, 2014, 1674–1682.

[48] M. Soltani, C. A. Vargas-Garcia, and A. Singh, "Conditional moment closure schemes for studying stochastic dynamics of genetic circuits," *IEEE Trans. Biomed. Circuits Syst.* 9(4): 518–526, 2015.

[49] A. Thibault, L. Chizat, C. Dossal, and N. Papadakis, "Overrelaxed Sinkhorn-Knopp algorithm for regularized optimal transport," 2017. arXiv: 1711.01851.

[50] M. Thomas and R. Schwartz, "A method for efficient Bayesian optimization of self-assembly systems from scattering data," *BMC Systems Biology*, 12(1): 65, 2018.

[51] D. Ulmasov, C. Baroukh, B. Chachuat, M. P. Deisenroth, and R. Misener, "Bayesian optimization with dimension scheduling: Application to biological systems," 2015. arXiv: 1511.05385.

[52] E. Vazquez and J. Bect, "Convergence properties of the Expected Improvement algorithm with fixed mean and covariance functions," *J. Stat. Plan. Inference*, 140(11): 3088–3095, 2010.

[53] C. Villani, *Optimal Transport: Old and New*, ser. Grundlehren der mathematischen Wissenschaften. Berlin Heidelberg: Springer, 2009.

[54] C. Zechner, J. Ruess, P. Krenn, *et al.*, "Moment-based inference predicts bimodality in transient gene expression," *PNAS*, 109(21): 8340–8345, 2012.

Appendix A

Appendix

A.1 The Reaction-Diffusion Master Equation

The Reaction-Diffusion Master Equation (RDME) is an extension of the CME that is more accurate in scenarios where the dynamics are affected by the limited diffusion speed of particles, which is frequently the case in the crowded intracellular environment. The RDME models the reaction network as a collection of interconnected systems described by the CME, usually called compartments or cells, such that particles are allowed to randomly move between adjacent compartments due to diffusion.

For our purposes the reaction volume is assumed to be cubic and partitioned into a symmetrical grid of N^d cubical cells, where d is the dimension of the reaction system, usually 2 or 3. We index grid cells by vectors $\vec{a} = (a_1, \ldots, a_d)$, where $a_i \in \{0, \ldots, N-1\}$ for all i. The state of the system is given by the number of particles of each species in each cell, that is, by an $\vec{a}$-indexed collection of tuples $\boldsymbol{n} = (n_1, \ldots, n_s)$. If the volume of the reaction container is denoted by V_T the volume of a single cell is given by $V_{\text{cell}} = V_T / N^d$.

The possible transitions in each state are reactions within a cell and particles moving between adjacent cells. The former are described by the renormalized propensity functions $\tilde{\rho}_i := \rho_i V_{\text{cell}}^{1-o_i}$, where o_i denotes the order of the i-th reaction and ρ_i denotes the propensity function in the CME. This renormalization accounts for the subdivision of the total reaction volume into several compartments modelled using the CME [46].

For a cubical grid the transition rates for particles jumping between adjacent cells have constant rates k_j, $j = 1, \ldots, s$, depending on the diffusion coefficients of the species and the mesh width of the grid. If one assumes that the particles follow Brownian motion one usually takes the jump rate for species j to be $k_j := \frac{1}{2}\frac{D_j}{h^2}$ for each pair

of adjacent cells, where h is the edge length of the voxels; in the limit as $h \to 0$ one then recovers the diffusion equation for the movement of particles.

We will use the standard basis vectors $\boldsymbol{e}_j = (0,\ldots,1,\ldots,0)$ for $j = 1,\ldots,j$, where the 1 is in the j-th position. Given a tuple $\boldsymbol{m} = (m_1,\ldots,m_s)$ and a cell $\vec{b}$ we define the vector $\left\{\boldsymbol{m}^{(\vec{b})}\right\}$ by

$$\boldsymbol{m}_{\vec{a}}^{(\vec{b})} = \begin{cases} \boldsymbol{m}, & \text{if } \vec{a} = \vec{b} \\ 0, & \text{else} \end{cases}$$

The RDME now reads:

$$\frac{\partial}{\partial t}P(\{\boldsymbol{n}_{\vec{a}}\},t) = \sum_{\vec{b}}\sum_{i=0}^{r}\left[\tilde{\rho}_i(\boldsymbol{n}_{\vec{b}} - \boldsymbol{S}_i)P\left(\{\boldsymbol{n}_{\vec{a}}\} - \boldsymbol{S}_i^{(\vec{b})},t\right) - \tilde{\rho}_i(\boldsymbol{n}_{\vec{b}})P(\{\boldsymbol{n}_{\vec{a}}\},t)\right] + \sum_{\substack{(\vec{b},\vec{c})\\ \text{adj.}}}\sum_{j=0}^{s} k_j\left[P\left(\{\boldsymbol{n}_{\vec{a}}\} - \boldsymbol{e}_j^{(\vec{b})} + \boldsymbol{e}_j^{(\vec{c})},t\right) - P(\{\boldsymbol{n}_{\vec{a}}\},t)\right]$$

Here the second sum is over all ordered pairs $(\vec{b},\vec{c})$ of neighbouring cells.

An important property of the RDME is the fact that while it is often more accurate than the CME, in the presence of bimolecular or higher-order reactions it breaks down as $h \to 0$. This is due to the fact that the RDME only considers reactions between particles in a cell; as $h \to 0$ the probability of finding two particles in any given cell approaches 0 and higher-order reactions become impossible in the limit. The RDME can thus only be thought of a mesoscopic model valid in a certain regime, namely when the mesh width h is small enough for diffusion to be fast in every compartment (so one can use the CME to model reactions within that compartment), but large compared to the reaction radii so that bimolecular reactions are not generally missed due to the reactant particles being in different cells.

A.2 Brownian dynamics

In Brownian dynamics a reaction system is modelled by a set of particles simultaneously undergoing Brownian diffusion in the reaction volume as well as (stochastic) chemical reactions. We will briefly present the basics below; a review of Brownian dynamics can be found in [46].

Molecules are modelled as point particles undergoing Brownian diffusion in a fixed reaction volume. As particles are diffusing they can undergo chemical reactions, and

the types of reactions commonly found in BD simulations are zero-molecular, unimolecular and bimolecular reactions. We will briefly describe the mechanisms for these three types below, with particular attention to the physical parameters involved in their description.

Zero-molecular reactions are represented by a Poisson point process which randomly adds new particles to the system at exponentially distributed intervals. The intensity of the point process is commonly taken to be uniform over the reaction volume, so the locus of every reaction is sampled uniformly in space. Under this assumption the probability of a reaction happening in the region X in the infinitesimal time interval $[t, t+\mathrm{d}t)$ is equal to $\lambda \,\mathrm{vol}(X)\mathrm{d}t$, for a scalar parameter $\lambda \in \mathbb{R}_{\geq 0}$ describing the rate of the reaction.

Unimolecular reactions simulate the spontaneous decay of particles; affected particles undergo a reaction after an exponentially distributed amount of time (unless they disappear in the context of a different reaction). Unimolecular reactions are parameterized by a scalar $\lambda \in \mathbb{R}_{\geq 0}$, the intensity of the waiting time distribution for each particle.

Bimolecular reactions can be described using either of two standard models used for BD. The *Smoluchowski model* specifies that two candidate molecules will react as soon as they come within a specified distance (the reaction radius $R > 0$) from each other, while under the *Doi* / λ-ρ model the candidate molecules react stochastically with rate $\lambda > 0$ as long as they are within a distance of ρ from each other, i.e. the reaction is modelled by an exponentially distributed waiting time that is frozen whenever the molecules are further than the distance ρ apart from each other.

We chose to focus on the Doi model for several reasons. First of all, obtaining realistic results with the Smoluchowski model in a chemical context often requires choosing the reaction radii to be much smaller than the size of the physical molecules involved, limiting the physical accuracy of this model. Second, the Doi model is more general than the Smoluchowski model, which can be recovered from the Doi model in the limit $\lambda \to \infty$. In addition the Doi model can be discretized to yield certain types of RDMEs under a suitable choice of parameters [19], making it more useful for model reduction and comparison in this case.

We remark that while more complicated reactions can be considered in the context of Brownian dynamics, true higher-order reactions are exceedingly rare in chemical applications and require their own reaction dynamics; we will not consider them as the dimerization model does not involve such reactions.

An important modelling choice is the behaviour of particles diffusing out of the reaction volume. We will focus on spatially periodic systems where particles leaving the reaction volume on one side will reenter on the other side; mathematically speaking the reaction volume can be described by an n-dimensional torus. An alternative would be to use reflective boundaries which redirect particles hitting the edge of the volume back into the system. The periodic model has the advantage of avoiding any kind of boundary effect and being slightly simpler to keep track of mathematically, and as such we will assume periodic boundary conditions for our model.

An outline of the BD algorithm used in this project is given below.

Algorithm 2 Simulation of Brownian dynamics

procedure UPDATE_POSITIONS(Δt)
  for all particles x_i **do**
    sample $\varepsilon_i \sim \mathcal{N}(0,1)$
    $x_i \leftarrow x_i + \sqrt{2(\Delta t)D_i}\,\varepsilon_i$

procedure UPDATE_REACTIONS(Δt)
  for all zero-molecular reactions R_i **do**
    sample $u \sim U(0,1)$
    $p \leftarrow \exp\left[-\lambda_i(\Delta t)\,\mathrm{vol}(\Omega)\right]$
    if $u > p$ **then**
      sample x uniformly in Ω
      add products **at** position x
  for all unimolecular reactions R_i **do**
    for all possible reactants x_j **do**
      sample $u \sim U(0,1)$
      $p \leftarrow \exp\left[-\lambda_i(\Delta t)\right]$
      if $u > p$ **then**
        remove particle x_j
        add products **at** position x_j
  for all bimolecular reactions R_i **do**
    for all pairs of possible reactants (x_j, y_j) **do**
      if $d(x_j, y_j) < \rho_i$ **then**
        sample $u \sim U(0,1)$
        $p \leftarrow \exp\left[-\lambda_i(\Delta t)\right]$
        if $u > p$ **then**
          remove particles x_j, y_j
          add products **at** midpoint of x_j, y_j

procedure MAIN_LOOP
  for $i = 1, \ldots, t_{\max}/\Delta t$ **do**
    UPDATE_POSITIONS(Δt)
    UPDATE_REACTIONS(Δt)