THE UNIVERSITY *of* EDINBURGH

Approximation Methods for Stochastic Systems Biology

Augustinas Sukys

THE UNIVERSITY
of EDINBURGH

Doctor of Philosophy

The University of Edinburgh

2024

Dedicated to my family

Abstract

Biochemical reactions involved in complex cellular mechanisms are driven by inherently stochastic molecular interactions. Although the intrinsic noise is often negligible in the macroscopic world, it has been established experimentally that intracellular processes can be subject to substantial stochasticity due to a low number of molecules present. Therefore, modelling the dynamics of such biological systems necessitates the use of stochastic rather than deterministic methods.

The Chemical Master Equation (CME) gives an accurate mathematical description of stochastic chemical reaction kinetics in well-mixed conditions. However, analytical solutions to the CME are available only for a handful of biologically relevant systems and its exact stochastic simulation with Monte Carlo methods can be prohibitively computationally expensive. This in turn motivates the development of approximation methods that provide more efficient ways of investigating the system behaviour. The study and the development of novel analytical and computational approximations to the CME is the focus of this thesis.

First, we develop an approximate time-dependent closed-form solution to the CME describing the Michaelis-Menten reaction mechanism of enzyme catalysis. The derivation is based on a time scale separation technique called averaging, allowing us to treat the Markovian dynamics on the slower time scale as a one-dimensional master equation that can be solved exactly in time using methods from linear algebra and complex analysis.

Second, we introduce MomentClosure.jl, a Julia package for automated derivation of the moment equations applicable to any biochemical system. As the moment expansion of the CME can lead to an infinite hierarchy of coupled moment equations, MomentClosure implements a wide array of moment closure methods that truncate the moment hierarchy and provide a closed set of equations describing approximate moment dynamics. The package integrates seamlessly with other Julia libraries and makes moment closure approximations more accessible to the broader scientific community.

Lastly, we propose a surrogate modelling framework that allows us to approximate the solution of the CME by training neural networks on stochastic simulation data. We showcase our approach on several models of gene expression, finding that relatively simple neural networks can learn to approximate highly complex distributions of molecule numbers over time and parameter space, and hence greatly accelerate otherwise computationally expensive parameter exploration and inference studies.

Lay Summary

In the intricate world of cells, all kinds of tiny molecules work together to produce energy, proteins and other essential materials needed for the cells to survive and function properly. These countless tasks and activities that take place inside living cells are known as cellular processes. They involve many interactions between the molecules which constantly move around, collide, combine, or change shape to carry out their assigned tasks. Sometimes, things might not go exactly as planned because of molecules randomly bumping into each other and lead to chemical reactions that might happen unpredictably, resulting in variations in how cellular processes occur.

Scientists are interested in this variability because understanding it helps to better grasp how cells work and how they respond to changes in their environment. To describe such unpredictable cellular processes, scientists need to model them using specialised methods from mathematics and statistics. However, there is a catch: getting exact answers with these methods is challenging for many important biological situations. Although it can be done using computer simulations, it can take a very long time and require a lot of computing power. Because of that, scientists are looking for shortcuts or clever methods that would be more efficient and still provide a good enough answer which gives insight into the functioning of cells. This thesis contributes to the development of such methods.

First, we studied how the complex molecules known as enzymes can speed up certain reactions in cells. We found a close-to-exact way of solving a specific reaction process called the Michaelis-Menten reaction mechanism, a fundamental model of enzyme activity. To achieve that, we used a technique called "averaging", which helps us to focus on the parts of the process that happen much more slowly and separately from the faster parts. By focusing on these slow parts, we can obtain a much simpler equation which can then be solved, allowing us to analyse the enzyme reaction in detail.

Second, we developed MomentClosure.jl, a software tool that can automatically write down a simplified description of any biological process using a specific toolset of mathematical methods. This simplified description, known as the moment equations, gives a statistical summary of how the molecules interact and change over time, focusing on the more general information about the system and ignoring the less important details. This software can be easily used by other scientists to study various models involving randomness.

Lastly, we introduced a novel technique of how neural networks, computer algorithms that are able to learn from examples to perform certain tasks, can be used to speed up the modelling of cellular processes. A neural network can be taught how molecules react in different cases from examples constructed using standard computer simulations. This way, the neural network builds up an understanding of the biological system and can predict how it will behave in unseen cases. We can use these networks to quickly explore various scenarios and make sense of the underlying molecular interactions without spending excessive time and resources on computations.

Acknowledgements

First and foremost, I would like to thank Ramon Grima for being a wonderful supervisor—I am deeply grateful for his unwavering support and guidance over the years, both personal and academic. I would also like to thank my co-supervisor Matthias Hennig and my thesis committee chair Matthew Nolan for their encouragement, feedback and assistance. In addition, I thank the Alan Turing Institute for funding my doctoral studies and providing continuous support.

I am grateful to all the past and present members of the Grima Group for creating such a friendly and motivating research environment. Namely, I would like to thank Zhixing Cao, Svitlana Braichenko, Xiaoming Fu, Tatiana Filatova, Lyndsay Kerr, Juraj Szavits-Nossan, Louis Headley, Rodrigo Garcia, Andrew Nicoll and Muhan Ma. I am especially thankful to James Holehouse and Kaan Öcal for the fun and fruitful collaborations as well as for the countless insightful discussions. I also thank my colleagues at the Alan Turing Institute whom I had the pleasure to meet on my trips to London, and in particular Sir Alan Wilson for his mentorship and advice.

Finally, I would like to express my gratitude to my parents, my brother and my late grandparents for their unconditional love and support throughout my life.

Declaration

I declare that the thesis has been composed by myself and that the work has not be submitted for any other degree or professional qualification. I confirm that the work submitted is my own, except where work which has formed part of jointly-authored publications has been included. My contribution and those of the other authors to this work have been explicitly indicated below. I confirm that appropriate credit has been given within this thesis where reference has been made to the work of others.

This thesis contains previously peer-reviewed research (* indicates equal contributions to the publication and † indicates the candidate's supervisor Ramon Grima):

[1] Holehouse, J.*, **Sukys, A.*** and Grima, R.†, 2020. Stochastic time-dependent enzyme kinetics: Closed-form solution and transient bimodality. The Journal of Chemical Physics, 153(16):164113. *Included in Chapter 3.*

[2] **Sukys, A.** and Grima, R.†, 2022. MomentClosure.jl: automated moment closure approximations in Julia. Bioinformatics, 38(1):289–290. *Included in Chapter 4.*

[3] **Sukys, A.***, Öcal, K.* and Grima, R.†, 2022. Approximating solutions of the Chemical Master equation using neural networks. iScience, 25(9):105010. *Included in Chapter 5.*

The writing and research conducted in the above publications was a collaborative effort between the candidate and the indicated co-authors, on each of which a major part of the writing and research was conducted by the candidate.

Augustinas Sukys
16th September 2024

Contents

Nomenclature

CDM	Conditional derivative matching
CME	Chemical Master Equation
FSP	Finite State Projection
LMA	Linear mapping approximation
MA	Moment closure approximation
MM	Michaelis-Menten
NN	Neural network
ODE	Ordinary differential equation
QEA	Quasi equilibrium approximation
QSSA	Quasi steady-state approximation
SSA	Stochastic simulation algorithm
SSE	System size expansion

Chapter 1

Introduction

1.1 Motivation and overview

Cellular processes are governed by complex networks of inherently stochastic molecular interactions [4]. The random nature of biochemical reactions and its effect on the macroscopic properties of a system can be disregarded in the limit of a large molecule number N because the relative fluctuations are typically of order $N^{-1/2}$ and hence become negligible—a rule of thumb rooted in statistical mechanics [5–7]. The behaviour of such biochemical systems is usually modelled with the reaction rate equations, a set of ordinary differential equations that describe the time evolution of the average concentrations of all chemical species in the system [8, 9].

However, a deterministic approach can be insufficient to accurately represent microscopic systems such as cells and their environment, considering that biological molecules within cells are usually present in low numbers [10]. For example, experimental studies [11–13] have revealed that in *E. coli* the copy numbers of each protein can vary from zero to ten thousand per cell, and the mRNA abundance ranges from zero to a hundred molecules. Coupled with the fact that there are usually only one or two gene copies per cell, these observations point out that random fluctuations can have a significant effect on cellular function. Therefore, stochastic modelling of chemical reaction networks has become a key element of systems biology [14, 15].

The importance of stochastic effects is particularly well established in gene expression, with many studies reporting a large amount of heterogeneity in mRNA and protein numbers in genetically identical cell populations [16–21]. A seminal contribution to the field was made by Elowitz et al. [16, 17], who demonstrated that both the probabilistic nature of the biochemical process of gene expression itself (intrinsic noise) and the influence of external fluctuations and other cell-specific factors (extrinsic noise) contribute substantially to the observed cell-to-cell variability in expression levels.

While most early stochastic gene expression experiments quantified the steady-state distributions of gene products (corresponding to a population snapshot measurement), a number of studies focused on visualising the production of mRNA molecules in real-time in individual cells [22–24]. Importantly, it was observed that genes tend to transiently switch between a relatively brief high transcriptional activity state and a longer-lived low activity state [25, 26]. Such transcriptional bursts can be modelled using the stochastic two-state (telegraph) model of gene expression [27], where a gene can randomly switch between an active and an inactive state, and the mRNA is produced with a certain probability only when the gene is active (the subsequent degradation of mRNA does not depend on the gene state). Although the telegraph model is phenomenological and greatly simplifies the complex chain of biochemical reactions underlying transcription, it could effectively capture many experimentally observed gene expression profiles [22, 23, 28, 29].

These findings, together with the advances in single-molecule imaging [30, 31] and single-cell sequencing [32, 33] technologies, have paved way for an increasingly large body of work focused on modelling stochastic gene expression [26, 34–53]. Some of these studies continue to use the two-state telegraph model to characterise the experimentally observed transcriptional bursting kinetics [26, 35–38], while others are focused on extending the telegraph model to incorporate the cell-cycle and cell-size effects or otherwise take into account the extrinsic noise [39–45] or consider more detailed models that capture genetic regulation [46–48], complex multi-state promoter switching mechanisms [49, 50] and polymerase dynamics [51–53].

Stochasticity is also important in other intracellular processes, for instance, enzyme catalysis. Enzymatic reactions are commonly characterised using the famous Michaelis-Menten reaction mechanism of enzyme action [54–56], where an enzyme can (reversibly) bind a substrate molecule to form an enzyme-substrate complex, which can then (irreversibly) yield a product molecule, leaving the enzyme free to catalyse another reaction. Using a similar argument as before, we expect to see significant noise in such enzymatic activity because single-cell proteomics show that, as for any other protein, enzyme abundance can span several orders of magnitude [11, 12]. Furthermore, modern single-molecule enzymatic turnover experiments report large fluctuations in the time-course measurements, which are otherwise masked in the conventional ensemble-averaged data [57–61]. As a consequence, theoretical research in enzyme kinetics has largely shifted to stochastic methods [62–70].

In addition, molecular noise can play a vital role in driving phenotypic diversification by inducing transitions between the metastable phenotypic states of a biochemical system [71–75]. This can be experimentally observed in the lactose uptake network (*lac* operon) in *E. coli* [76, 77]. In a deterministic framework the network is bistable (the

expression of lactose permease is either repressed or induced) due to the presence of a positive feedback loop in the system: as the extracellular concentration of an inducer is increased, the increased synthesis of the lactose permease facilitates further uptake of the inducer and hence the network remains in a high expression state, committing to lactose growth. However, at intermediate inducer concentration levels the noise-induced transitions between the two equilibria give rise to a bimodal distribution, where a fraction of the cell population is either in a low or high expression state. In fact, the switch in phenotype can be induced by a rare random event of the *lac* repressor completely disassociating from the promoter region, which can lead to a large burst of expression and trigger the positive feedback mechanism in the network [72, 78].

Furthermore, stochastic phenotype switching has been suggested as a possible explanation of bacterial persistence in *E. coli* [79, 80]: bacterial cells appear to spontaneously switch in and out of an antibiotic-tolerant state characterised by reduced growth, and the small bacterial subpopulation found in this state has a much higher chance to survive exposure to antibiotic treatment. Maintaining non-genetic population heterogeneity can be evolutionarily beneficial in facing unpredictable environmental changes [71, 81, 82], and similar bet-hedging strategies have been considered in a variety of biological contexts using a stochastic framework, including studies on sporulation decision in *B. subtilis* [83, 84], and drug resistance modelling in cancer cell populations [85, 86].

Deterministic models based on the reaction rate equations capture only the mean system behaviour and hence are fundamentally unable to describe the variability observed in gene expression or enzymatic turnover data. On the other hand, a stochastic model that accounts for intrinsic (and possibly extrinsic) noise can be readily simulated to generate stochastic realisations of a system that much better represent the experimental reality [14]. Therefore, theoretical understanding of the mechanisms governing cellular function cannot be complete without accounting for stochastic effects.

Although in a more practical setting one may be inclined to believe that population level measurements would average out the inherent fluctuations and hence stick to a deterministic modelling framework to explain the mean abundances of experimental observables, such approach may lead to misleading conclusions about the underlying kinetics. That is because a deterministic model and its stochastic equivalent may exhibit quantitatively as well as qualitatively different behaviour, which can become especially apparent in low molecule number regimes [87, 88].

It can be shown that for a chemical reaction network involving bimolecular or higher-order reactions the mean will depend on the size of intrinsic fluctuations and differ from the mean obtained using the deterministic reaction rate equations [15, 89, 90]. This theoretical discrepancy is well illustrated in [66] for the Michaelis-Menten reaction mechanism with substrate inflow, showing how the deterministic model can dramat-

ically underestimate the true mean substrate concentration. Such deviance in system behaviour can be expected as the steady-state distribution of the substrate is heavy-tailed, and the mean predicted by the rate equations is closer to the mode of the distribution rather than its average (see [91] for a more general discussion).

Another quantity that can be significantly affected by intrinsic noise is the mean first passage time, i.e., the average time a system takes to reach a given state. Such differences between stochastic and deterministic regimes are studied in detail by Ham et al. [92] for a number of simple biochemical systems. For instance, they find that the mean first passage time can be substantially reduced or lengthened in the telegraph model depending on the parameters controlling the gene state switching and the mRNA production. In addition, the authors demonstrate for the Michaelis-Menten enzyme kinetic model (with substrate production) that the expected waiting time is always reduced in the stochastic regime, and this discrepancy tends to increase with the decreasing system size as the intrinsic noise becomes more pronounced.

Qualitative changes in system dynamics due to stochastic effects are also observed in biochemical oscillators that underlie circadian rhythms or glycolytic oscillations. Namely, it has been shown both analytically and via stochastic simulations that noise can induce sustained oscillations in systems which approach the steady state via damped oscillations in the deterministic regime (systems with a stable focus) [93–97], and even in some models which do not exhibit deterministic damped oscillations (systems with a stable node) [98].

Noise-induced oscillations (quasi-cycles) can also serve as a potential explanation for experimentally observed single-cell power spectra. This is illustrated in [97], where the authors derive approximate analytical expressions for the power spectrum of noise-induced oscillations in a gene regulatory network motif and demonstrate that the theory accurately matches the protein luminescence data of a circadian rhythm in fibroblast cells [96]. While it can be difficult to distinguish whether such experimental data is best explained by a deterministically self-sustained or damped oscillator mechanism (as adding noise in both cases can result in sustained oscillations) [95, 97], such studies contribute to our theoretical understanding of how biological oscillators can exploit noise to their advantage in maintaining oscillations [93, 96] and synchronising circadian clocks [99].

Furthermore, stochastic effects can substantially alter the behaviour of multistable systems. A deterministically multistable system will evolve towards a stable fixed point determined by the initial conditions [100], but in a stochastic framework the random fluctuations can drive the system to reach either of its steady states [71, 74]. Although in the thermodynamic limit the steady states of a deterministic multistable model tend to match the modes of the stationary probability distribution of an equivalent

stochastic model, this correspondence can break down for low molecule numbers [90]. For instance, noise can lead to bimodal distributions not only in the parameter ranges with deterministic bistability, but also in some parameter regimes where only a single deterministic steady state is predicted [101, 102]. This is evident in the behaviour of the genetic toggle switch—a synthetic, bistable gene regulatory network comprised of two mutually repressive genes [103]. While the deterministic equations predict a bifurcation between monostability and bistability at a certain inductor concentration, the exact location of this transition becomes blurred due to random fluctuations in gene expression, leading to an experimentally observable bimodal population distribution in the vicinity of the bifurcation point [103, 104].

In addition, noise-induced bimodality (and multimodality) can emerge even in models that are deterministically monostable throughout their entire parameter space [102, 105–112]. Samoilov et al. [105] have analysed this in enzymatic futile cycles, a common component of various biochemical pathways: while the deterministic system can have only one steady state, a bimodal system response can be induced by the addition of external noise on the enzyme concentration. Moreover, under certain network conditions bimodality can be present in deterministically monostable autoregulatory genetic feedback loops [48, 107, 108, 112]. One experimental verification of such system behaviour is provided by To and Mahesri [107], who constructed a minimal positive feedback loop with multiple transcription factor binding sites in budding yeast, and demonstrated that highly bursty expression of short-lived transcription factors can lead to a bimodal population distribution without cooperative binding.

The discrepancies between deterministic and stochastic models in their quantitative predictions and qualitative system behaviour suggest that performing parameter inference and model selection from experimental biological data using a deterministic modelling framework can be unreliable and insufficient [14]. This is particularly apparent in single-cell data, often characterised by highly non-Gaussian mRNA or protein count distributions which can be highly skewed or bimodal [113]. As discussed earlier, the average abundances predicted by deterministic and stochastic models can be very different in such cases [34], and hence model calibration on the mean level can lead to large discrepancies in the inferred model parameters. Furthermore, estimating the unique set of parameters using the reaction rate equations is often impossible due to identifiability constraints, and that can be resolved by switching to a stochastic model that captures more information about the system dynamics [14, 114–117]. Nevertheless, statistical inference for stochastic models is a challenging problem—its results can strongly depend on the choice of inference method and the quality of data [14, 116–120], and eliminating any bias in parameter estimates requires a careful consideration of the extrinsic and technical sources of noise [121, 122].

The main stochastic modelling framework for chemical kinetics is provided by the Chemical Master Equation (CME) [123–125], otherwise known as the forward Kolmogorov equation [14]. More specifically, a chemical reaction network is treated as a continuous-time Markov chain where the state of the system is defined by the copy numbers of all chemical species in the network, and the CME describes the time evolution of the probability distribution over all possible states. The physical validity of the CME formalism can be rigorously established under well-mixed and dilute conditions [124, 126]. Moreover, it can be shown that the CME reduces to the deterministic reaction rate equations in the thermodynamic limit [127].

Solving the CME analytically can be a daunting task [125]. An exact time-dependent solution can be found for mono-molecular systems (comprised of reactions that involve at most one reactant and one product molecule) with arbitrary initial conditions [128] and for one-dimensional one-step systems [129]. Deriving the stationary (steady-state) distribution is usually significantly easier. It is possible to obtain the steady-state solution for systems that satisfy detailed balance [130], and, more generally, for weakly reversible chemical reaction networks with deficiency zero [131]. For most biological systems that do not fall into these subcategories time-dependent or steady-state solutions are not known in general. Despite that, an exact solution can be found for certain specific systems—some notable examples include [29, 43, 49, 132–136]. This is most often achieved by means of the generating function method that transforms the CME into a system of partial differential equations, which can then be tackled using a variety of mathematical techniques.

The problem becomes more analytically tractable if we do not require the whole probability distribution defined by the CME and are only interested in its first few statistical moments, such as means and (co)variances. In this case, a set of ordinary differential equations (ODEs) that describe the temporal evolution of the moments can be derived from the CME [125, 126]. If a given chemical reaction network contains only zero- and first-order reactions, the obtained moment equations will constitute a finite linear system of ODEs, which can be defined up to arbitrary order and solved exactly. However, if a system contains bimolecular or higher-order reactions, each moment equation will depend on the higher-order moments and hence lead to an infinite hierarchy of coupled ODEs that cannot be directly solved.

Solving the CME numerically is also challenging as it usually represents an infinite-dimensional system of ODEs. This issue can in principle be tackled by the Finite State Projection (FSP), a method that truncates the infinite state space of the system and makes it possible to find a numerical solution on a finite subset of states [137]. It also comes with an accuracy guarantee: the error on the computed solution can be systematically reduced by expanding the finite state space. A variety of algorithmic

improvements to the FSP have been devised, which enable a more efficient selection of the state space [138–141] or reduce the computational cost by using more sophisticated numerical approaches, for example, based on Krylov subspace methods [142, 143] or tensor decompositions [144–150]. However, the FSP remains limited in its practical applications to simple biochemical systems. Due to the combinatorial explosion of the size of the state space required for the FSP to obtain a reliable solution, it does not scale well to more complex networks and becomes prohibitively computationally expensive [125, 141, 149].

Alternatively, the stochastic process defined by the CME can be simulated using the Stochastic Simulation Algorithm (SSA), a Monte Carlo method popularised by Gillespie [151–153]. The SSA allows one to generate exact stochastic realisations of a system by simulating each and every reaction event explicitly in a chronological order. By repeatedly sampling the system using the SSA, the molecule number distribution and its moments can be accurately reconstructed. A variety of reformulations of the SSA can be found in the literature, which attempt to optimise the Monte Carlo step that underlies the simulation [127, 153–160], leading to a notable performance improvement for large biochemical networks involving many reactions. Despite that, the sequential simulation of each reaction event makes the SSA inevitably slow, especially for systems that contain large numbers of molecules or involve many fast reactions. Moreover, obtaining sufficiently accurate statistics of interest may require averaging over an extremely large number of SSA realisations.

The complexities of solving the CME have motivated the development of numerous approximation techniques which can considerably simplify the analysis of stochastic biochemical systems and lead to deeper mechanistic and biological understanding of their underlying dynamics [125, 161]. The vast literature on the approximations of the CME includes, but is not limited to: diffusion approximations such as the chemical Langevin equation [125, 130, 162–168]; the system size expansion [5, 88, 169] and in particular its leading-order truncation, the linear noise approximation [170–173]; moment closure approximations [126, 174–181]; linear mapping approximation [182]; model reduction techniques based on time scale separation [62, 125, 183–189]; accelerated simulation algorithms such as tau leaping [153, 190–192]; and hybrid modelling approaches [193–197].

Approximation schemes by definition trade some of the exactness of the CME for analytical tractability and computational efficiency, and therefore it is important to establish their accuracy and validity conditions [125, 165, 166, 198]. However, in many cases systematic analysis and comparison of different approximations and their limitations is challenging and remains an open problem, which in turn drives the search for novel and more versatile approaches. The study and the development of approximation methods for the CME is the central theme of this thesis.

First, we consider the stochastic Michaelis-Menten (MM) reaction mechanism of enzyme catalysis [54, 55]. This famous model has been studied extensively in the deterministic setting [54, 199, 200] and, as previously mentioned, its stochastic formulation has received increasing attention [62–70]. However, an exact transient solution to the CME of the MM reaction mechanism is only known for the case of a single enzyme [132]. We systematically derive an approximate closed-form time-dependent solution to the CME in the general case of multiple enzymes using a time scale separation technique known as averaging [189]. Averaging can be interpreted as the stochastic equivalent of the commonly used quasi-equilibrium approximation [185, 201]: we can identify the slow and fast reactions in the system and aggregate the rapidly equilibrating states together into groups that undergo effective Markovian dynamics on a slower time scale. This makes the problem much simpler as we can find the steady-state solution for the dynamics in each group and obtain the transient solution to the CME describing the transitions between the groups using an exact method proposed in [129]. We analyse the accuracy of the obtained closed-form solution over the parameter space and explore the temporal dynamics of the system, focusing in particular on the occurrence of transient bimodality. We also extend the approximation procedure to more complex multi-substrate mechanisms.

Second, we introduce MomentClosure.jl, a Julia package that provides automated generation of moment equations for any given chemical reaction network. As discussed before, the moment expansion of the CME generally leads to an infinite hierarchy of coupled equations that is not amenable to exact analytical treatment. A common approach in this situation is to truncate the moment hierarchy at a certain order by expressing the higher order moments as functions of the lower order moments, resulting in a finite set of approximate moment equations that are effectively *closed* and can be solved. This can be done using a myriad of different moment closure approximations (MAs) [125, 180, 181, 202]. However, the process of deriving the moment equations and applying the MAs manually can be laborious and prone to mistakes, limiting the use of this class of methods in practice. This lead us to develop MomentClosure, a tool to automatically derive and approximate the moment equations using a wide range of

MAs, which can then be solved using any high-performance ODE solver from DifferentialEquations.jl [203]. Overall, MomentClosure enhances the biochemical modelling capabilities of Julia [204, 205] and can facilitate novel applications, comparative studies and the development of MAs.

Lastly, we explore the potential applications of artificial neural networks for stochastic modelling of chemical kinetics. Neural networks have become highly successful in the past decade due to the advancements in efficient learning algorithms, growing computing power and data availability [206, 207]. Although they have been applied to a wide range of problems in the physical and biological sciences [208–213], relatively few studies have considered their use for CME-based models specifically [214–219]. We develop a novel framework to approximate the solutions of the CME using neural networks, which we name Neural Estimation of Stochastic Simulations for Inference and Exploration (Nessie). The main idea is to train simple neural networks on stochastic simulation data to predict the marginal distributions over time and parameter space for a given system, where we represent the target distributions as mixtures of negative binomials. We demonstrate on multiple biologically relevant models of gene expression that Nessie can learn to accurately capture complex system dynamics across parameter space: it becomes an effective surrogate model for the CME [220] and can be particularly useful in accelerating parameter exploration and inference studies.

1.2 Thesis structure

This thesis is organised as follows. In Chapter 2, we briefly review the CME formalism and the SSA, both of which are fundamental to the modelling of stochastic chemical kinetics and hence the research conducted in this thesis. We also discuss the FSP—an important numerical method that can be a viable alternative to the SSA.

In Chapter 3, we provide an approximate closed-form transient solution to the CME governing the dynamics of the Michaelis-Menten reaction system. We begin by summarising the main deterministic time scale separation approaches to enzyme kinetics. We then introduce our stochastic approximation procedure based on averaging: we use the method to solve the reaction system with a single enzyme and subsequently extend the solution to the general case of multiple enzymes. We use the obtained solution to investigate where the transient bimodality is manifest over the parameter space, and compare our approach to the commonly used stochastic Michaelis-Menten approximation. Finally, we demonstrate how our method can be applied to more complex reaction mechanisms such as ternary complex formation.

In Chapter 4, we present MomentClosure.jl, a software tool for automated derivation of moment equations and moment closure approximations. We provide a comprehensive review of the theory behind the method of moment expansion and a variety of moment closure methods implemented in the package. We then describe how MomentClosure is implemented and integrated with the broader Julia package ecosystem, and showcase its mains features by applying MomentClosure to an example model. Finally, we highlight some potential future extensions to the package, and touch upon the general drawbacks of moment closure methods.

In Chapter 5, we introduce the Nessie framework. We start with an overview of the basics of artificial neural networks and then describe the general workflow of using Nessie to approximate the marginal distributions over time and parameter space, discussing our technical implementation and the neural network training specifics. We then apply Nessie to several models of gene expression—an autoregulatory feedback loop, a genetic toggle switch, a model of mRNA turnover and a model of mitogen-activated protein kinase pathway—demonstrating Nessie's performance in capturing complex system dynamics and discussing its potential applications for parameter exploration and inference studies.

In Chapter 6, we conclude the thesis and discuss some of the possible avenues for future research.

Chapter 2

Preliminaries

In this chapter, we briefly outline the formalism of chemical reaction networks and the CME that is widely applied to model stochastic biochemical systems and used throughout this thesis. We also summarise the main theory behind the two important computer modelling tools: the Gillespie's Stochastic Simulation Algorithm (SSA) and the Finite State Projection (FSP). We refer to [125] for a more comprehensive treatment of the theory. Other more specific approximation methods and computational tools used in this thesis are discussed in their relevant context throughout the main text of the following chapters.

2.1 Chemical Master Equation

A chemical reaction network in a closed compartment with volume Ω consists of N different molecular species X_i $(i = 1, \ldots, N)$ and R chemical reactions of the form

$$\sum_{i=1}^{N} a_{ir} X_i \xrightarrow{k_r} \sum_{i=1}^{N} b_{ir} X_r, \quad \text{for } r \in \{1, \ldots, R\}, \tag{2.1}$$

where a_{ir} and b_{ir} respectively denote the integer numbers of reactant and product molecules of species i in the chemical reaction r with the associated reaction rate constant k_r. The state of the system is determined by the state vector $\boldsymbol{n} = (n_1, \ldots, n_N)$, where n_i is the number of molecules of species X_i present in the system.

Consider the probability $P(\boldsymbol{n}, t \,|\, \boldsymbol{n}_0, t_0)$ that the system is in state $\boldsymbol{n}$ at time t conditional on being in state $\boldsymbol{n}_0$ at time t_0. Note that throughout the text we use the shorthand notation $P(\boldsymbol{n}, t \,|\, \boldsymbol{n}_0, t_0) = P(\boldsymbol{n}, t)$. The time evolution of the probability distribution over the states $\boldsymbol{n}$ is then described by the CME [5, 124, 130]:

$$\frac{dP(\boldsymbol{n}, t)}{dt} = \sum_{r=1}^{R} \left[a_r(\boldsymbol{n} - \boldsymbol{S}_r) P(\boldsymbol{n} - \boldsymbol{S}_r, t) - a_r(\boldsymbol{n}) P(\boldsymbol{n}, t) \right], \tag{2.2}$$

where $a_r(\boldsymbol{n}, t)$ is the propensity function of the r^{th} reaction and $\boldsymbol{S}_r$ is the r^{th} column of the stoichiometric matrix $\boldsymbol{S}$, the elements of which are defined as $\boldsymbol{S}_{ir} = b_{ir} - a_{ir}$ (the net change in the number of molecules of species X_i when the r^{th} reaction occurs). The propensity $a_r(\boldsymbol{n})$ is the rate at which reaction r occurs when the system is in state $\boldsymbol{n}$; more formally, $a_r(\boldsymbol{n})dt$ is the probability that reaction r will take place in the infinitesimally short time interval $[t, t+dt)$ [127]. While the states have a lower bound of $n_i = 0$ as negative molecule numbers are unphysical, they have no upper limit (except when the networks dynamics are constrained by its underlying structure) and hence the CME generally forms an infinite system of coupled ordinary differential equations.

The CME can be derived from the Chapman-Kolmogorov equation [5, 130] and the propensity functions associated with different types of reactions can be formulated using phenomenological arguments based on chemical kinetics [14, 123]. Alternatively, a more fundamental derivation of the CME follows from a statistical mechanical argument comprehensively described by Gillespie [124]. Namely, a chemical system of reacting molecules in a compartment with volume Ω is treated as a dilute and well-mixed gas. This assumption implies that the total volume of all molecules is negligible compared to Ω, and elastic (non-reactive) collisions are much more frequent than reactive collisions, leading to a spatially homogeneous and thermally equilibrated system—the molecules are uniformly distributed throughout the compartment and have a Maxwell-Boltzmann velocity distribution. Under these assumptions the system becomes a continuous-time Markov chain with the state fully determined by the state vector $\boldsymbol{n}$, and the form of the propensities for unimolecular and bimolecular reactions can be obtained from the kinetic theory. The molecular physics-based derivation of the CME has also been extended to well-mixed dilute solutions that are more representative of a typical cellular system, and which reduces to the dilute gas treatment discussed above in the limit of fast diffusion (when diffusion is very fast compared to the rates of reactive collisions) [126, 221, 222].

It follows from combinatorial considerations and the arguments above that in the general form the mass action kinetics type propensity functions can be defined as [125]:

$$a_r(\boldsymbol{n}) = k_r \Omega \prod_{i=1}^{N} \frac{n_i!}{(n_i - a_{ir})! \Omega^{a_{ir}}}. \tag{2.3}$$

Moreover, one can define chemical reactions that do not follow mass action kinetics and have arbitrary associated propensity functions that are not necessarily polynomials in $\boldsymbol{n}$. Examples of such propensities include Michaelis-Menten kinetics or Hill kinetics, which arise as effective rate laws in reduced models under certain approximations [125, 223].

2.2 Stochastic Simulation Algorithm

The Stochastic Simulation Algorithm (SSA) is a Monte Carlo method which can be used to sample exact paths of the stochastic process defined by the CME [151, 152]. The SSA is routinely used for biochemical system simulation and serves as a benchmark for other simulation and approximation methods [125].

To generate stochastic trajectories of a system using the SSA, we consider the probability density function $p(\tau, r \mid \boldsymbol{n}, t)$: given the system is in state $\boldsymbol{n}$ at time t, $p(\tau, r \mid \boldsymbol{n}, t)d\tau$ defines the probability that the next reaction event will take place in the infinitesimal time interval $[t+\tau, t+\tau+d\tau)$ and it will be a reaction r. The derivation of an exact formula for $p(\tau, r \mid \boldsymbol{n}, t)d\tau$ is based on the fundamental premise that $a_r(\boldsymbol{n})dt$ is the probability that one reaction r will occur in the next differential time interval dt [151]. As discussed in Section 2.1, this formulation of the propensity function can be physically justified under certain assumptions. It can then be shown that $p(\tau, r \mid \boldsymbol{n}, t)$ is defined by [151]:

$$p(\tau, r \mid \boldsymbol{n}, t) = a_r(\boldsymbol{n}) \exp\left(-a_0(\boldsymbol{n})\tau\right),$$

where $a_0(\boldsymbol{n}) = \sum_{i=1}^{R} a_r(\boldsymbol{n})$ is the sum of all reaction propensities for state $\boldsymbol{n}$ and the term $\exp\left(-a_0(\boldsymbol{n})\tau\right)$ can be interpreted as the probability at time t that no reaction events will take place in the time interval $(t, t+\tau)$. By conditioning $p(\tau, r \mid \boldsymbol{n}, t)$, we can also express the following:

$$\begin{aligned} p(\tau \mid \boldsymbol{n}, t) &= a_0(\boldsymbol{n}) \exp\left(-a_0(\boldsymbol{n})\tau\right), \\ p(r \mid \boldsymbol{n}, t) &= \frac{a_r(\boldsymbol{n})}{a_0(\boldsymbol{n})}. \end{aligned}$$

One can draw samples from the probability distributions defined above using the inverse transform sampling [15, 151]:

$$\begin{aligned} &u_1, u_2 \sim U(0,1), \\ &\quad \tau = -\frac{\log(u_1)}{a_0(\boldsymbol{n})}, && (2.4) \\ &\quad r = \text{the smallest integer satisfying } \sum_{i=1}^{r} a_r(\boldsymbol{n}) > u_2 a_0(\boldsymbol{n}), && (2.5) \end{aligned}$$

where $U(0,1)$ denotes the uniform distribution on the interval $[0,1]$. This mathematical formulation underlies the Monte Carlo step of Gillespie's direct method [151]. Finally, the entire SSA algorithm can be summarised as follows:

0. Initialisation step. Set the time $t = t_0$ and the state of the system $\boldsymbol{n} = \boldsymbol{n_0}$. Specify the simulation end time t_{end}.

1. Compute the reaction propensities $a_1(\boldsymbol{n}), \ldots, a_R(\boldsymbol{n})$ and their total $a_0(\boldsymbol{n})$.
2. Draw two random numbers from the uniform distribution $U(0, 1)$.
3. Compute τ and r using Eqs. (2.4) and (2.5).
4. Execute the reaction by replacing $t \leftarrow t + \tau$ and $\boldsymbol{n} \leftarrow \boldsymbol{n} + \boldsymbol{S}_r$. Note that $\boldsymbol{S}_r$ is the r^{th} column of the stoichiometric matrix $\boldsymbol{S}$ defined in Section 2.1.
5. Record the simulation state $(\boldsymbol{n}, t)$. End the simulation if $t \geq t_{\text{end}}$; otherwise return to step 1.

A variety of exact SSA modifications have been proposed in the literature that try to refine the Monte Carlo step in the simulation to make it more computationally efficient, which can yield notable performance and scalability improvements, especially for large and complex chemical reaction networks [153–160]. However, the SSA is inherently slow due to the explicit simulation of each reaction event, and reconstructing a sufficiently precise probability distribution or other statistics using the SSA may require a very large number of repeated simulations, making it impractical [127, 137].

2.3 Finite State Projection

The Finite State Projection (FSP) method truncates the discrete state space of the system defined by the CME and hence can provide a direct numerical solution [137]. It is particularly effective for small scale chemical reaction networks and can be much more efficient than the SSA. In this section, we briefly outline the basic methodology behind the FSP.

We begin by reformulating the CME given by Eq. (2.2) in a more compact form. First, we select a particular (arbitrary) enumeration of the entire state space of the system and express it as a set $\boldsymbol{N} = \{\boldsymbol{n}_1, \boldsymbol{n}_2, \ldots, \}$. Note that $\boldsymbol{n}_i$ denotes a specific configuration of the state vector $\boldsymbol{n}$, and not the i^{th} element of $\boldsymbol{n}$. We can then define the complete probability state vector $\boldsymbol{P}(\boldsymbol{N}, t) = (p(\boldsymbol{n}_1, t), p(\boldsymbol{n}_2, t), \ldots)^\top$ and introduce the transition rate matrix $\boldsymbol{A}$ [137]:

$$\boldsymbol{A}_{ij} = \begin{cases} -\sum_{r=1}^{R} a_r(\boldsymbol{n}_i), & \text{for } i = j \\ a_r(\boldsymbol{n}_j), & \text{for all } j \text{ that satisfy } \boldsymbol{n}_i = \boldsymbol{n}_j + \boldsymbol{S}_r \\ 0, & \text{otherwise} \end{cases}. \tag{2.6}$$

This allows us to express the CME as

$$\frac{d}{dt}\boldsymbol{P}(\boldsymbol{N}, t) = \boldsymbol{A}\boldsymbol{P}(\boldsymbol{N}, t). \tag{2.7}$$

The formal solution to this equation with an initial condition $\boldsymbol{P}(\boldsymbol{N}, 0)$ is given by

$$\boldsymbol{P}(\boldsymbol{N}, t) = e^{t\boldsymbol{A}}\boldsymbol{P}(\boldsymbol{N}, 0). \tag{2.8}$$

For simple chemical reaction networks with a finite state space the matrix exponential in Eq. (2.8) can be readily computed (alternatively, Eq. (2.7) can be solved using standard ODE methods). However, most systems of interest have an infinite state space, in which case the solution cannot be found directly. The FSP method attempts to resolve this issue by truncating the state space. Namely, we choose a finite subset of the state space $\{\boldsymbol{n}_1, \ldots, \boldsymbol{n}_k\}$ of length k, and define $\boldsymbol{A}_k$ as the corresponding $k \times k$ submatrix of $\boldsymbol{A}$. Finding the matrix exponential of $\boldsymbol{A}_k$ then gives us an approximate solution: $\boldsymbol{P}(\boldsymbol{N}, t) \approx e^{t\boldsymbol{A}_k}\boldsymbol{P}(\boldsymbol{N}, 0)$. It can be shown that the probability "leakage" out of the truncated space is the error bound on the approximation

$$\epsilon = 1 - \mathbb{1}^\top e^{t\boldsymbol{A}_k}\boldsymbol{P}(\boldsymbol{N}, 0), \tag{2.9}$$

where $\mathbb{1}$ is a column vector of all ones so that $\mathbb{1}^\top e^{t\boldsymbol{A}_k}\boldsymbol{P}(\boldsymbol{N}, 0)$ is the sum over all states in the truncated domain. Moreover, ϵ decreases monotonically with the increasing state space size k, so the error on the numerical solution can be made arbitrarily small by systematically increasing the projection space [137]. The original FSP algorithm involves solving Eq. (2.7) for some finite domain of the system and tracking the approximation error ϵ: if it is lower than the chosen error tolerance, the algorithm can be terminated; otherwise, one can add more states to the projection and repeat the process until sufficient accuracy is reached.

The initial choice of the truncation can be made by a heuristic understanding of the underlying system dynamics so that it includes most states that notably contribute to the probability mass, and a number of strategies to optimally expand the state space have been proposed in the literature [138–141]. Furthermore, the computational cost of the FSP algorithm can be significantly reduced by utilising advanced numerical techniques based on Krylov subspace methods [142, 143] or low-rank tensor formats [144–150]. However, the FSP is limited in practice because it suffers from the curse of dimensionality: as the number of chemical species in the system is increased, the finite subspace required for an accurate solution can grow exponentially in size, making the algorithm computationally intractable.

Chapter 3

Stochastic time-dependent enzyme kinetics: closed-form solution and transient bimodality

This chapter has been published as [1] entitled *Stochastic time-dependent enzyme kinetics: closed-form solution and transient bimodality* in the *Journal of Chemical Physics*. Minor modifications have been made for its inclusion in this thesis.

3.1 Abstract

We derive an approximate closed-form solution to the chemical master equation describing the Michaelis-Menten reaction mechanism of enzyme action. In particular, assuming that the probability of a complex dissociating into enzyme and substrate is significantly larger than the probability of a product formation event, we obtain expressions for the time-dependent marginal probability distributions of the number of substrate and enzyme molecules. For delta function initial conditions, we show that the substrate distribution is either unimodal at all times or else becomes bimodal at intermediate times. This transient bimodality, which has no deterministic counterpart, manifests when the initial number of substrate molecules is much larger than the total number of enzyme molecules and if the frequency of enzyme-substrate binding events is large enough. Furthermore, we show that our closed-form solution is different from the solution of the chemical master equation reduced by means of the widely used discrete stochastic Michaelis-Menten approximation, where the propensity for substrate decay has a hyperbolic dependence on the number of substrate molecules. The differences arise because the latter does not take into account enzyme number fluctuations while our approach includes them. We confirm by means of stochastic simulation of all the elementary reaction steps in the Michaelis-Menten mechanism that our closed-form solution is accurate over a larger region of parameter space than that obtained using the discrete stochastic Michaelis-Menten approximation.

3.2 Introduction

The mechanistic basis of the simplest single-enzyme, single-substrate reaction consists of a reversible step between an enzyme and a substrate, yielding the enzymesubstrate complex, which subsequently forms the product. This reaction is commonly called the Michaelis-Menten (MM) reaction [54, 56].

For over a century, the dynamics of this reaction have been extensively studied using deterministic rate equations. Because these equations do not admit an exact closed-form solution, various approximations have been devised to obtain insight into the underlying dynamics. Use of the quasi-equilibrium or quasi steady-state approximations lead to the famous Michaelis-Menten equation, an ordinary differential equation relating the rate of product formation and the substrate concentration (see [55] for a discussion of these approximations and their range of validity). This equation provides a simple means to extract the relevant kinetic parameters (the Michaelis-Menten constant and the maximum velocity) from experimental data. The Michaelis-Menten equation has also been solved exactly leading to explicit expressions for the time-evolution of the substrate (and product) concentration [200].

The stochastic formulation of enzyme kinetics, while not as much studied as its deterministic counterpart, has received increasing attention since the 1960s when the chemical master equation (CME) for the MM reaction mechanism was first derived and studied by Anthony F. Bartholomay [224]. The CME is a probabilistic discrete description of chemical reaction kinetics that is valid in well-mixed environments for point reacting particles [127, 130]. Its relevance lies in its ability to describe kinetics when the molecule numbers are low, conditions typical in intracellular environments, e.g., the median copy number per cell of most enzymes in *E. coli* is below a thousand [11]. Research efforts concerning the MM mechanism in the area of stochastic chemical kinetics can be, broadly speaking, categorised into three types: (i) The search for a solution of the CME for the MM reaction and its various extensions, i.e., obtaining a closed-form solution for the time-dependent or steady-state probability distribution of the molecule numbers of each species in the reaction system [65, 132, 225]. (ii) The reduction of the CME and the construction of the stochastic equivalent of deterministic approximations (such as the fast equilibrium, quasi steady-state and total quasi steady-state approximations) and understanding their regime of validity [62, 64, 70, 125, 185, 186, 188, 226–232]. (iii) The derivation of exact or approximate expressions for the mean of the stochastic rate of product formation and an investigation of the differences or similarities from the predictions of the deterministic Michaelis-Menten equation [61, 63, 69, 233–236]. The majority of the literature has focused on (ii) and (iii). There are very few studies that focus on (i) principally because the CME is notoriously difficult to solve analytically [125]. In this chapter, we are interested in deriving new solutions of the CME for

enzyme kinetic systems and hence next we briefly review the known solutions (see also [66] for a lengthier discussion). Arányi and Tóth [132] were the first to exactly solve the CME introduced by Bartholomay for the special case in which there is only one enzyme molecule with several substrate molecules in a closed compartment; in particular, they obtained an exact expression for the joint distribution of the number of substrate and enzyme molecules as a function of time (since the original paper is rather difficult to find, in Appendix A.1 we have reproduced the derivation in a concise manner). Another exact solution is reported in [225] by Schnoerr *et al.* who derive the exact steady-state solution for the CME describing the MM reaction system with one enzyme molecule and augmented with a substrate production reaction step (to model for example the production of substrate via translation). To our knowledge, there are no known exact solutions for the time-dependent probability distribution of the CME of the MM reaction system with multiple enzyme molecules. However, an approximate closed-form solution was derived by Dóka and Lente [65], using a so-called stochastic equivalent of the quasi steady-state approximation. Namely, they make an ansatz that the joint distribution of the number of substrate and enzyme molecules takes the form of a product of a time-dependent function and a constant value which characterises the state occupied by the system. Using this assumption and a number of heuristic arguments, the authors reduce the problem to a one-variable master equation which they then solve iteratively. However, one could argue that the derivation outlined in [65] lacks a certain degree of rigour and the analysis of the accuracy of the solution over time and parameter space is rather limited, which raises questions about the validity of the approximation.

In this chapter, our aims are to: (a) Derive an expression for the approximate time-dependent solution of the CME of the MM reaction system with multiple enzyme molecules under quasi-equilibrium conditions using an approach that is more rigorous and systematic than in previously published works. (b) Compare and contrast this solution with the solution of an often used reduced CME for the MM reaction in the literature. (c) Use the closed-form solution to identify interesting dynamical phenomena. We verify our approximate analytic results against the benchmark stochastic simulation algorithm (SSA) [127]. This chapter is divided as follows. In Section 3.3, we briefly review the main results known for deterministic enzyme kinetics, focusing in particular on the quasi-equilibrium approximation. In Sections 3.4.1 and 3.4.2, we introduce our method by first applying it to the MM reaction with a single enzyme molecule and subsequently to the case of multiple enzyme molecules. The method consists of three steps: (1) using a time scale separation method called averaging [189] to define groups of rapidly equilibrating states which then allows the derivation of a master equation describing the Markovian dynamics of these groups on the slower time scale; (2) solving the resultant time-dependent, single variable master equation for the group

dynamics using the method developed in [129] which has the advantage of bypassing the calculation of the eigenvectors of the transition matrix and hence considerably simplifies the analytical computations; (3) using the time-dependent solution describing the group dynamics to construct the marginal time-dependent distributions for both the numbers of substrate and enzyme molecules. We use the closed-form solution to find the regions of parameter space where transient bimodality of the distribution of substrate molecules occur. In Section 3.5, we show that our solution is accurate over a wider region of parameter space than the solution of a commonly used reduced master equation with a propensity that has the same hyperbolic dependence on the number of substrate molecules as the deterministic Michaelis-Menten equation (an approach popularised by Rao and Arkin [62]). In Section 3.6, we show that the same three-step method used in Sections 3.4.1 and 3.4.2, can be used to derive time-dependent distributions for multi-substrate enzyme reactions. We finish by discussing our results in Section 3.7.

3.3 Deterministic enzyme kinetics

Before progressing to stochastic enzyme kinetics we first briefly outline some of the main results known for deterministic enzyme kinetics. We consider the chemical reaction system:

$$S + E \underset{k_1}{\overset{k_0}{\rightleftharpoons}} C \xrightarrow{k_2} E + P, \tag{3.1}$$

where S denotes the substrate species, E denotes the enzyme species, C denotes the enzyme-substrate complex and P denotes the product. This system can be thought of as a reduction of the more biologically realistic set of reactions:

$$S + E \rightleftharpoons ES \rightarrow EP \rightarrow E + P, \tag{3.2}$$

where the unbinding of the product from the enzyme is very fast. For simplicity, we assume the initial condition for this system is that all enzymes are unbound to the substrate. There are two conservation laws for this system: $[E] + [C] = [E]_0$ and $[S] + [C] + [P] = [S]_0$, where $[i]$ denotes the concentration of species i and $[i]_0$ the initial concentration of species i. Assuming well-mixed conditions and the law of mass action, the deterministic dynamics of the reaction system in Eq. (3.1) are described by a set of coupled ordinary differential equations (commonly called the rate equations) describing

the time-evolution of the substrate and complex concentrations:

$$\begin{aligned}\frac{d[S(t)]}{dt} &= -k_0[S(t)]([E]_0 - [C(t)]) + k_1[C(t)], \\ \frac{d[C(t)]}{dt} &= -(k_1 + k_2)[C(t)] + k_0[S(t)]([E]_0 - [C(t)]).\end{aligned} \tag{3.3}$$

Note that the time-dependent concentrations of E and P can be straightforwardly obtained from the time-dependent solutions of C and S by means of the conservation laws previously stated. Although seemingly simple, the rate equations given by Eq. (3.3) are not easy to solve analytically for the time-dependent analytic solution, and as such one is limited to finding approximate solutions. Two of the most common approximations used in the literature are the (i) *quasi steady-state assumption* (QSSA) and (ii) the *quasi-equilibrium approximation* (QEA), also called the rapid equilibrium approximation or the reverse quasi steady-state assumption. The QSSA, derived by Briggs and Haldane [199], assumes that after a short transient, the concentration of the complex (and enzyme) is in a quasi steady-state (with regard to the substrate and product); thus under the QSSA, it is assumed that $d[C(t)]/dt \approx 0$. See [237] for a detailed discussion of this approximation and for its range of validity. On the other hand, the QEA assumes that substrate binding and dissociation occur much more rapidly than product formation such that the substrate, enzyme and complex are approximately in equilibrium. Thus under the QEA, it is assumed that $d[S(t)]/dt \approx 0$; this approximation, popularised by Michaelis and Menten [54], is commonly used in the analysis of various biochemical models [201].

Enforcing either the QSSA or QEA leads to the following effective rate equation describing the time-evolution of the substrate concentration:

$$\frac{d[S(t)]}{dt} = \frac{-V_{\max}[S(t)]}{k + [S(t)]}, \tag{3.4}$$

where $V_{\max} = k_2[E]_0$, $k = (k_1 + k_2)/k_0$ if the QSSA is used, $k = k_1/k_0$ if the QEA is used, and where the conservation law $[S] + [P] = [S]_0$ holds. Note that a necessary limitation of Eq. (3.4) is that we have assumed that $[S] + [C] \approx [S]$, which is true in the limit $[S]_0/[E]_0 \gg 1$. Eq. (3.4) has been solved perturbatively in a number of studies, all of which also assessed the validity of the QSSA [237, 238]. An exact solution was reported in [200] which is given by:

$$\langle n(t)\rangle_a = \Omega[S(t)] = \Omega k\ W\left(\frac{[S]_0}{k}\exp\left(\frac{-V_{\max}t + [S]_0}{k}\right)\right), \tag{3.5}$$

where $\langle n(t)\rangle_a$ gives the (deterministic) *number* of bound and unbound substrate molecules obtained in the limit $[S]_0/[E]_0 \gg 1$ at time t, Ω is the volume of the system, and $W(\cdot)$ is the principal branch of the Lambert W function (also known as the Omega function). Note that within van Kampen's *system size expansion* [5] for monostable systems, the rate equations are obtained as the macroscopic limit of the stochastic description of a well mixed chemical system; within this formalism, the concentration of a species i multiplied by the volume is the same as the mean number of molecules of species i. Hence in our case $\langle n(t)\rangle_a$ can also be interpreted as the *mean* number of substrate molecules in the macroscopic limit. In the rest of this chapter, we study the stochastic equivalent of the QEA and thus we shall use $k = k_1/k_0$.

3.4 Stochastic QEA analysis

3.4.1 Single enzyme

For simplicity, we first illustrate the method by solving the enzyme system described in Eq. (3.1) for the case of one enzyme molecule with initially N substrate molecules. Since there are no birth-death processes coupled to any species, the conservation equations $n_E + n_C = 1$ and $n + n_C + n_P = N$ hold, where n denotes the number of substrate molecules and all other n_i denote the number of species i.

We label the microstate of the reaction network in Eq. (3.1) as (n, n_E), which fully specifies the state of the system due to the conservation laws stated previously. The possible transitions between all of the discrete microstates of this system are illustrated in Fig. 3.1(i): the system starts from the state $(N, 1)$ and eventually ends up in the state $(0, 1)$. Our goal now will be to find the marginal probability distribution $P(n; t)$, i.e., the probability of observing n substrate molecules at a time t.

Assuming Markovian dynamics [125], it follows that the time-evolution of $P(n, n_E; t)$ (the probability of observing n substrate molecules and n_E enzyme molecules at a time t) is given by the CME:

$$\begin{aligned}\frac{\partial P(n, n_E; t)}{\partial t} = \; & k_0(n+1)(n_E+1)P(n+1, n_E+1; t) \\ & + (2 - n_E)\left(k_1 P(n-1, n_E-1; t) + k_2 P(n, n_E-1; t)\right) \\ & - \left(k_0\, n\, n_E + (1-n_E)(k_1+k_2)\right) P(n, n_E; t).\end{aligned} \tag{3.6}$$

Note that this CME is valid only for a single enzyme system, i.e., $n_E \in \{0, 1\}$. Furthermore note that the bimolecular propensity is inversely proportional to the volume Ω but for simplicity we set $\Omega = 1$ (a convention throughout the chapter). The standard approach involves introducing the time-dependent marginal generating functions

$G_{n_E}(z;t) = \sum_n z^n P(n, n_E; t)$ and attempting to solve the generating function partial differential equations, e.g., using eigenfunction methods [5, 130]. However, this standard method quickly leads one to mathematical difficulty. An analytic solution only presents itself in a non-cumbersome form where one assumes the initial state contains a single substrate molecule [132]. In Appendix A.1 we summarise the single enzyme solution provided by [132], and its complexity even in the single substrate molecule case motivates the analysis we present below.

We take a different approach. We first simplify the problem through the use of averaging [189, 239, 240]. Specifically the procedure lumps together microstates equilibrating on a fast timescale in groups which then allows one to write a master equation describing the dynamics of the groups on the slow timescale. We shall assume that the slow timescale is that associated with product formation, i.e., k_2 is sufficiently small (we will be more precise what this really means later) and hence the averaging procedure is in the same spirit as the QEA discussed in Section 3.3.

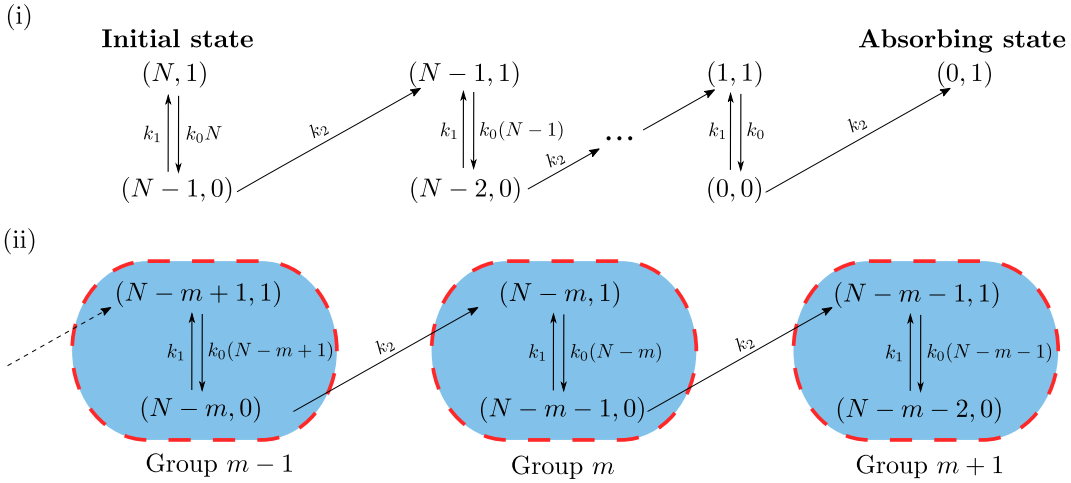

Figure 3.1: Illustration of the enzymatic system described by a single enzyme and N initial substrate molecules. (i) Markovian dynamics of the enzyme kinetic system described by a single enzyme. The initial condition for the system is $(N, 1)$, and as $t \to \infty$ the microstate of the system is guaranteed to be that of the absorbing state $(0, 1)$, with no remaining substrate and one free enzyme. (ii) Markovian dynamics in the reduced model, where processes occurring in a group are assumed to be much faster than the interactions between the groups themselves. The label 'group m' denotes the set of microstates that exist when m product molecules have been formed, given that there are no product molecules initially; hence, it is easily seen that there are $N + 1$ groups in total with labels $m = \{0, 1, 2, ..., N-1, N\}$.

Since k_2 is small, it follows that we can group all microstates that are in rapid equilibrium with each other (due to the fast processes of binding and unbinding of substrate from the enzyme) as shown in Fig. 3.1(ii); group m is then the set of microstates of the system accessible when m product molecules have been produced. We define $p^g_m(t)$ as the probability to be in group m at a time t, and $p^{qe}_{i,m}$ as the probability of having i free enzymes for the *reduced* system given by considering only reactions among microstates in group m. Once these probabilities are found, we can construct $P(n;t)$, based on the

fact that there are two microstates that contain n substrate molecules: $(n, 0)$ and $(n, 1)$ associated with groups $N-(n+1)$ and $N-n$ respectively. This means that under the stochastic QEA:

$$P(n;t) = p^g_{N-n}(t)p^{qe}_{1,N-n} + p^g_{N-(n+1)}(t)p^{qe}_{0,N-(n+1)}. \tag{3.7}$$

In the case of the single enzyme system studied in this section, the quasi-equilibrium probabilities are trivial (since there are only two microstates in each group) and are given by:

$$p^{qe}_{1,N-n} = \frac{k_1}{k_1 + k_0 n} \quad \text{and} \quad p^{qe}_{0,N-(n+1)} = \frac{k_0(n+1)}{k_1 + k_0(n+1)}. \tag{3.8}$$

All that remains is the task of finding $p^g_m(t)$. To do this we first write the master equation for the transitions between groups. Rescaling time as $t' = k_2 t$ and making use of the previous definition, $k = k_1/k_0$, the master equation for the groups is:

$$\partial_{t'} p^g_m(t') = a_m p^g_{m-1}(t') - a_{m+1} p^g_m(t'), \tag{3.9}$$

where:

$$a_m = \frac{N-(m-1)}{k+N-(m-1)},\ 1 \le m \le N+1, \tag{3.10}$$

and $a_{i\le 0} = 0$. Note that a_m is the probability of the jump from group $m-1$ to group m in a unit interval of rescaled time. From Fig. 3.1 the probability of the jump from group $m-1$ to group m in a unit interval of normal time is equal to k_2 multiplied by the probability of being in the microstate $(N-m, 0)$ which under the rapid equilibrium assumption is $k_0(N-m+1)/(k_1 + k_0(N-(m-1)))$. Due to time rescaling, the factor of k_2 disappears and hence follows Eq. ((3.10)).

Since there are $N+1$ groups in total, Eq. (3.9) corresponds to a system of $N+1$ ODEs which can be concisely written as the matrix equation:

$$\partial_{t'} \underline{p}^g(t') = \mathcal{Q} \cdot \underline{p}^g(t'), \tag{3.11}$$

where $\underline{p}^g(t')$ is a $N+1$ element column vector defined by $\underline{p}^g(t') = (p_0^g(t'), p_1^g(t'), ..., p_N^g(t'))$ and $\mathcal{Q}$ is a $(N+1) \times (N+1)$ lower-bidiagonal square matrix defined by:

$$\mathcal{Q} = \begin{pmatrix} -a_1 & & & & \\ a_1 & -a_2 & & & \\ & a_2 & -a_3 & & \\ & & \ddots & \ddots & \\ & & & a_N & -a_{N+1} \end{pmatrix}. \tag{3.12}$$

As we will describe below, we solve the set of ODEs given by Eq. (3.11) using the method described in [129] which provides an *exact* time-dependent solution for any one-variable one-step master equation with finite number of microstates *as long as one can find the eigenvalues of the transition rate matrix exactly.* In our case, the eigenvalues of $\mathcal{Q}$ are trivial, since $\mathcal{Q}$ is lower-bidiagonal, and they are given by the diagonal elements. Hence the eigenvalues of $\mathcal{Q}$ are given by $\lambda_i = -a_i$, $1 \leq i \leq N+1$. Note that $\lambda_{N+1} = 0$ and is the largest eigenvalue, with all $\lambda_{1 \leq i \leq N} < 0$.

We now proceed to use these eigenvalues to find the time-dependent solution to Eq. (3.11). The solution to this set of ODEs is formally given by:

$$\underline{p}^g(t') = \exp\left(\mathcal{Q}t'\right) \cdot \underline{p}^g(0), \tag{3.13}$$

where $\exp(\mathcal{Q}t')$ is defined as a matrix exponential. For a general master equation, this matrix exponential is typically difficult to deal with, however in our case $\mathcal{Q}$ is lower-bidiagonal and hence we can proceed via the method of [129]. We first consider Cauchy's integral formula for matrices, explicitly given by [241]:

$$\underline{f}(\mathcal{Q}) = \frac{1}{2\pi i} \oint_C (zI - \mathcal{Q})^{-1} \cdot \underline{f}(z) dz, \tag{3.14}$$

where C is a closed contour in the complex plane that encloses all the eigenvalues of $\mathcal{Q}$ and I is the identity matrix. Taking $\underline{f}(z) = e^{zt'}\underline{p}^g(0)$ we then arrive at:

$$\underline{p}^g(t') = \frac{1}{2\pi i} \oint_C (zI - \mathcal{Q})^{-1} \cdot \underline{p}^g(0) e^{zt'} dz. \tag{3.15}$$

A typical initial condition is $p_m^g(0) = \delta_{m,0}$, meaning that we always start in group 0 which contains the microstates $(N, 1)$ and $(N-1, 0)$, as is shown in Fig. 3.1(i). Note that $\delta_{i,j}$ is the Kronecker delta. Using this initial condition, Eq. (3.15) becomes:

$$p_m^g(t') = \frac{1}{2\pi i} \oint_C (zI - \mathcal{Q})^{-1}_{m+1,1} e^{zt'} dz. \tag{3.16}$$

We show at the end of this section how to extend the time-dependent solution for a general initial distribution. Since it is bidiagonal, the inverse of $zI - \mathcal{Q}$ can easily be found via Cramer's rule [242]:

$$(zI - \mathcal{Q})^{-1}_{ij} = \begin{cases} 0, & i < j, \\ \frac{1}{a_i + z}, & i = j, \\ \frac{1}{a_j + z} \prod_{k=j+1}^{i} \frac{a_{k-1}}{a_k + z}, & i > j. \end{cases} \tag{3.17}$$

Substituting this into Eq. (3.16) then gives us:

$$p_m^g(t') = \begin{cases} 0, & m < 0, \\ \frac{1}{2\pi i} \oint_C \frac{e^{zt'}}{z - \lambda_1} dz, & m = 0, \\ \frac{1}{2\pi i} \left\{(-1)^m \prod_{k=1}^{m} \lambda_k\right\} \times \left\{ \oint_C \frac{e^{zt'}}{\prod_{k=1}^{m+1} (z - \lambda_k)} dz \right\}, & m > 0, \end{cases} \tag{3.18}$$

where we have utilised the relation $\lambda_i = -a_i$. These integrals can then be evaluated using Cauchy's residue theorem [243], explicitly stated as:

$$\oint_C f(z) dz = 2\pi i \sum_k \mathrm{Res}(f(z), z_k), \tag{3.19}$$

where the values $z = z_k$ are poles of $f(z)$ within C and the residues are $\mathrm{Res}(f(z), z_k) = \lim_{z \to z_k} (z - z_k) f(z)$ for the simple poles in Eq. (3.18). Note that the poles of the complex integrals in Eq. (3.18) are the eigenvalues of $\mathcal{Q}$. Therefore, from Eq. (3.18) we finally get an expression for $p_m^g(t')$ as:

$$p_m^g(t') = \begin{cases} 0, & m < 0, \\ e^{\lambda_1 t'}, & m = 0, \\ \left\{(-1)^m \prod_{k=1}^{m} \lambda_k\right\} \times \left\{ \sum_{k=1}^{m+1} \frac{e^{\lambda_k t'}}{\prod_{j=1, j \neq k}^{m+1} (\lambda_k - \lambda_j)} \right\}, & m > 0. \end{cases} \tag{3.20}$$

Hence the time-dependent probability distribution $P(n; t)$ is given by Eq. (3.7) together with Eqs. (3.8) and (3.20). The extension to a more general initial distribution is then relatively simple. Consider some initial distribution $\underline{p}^g(0) = \underline{q}$, where $\underline{q}$ is an $N + 1$ element vector; the time-dependent group probability $p^g_{m|\underline{q}}(t')$ is then given by the

weighted sum:

$$p^g_{m|\underline{q}}(t') = \sum_{j=0}^{N} p_{m|q_j}(t') q_j. \tag{3.21}$$

This initial condition could be useful to model variation in the initial number of substrate molecules due to uncertainty introduced by experimental error or else due to the intrinsic noise in the reaction mechanism generating the substrate. Note that if $q_m = \delta_{m,0}$, one clearly recovers the analysis shown above. For the rest of this chapter we only consider the initial condition $p^g_m(0) = \delta_{m,0}$, *specifically where all enzymes are initially unbound to the substrate and where there are initially zero product molecules*, but note that the analysis that follows can be easily extended for more general initial distributions.

In the beginning of this derivation, we stated that the main assumption is that k_2 is sufficiently small. This statement can be made more precise as follows. From Fig. 3.1(ii) it is clear that the exit from group m can only occur when the enzyme is bound to substrate, i.e., from state $(N-m-1, 0)$. Now given that we are in this state, it follows that only two reactions can occur: either a reaction which causes a group change, i.e., $(N-m-1, 0) \to (N-m-1, 1)$ which occurs with rate k_2 or a reaction that leads to no group change, i.e., $(N-m-1, 0) \to (N-m, 1)$ which occurs with rate k_1. Hence the probability of leaving the group is $k_2/(k_1+k_2)$, from which follows that the microstates in each group will achieve quasi-equilibrium if $k_2 \ll k_1$. Therefore, this is the condition under which our method provides a good approximation to the distribution of substrate molecules at all times.

We test the distributions predicted by Eq. (3.7) against the SSA in Fig. 3.2A(i-iii) and Fig. 3.2B(i-iii). In Fig. 3.2A(i-iii) we show that the solution is accurate for small $N = 8$, over a time range from $t' = 1$ near the initial condition, to $t' = 12$ close to the absorbing state, where the validity criterion $k_1 \gg k_2$ holds. In Fig. 3.2B(i-iii) we observe that our solution agrees similarly well to the SSA for larger values of N. For a more general comparison of the exact solution to SSA through time, we can compute the mean and standard deviation from Eq. (3.7):

$$\langle n(t') \rangle = \sum_{n=0}^{N} n P(n; t'), \tag{3.22}$$

$$\sigma(t') = \sqrt{\left(\sum_{n=0}^{N} n^2 P(n; t') \right) - \langle n(t') \rangle^2}. \tag{3.23}$$

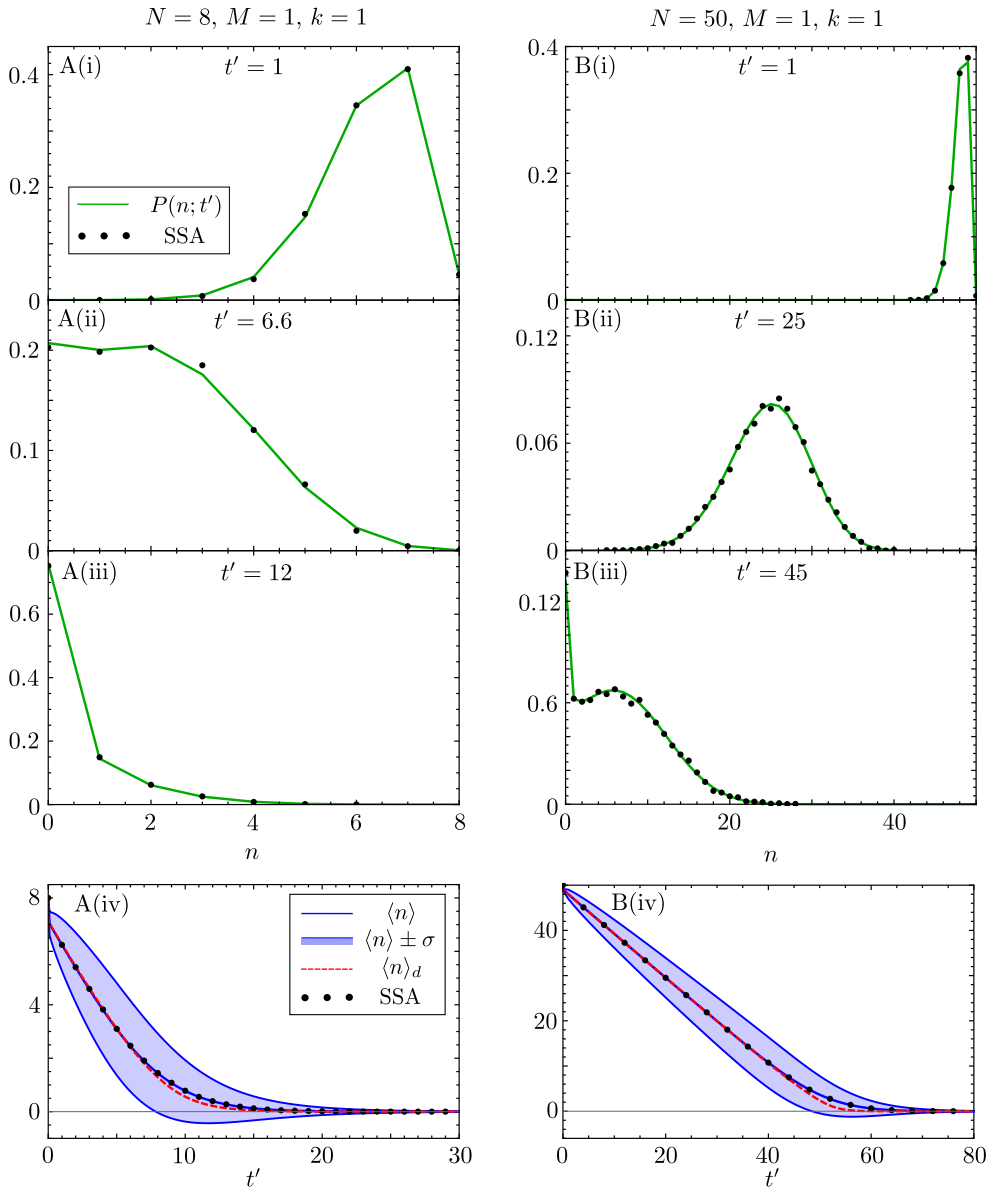

Figure 3.2: Comparison of the analytic time-dependent probability distribution of substrate molecules for the enzyme reaction in (3.1) with one enzyme molecule, i.e., $M = 1$, and N initial substrate molecules to the distribution obtained from the SSA [127]. Note that the analytic solution is given by Eq. (3.7) together with Eqs. (3.8) and (3.20). In all cases we enforce $k_1/k_2 \gg 1$ such that the quasi-equilibrium assumption behind the QEA is justified. We show the time-evolution of the distribution for substrate numbers, from near the initial condition to near the absorbing state, in two cases: A(i-iii) is for $N = 8$, $k_0 = k_1 = 10^3, k_2 = 1$, meaning that $k = k_1/k_0 = 1$. B(i-iii) is for $N = 50$, and all rate parameters as in the previous case. Note that the analytical solution (green lines) matches the SSA (black dots) for all times, for both a small and large initial number of substrate molecules. In A(iv) and B(iv) we show the corresponding plots of the time-evolution of the mean $\langle n \rangle$ and of the standard deviation σ of the distributions of substrate molecules, as predicted by our theory; these are compared with the mean calculated from the SSA and the mean $\langle n \rangle_d$ obtained from the numerical solution of the deterministic rate equations given by Eq. (3.3). Note that the deterministic mean is a better approximation to the stochastic mean for larger N. As shown in B(iii), and mildly in A(ii), the distribution can be bimodal at intermediate times. Each SSA probability distribution is constructed from 10^5 individual reaction trajectories.

The stochastic mean number of substrate $\langle n \rangle$ can then be compared to the deterministic mean number $\langle n \rangle_d$ obtained from the rate equations. That is, we numerically solve Eq. (3.3) for $[S(t)]$ with $k_2 = 1$, noting that $\langle n \rangle_d = [S(t)]$ as we have previously set $\Omega = 1$.

In Figs. 3.2A(iv) and 3.2B(iv) we plot the evolution of the stochastic and deterministic mean substrate numbers in time, and compare them to the SSA for parameters sets $N = 8$, $k = 1$ and $N = 50$, $k = 1$ respectively. We also show the standard deviation about the mean, i.e., $\langle n \rangle \pm \sigma$, where we have dropped the time-dependence for brevity, given in the blue envelope. Clearly, $\langle n \rangle$ from Eq. (3.22) matches the mean predicted by the SSA for most times, aside from the initial condition at $t' = 0$, where a step-like drop is observed in the mean predicted by the SSA to the value predicted by the quasi-equilibrium analysis. This step-like drop to the quasi-equilibrium value of the mean is known as the *initial transient*, and is seen in more detail in Appendix A.2. The explanation of the initial transient follows by considering the system after quasi-equilibrium has been reached between the two microstates in group 0, $(N, 1)$ and $(N-1, 0)$, after a time $t'_c \approx 1/\min\{k_0 N, k_1\}$ which is small under the rapid equilibrium assumption. Because of the discreteness of the substrate molecules, $\langle n \rangle$ after a time $t'_c \ll 1$ becomes an average over $n = N$ and $n = N - 1$ weighted by the quasi steady-state probabilities $p^{qe}_{1,0}$ and $p^{qe}_{0,0}$ respectively, hence the step-like drop in the mean predicted by the SSA at $t' = t'_c$. The method of averaging in the stochastic QEA assumes the immediate occurrence of the equilibrium in group 0 at $t' = 0$, hence the dispatch of $\langle n(t' = 0) \rangle$ from the exact initial condition. This also explains why the standard deviation predicted by the stochastic QEA (notably in Fig. 3.2A(iv)) appears to be non-zero at $t' = 0$: *since the system is modelled to be in quasi-equilibrium at initiation, equilibrium fluctuations are present even at* $t' = 0$. However, so long as the SSA parameters are chosen such that $k_1/k_2 \gg 1$ the stochastic QEA provides a very good approximation even for small times so long as $t' > t'_c$. Additionally, we compare $\langle n \rangle$ to the deterministic mean number of free substrate, $\langle n \rangle_d$, predicted the numerical solution of Eq. (3.3). Overall, the deterministic solution is found to be in good agreement with the mean predicted by the SSA and the stochastic QEA, however there does exist a small disagreement where the mean number of substrate molecules is small (seen more explicitly in Fig. 3.2B(iv)). This disagreement occurs since molecular discreteness is very important where $\langle n \rangle$ is very small, and properly accounting for it leads to differing dynamics for $\langle n \rangle$ in this region, whereas the behaviour of $\langle n \rangle_d$ does not change compared to $\langle n \rangle_d \gtrapprox 1$, since the deterministic analysis considers molecule number to be continuous. As we shall see later, increasing the number of enzyme molecules removes this discrepancy between the stochastic and deterministic means, highlighting that the discrepancy seen here is because we do not consider enzyme molecules to be discrete in the deterministic analysis.

From Fig. 3.2B(iii) we observe that the distribution of substrate molecule numbers can be bimodal at intermediate times (there are two peaks at $n = 0$ and $n = 6$ at $t' = 45$). This bimodality, though less conspicuous, can in fact be also observed in Fig. 3.2A(ii) with peaks at $n = 0$ and $n = 2$. From Fig. 3.2A(iv) and B(iv), we can see that in both cases the bimodality occurs at a time t' when $\langle n \rangle - \sigma \approx 0$, i.e., when the fluctuations are large enough to cause frequent transitions to the absorbing state. This type of dynamical phase transition (which we shall refer to as transient bimodality), from a unimodal distribution to a bimodal one and then back to a unimodal one, as time progresses, has also been recently observed in genetic feedback loops [239] and is known in non-biological systems [244, 245]. We will discuss this phenomenon more extensively in later sections.

3.4.2 Multiple enzymes

We now extend the solution to the enzyme system (3.1) to the case where initially there are N free substrate molecules and M free enzyme molecules with the constraint of substrate abundance, i.e., $N \geq M$. Note that the solution to the system with $M \geq N$ follows as a special case of the $N \geq M$ system, discussed at the end of this section.

We proceed in solving this system as we did in the single enzyme case: assuming k_2 is sufficiently small, we group the microstates governed by the fast processes together to form $N + 1$ groups between which the transitions are significantly slower than those between the fast internal states of an individual group. The Markov chain describing the system split into groups is shown in Fig. 3.3. Our task is then to find (i) the equilibrium probabilities $p^{qe}_{i,m}$ of being in each fast internal state i (considering only the reactions between the internal states in group m) and (ii) to find the time-dependent probability $p^{g}_{m}(t)$ of being in group m. Knowledge of both (i) and (ii) will allow us to approximate the distribution of interest, $P(n;t)$.

We begin by finding the probabilities $p^{qe}_{i,m}$ and redefine it for the case of multiple enzyme: $p^{qe}_{i,m}$ is the equilibrium probability of having $M-i$ free enzymes in the case of a reduced system involving only the reactions among the fast internal states contained in group m. Now, finding $p^{qe}_{i,m}$ for any group $0 \leq m < (N-1)$ is more complicated than was the case for a single enzyme system, since there we had only two fast internal states in each group. To proceed we consider the following Markovian dynamics of a system with $L + 1$ possible microstates:

$$0 \underset{k_{1,0}}{\overset{k_{0,1}}{\rightleftharpoons}} 1 \underset{k_{2,1}}{\overset{k_{1,2}}{\rightleftharpoons}} \ldots \underset{k_{L,L-1}}{\overset{k_{L-1,L}}{\rightleftharpoons}} L. \tag{3.24}$$

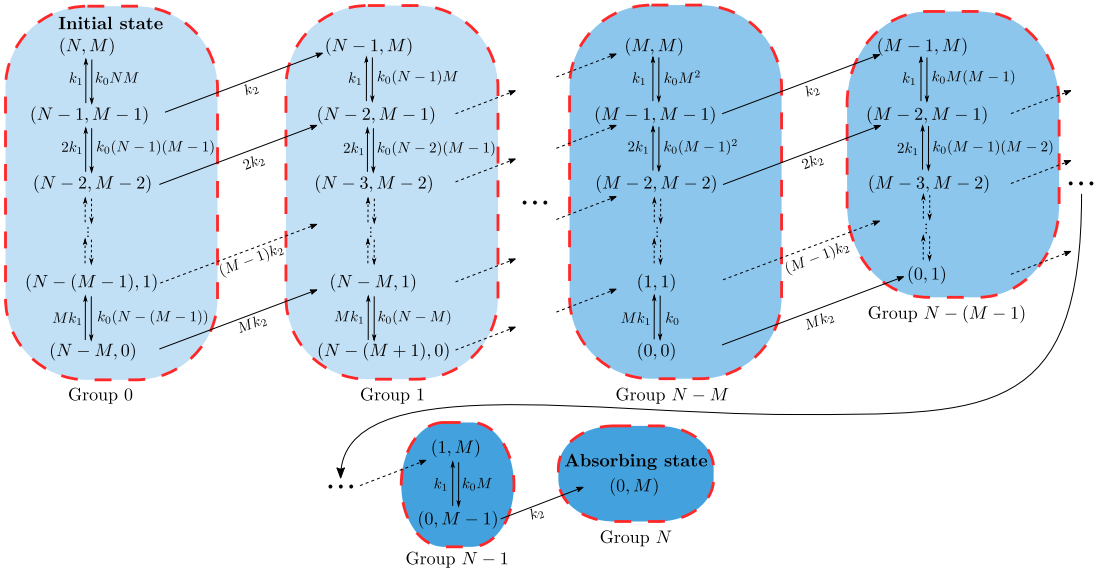

Figure 3.3: Illustration showing the transitions between the discrete microstates of the enzyme system (3.1) with initially M enzymes and N substrate molecules where $N \geq M$. Microstates that are in rapid equilibrium with each other are aggregated together, with each set of such fast internal states corresponding to a *group*. Namely, the label 'group m' denotes the set of microstates that exist when there are m product molecules, given that there are no product molecules initially. Groups $0 \leq m \leq N-M$ have $M+1$ fast internal states, whereas groups $N-M < m \leq N$ have $N-m+1$ fast internal states. Note that as $t \to \infty$ we are guaranteed to be in the absorbing state $(0, M)$.

One can then write the master equation for this dynamical system in matrix form:

$$\partial_t \underline{P}_t = \mathcal{M} \cdot \underline{P}_t, \tag{3.25}$$

where $\underline{P}_t = (P_t(0), P_t(1), ..., P_t(L))$, $P_t(i)$ is the probability of being in microstate i at time t and

$$\mathcal{M} = \begin{pmatrix} -k_{0,1} & k_{1,0} & & & & \\ k_{0,1} & -(k_{1,0}+k_{1,2}) & k_{2,1} & & & \\ & k_{1,2} & -(k_{2,1}+k_{2,3}) & k_{3,2} & & \\ & & \ddots & \ddots & \ddots & \\ & & & & k_{L-1,L} & -k_{L,L-1} \end{pmatrix}. \tag{3.26}$$

Enforcing the quasi-equilibrium condition, $\partial_t(\cdot) = 0$, converts the system of $L+1$ ODEs in Eq. (3.25) into a system of $L+1$ simultaneous equations in the equilibrium microstate probabilities $P(i)$, given by $\mathcal{M} \cdot \underline{P} = 0$. One can explicitly solve this set of simultaneous equations under the constraint $\sum_i P(i) = 1$, yielding the probabilities:

$$P(i) = \frac{\left(\prod_{j=1}^{i} k_{j-1,j}\right) \times \left(\prod_{j=i+1}^{L} k_{j,j-1}\right)}{\sum_{i=0}^{L} \left[\left(\prod_{j=1}^{i} k_{j-1,j}\right) \times \left(\prod_{j=i+1}^{L} k_{j,j-1}\right)\right]}. \tag{3.27}$$

Note that due to the definition of the empty product being equal to 1, when we have either $i = 1$ or $i = L$ the numerator of Eq. (3.27) is equal to 1. Further note that one could also utilise the King-Altman method [246, 247] to arrive at Eq. (3.27). Using this result we can find the quasi-equilibrium probabilities for each group shown in Fig. 3.3. First, we consider the groups $0 \leq m \leq N - M$, each with $M + 1$ fast internal states as these groups contain more (or the same number) free substrate molecules than enzymes. Taking the specific example of group $m = 0$, we see that we have a total of $M + 1$ microstates, i.e., $L = M$, $k_{j-1,j} = k_0(N - (j-1))(M - (j-1))$ and $k_{j,j-1} = jk_1$, with $1 \leq j \leq M$. Identifying $p^{qe}_{i,0}$ with $P(i)$ in Eq. (3.27), we find that:

$$p^{qe}_{i,0} = \frac{k_0^i k_1^{M-i} \left\{\prod_{j=1}^{i}(N-(j-1))(M-(j-1))\right\} \times \left\{\prod_{j=i+1}^{M} j\right\}}{\sum_{i=0}^{M} \left[k_0^i k_1^{M-i} \left\{\prod_{j=1}^{i}(N-(j-1))(M-(j-1))\right\} \times \left\{\prod_{j=i+1}^{M} j\right\}\right]}. \tag{3.28}$$

The result can be easily generalised for groups $0 \leq m \leq N - M$ and $0 \leq i \leq M$:

$$p^{qe}_{i,m} = \frac{k^{-i} \left\{\prod_{j=1}^{i}((N-m)-(j-1))(M-(j-1))\right\} \times \left\{\prod_{j=i+1}^{M} j\right\}}{\sum_{i=0}^{M} \left[k^{-i} \left\{\prod_{j=1}^{i}((N-m)-(j-1))(M-(j-1))\right\} \times \left\{\prod_{j=i+1}^{M} j\right\}\right]}, \tag{3.29}$$

where we have re-introduced $k = k_1/k_0$. The dynamics of groups $N - M < m \leq N$ are slightly different as they contain fewer substrate molecules than enzymes. These groups correspondingly have $N - m + 1$ fast internal states, i.e., $0 \leq i \leq N - m$. This leads to quasi-equilibrium probabilities of the form:

$$p^{qe}_{i,m} = \frac{k^{-i} \left\{\prod_{j=1}^{i}((N-m)-(j-1))(M-(j-1))\right\} \times \left\{\prod_{j=i+1}^{N-m} j\right\}}{\sum_{i=0}^{N-m} \left[k^{-i} \left\{\prod_{j=1}^{i}((N-m)-(j-1))(M-(j-1))\right\} \times \left\{\prod_{j=i+1}^{N-m} j\right\}\right]}, \tag{3.30}$$

Finally, by defining

$$g(m) = \Theta(m - (N - M)) \times (m - (N - M)), \tag{3.31}$$

where $\Theta(m-(N-M))$ is the Heaviside step function, we can write down a joint expression for all groups $0 \leq m \leq N$ and $0 \leq i \leq M - g(m)$:

$$p^{qe}_{i,m} = \frac{z_{i,m}}{\mathcal{Z}_m}, \tag{3.32}$$

with

$$z_{i,m} = k^{-i}\left\{\prod_{j=1}^{i}((N-m)-(j-1))(M-(j-1))\right\} \times \left\{\prod_{j=i+1}^{M-g(m)} j\right\}, \tag{3.33}$$

$$\mathcal{Z}_m = \sum_{i=0}^{M-g(m)} z_{i,m}. \tag{3.34}$$

We now proceed to calculate $p^g_m(t)$. From Fig. 3.3, we observe that the transitions between the groups are described by the master equation identical in form to Eq. (3.9). However, the transition rates a_m in this case are different, as the group m can be reached from any of the $M - g(m-1)$ microstates in the group $m-1$ (excluding only the microstate with M free enzymes) and we must also take into account the quasi-equilibrium probabilities of being in the corresponding microstate. It follows that the transition rates can be defined as:

$$a_m = \sum_{n=1}^{M-g(m-1)} n p^{qe}_{n,m-1} = -k\partial_k(\ln(\mathcal{Z}_{m-1})),\ 1 \leq m \leq N+1, \tag{3.35}$$

$$= \begin{cases} -M \times \left(\frac{k\ {}_1F_1(1-M,-m-M+N+3;-k)}{(-m-M+N+2)\ {}_1F_1(-M,-m-M+N+2;-k)} - 1\right), & m \leq N-M+1, \\ -(N-m+1) \times \left(\frac{k\ {}_1F_1(m-N,m+M-N+1;-k)}{(m+M-N)\ {}_1F_1(m-N-1,m+M-N;-k)} - 1\right), & m > N-M+1, \end{cases}$$

where ${}_1F_1(a,b;c)$ is the confluent hypergeometric function, a result which we prove in Appendix A.3. As the dynamics between the groups are identical to the single enzyme case, $p^g_m(t')$ has exactly the same form as Eq. (3.20) but with the eigenvalues of $\mathcal{Q}$ being given by $\lambda_i = -a_i$, where the a_i are now defined in Eq. (3.35).

We can now obtain the probability distribution $P(n;t)$, which requires us to find all microstates in the system containing n free substrate molecules. From Fig. 3.3 we see that for substrate numbers n, where $0 \leq n \leq N-M$, there are $M+1$ corresponding microstates given by $(n,0),(n,1),...,(n,M)$ which respectively belong to groups $(N-M)-n,(N-M)-n+1,...,N-n$. Therefore, the distribution has the form:

$$P(n;t') = \sum_{j=0}^{M} p^{qe}_{j,N-(n+j)}\, p^g_{N-(n+j)}(t')\ , \quad \text{where} \quad 0 \leq n \leq N-M. \tag{3.36}$$

In the case of $N-M < n \leq N$, there are $N-(n-1)$ microstates containing n substrate molecules, explicitly defined as $(n, M-(N-n)), (n, M-(N-n)+1), ..., (n, M)$ and associated with groups $0, 1, ..., N-n$ respectively. Hence we have:

$$P(n;t') = \sum_{j=0}^{N-n} p^{qe}_{j,N-(n+j)}\, p^{g}_{N-(n+j)}(t') \;, \quad \text{where} \quad N-M < n \leq N. \tag{3.37}$$

Finally, using the function $g(m)$ previously defined in Eq. (3.31), we obtain:

$$P(n;t') = \sum_{j=0}^{M-g(n)} p^{qe}_{j,N-(n+j)}\, p^{g}_{N-(n+j)}(t') \;, \quad \text{where} \quad 0 \leq n \leq N, \tag{3.38}$$

which fully describes the time-dependent solution for the multiple enzyme system $N \geq M$ with the initial condition $p^g_m(0) = \delta_{m,0}$. Note that the solution can also be extended to a more general initial distribution in the same way as was done for the single enzyme system in Section 3.4.1. The equations for mean number of substrate, $\langle n(t')\rangle$, and standard deviation, $\sigma(t')$, at rescaled time t' are the same as in Eqs. (3.22)–(3.23), but where $P(n;t')$ is now given by Eq. (3.38).

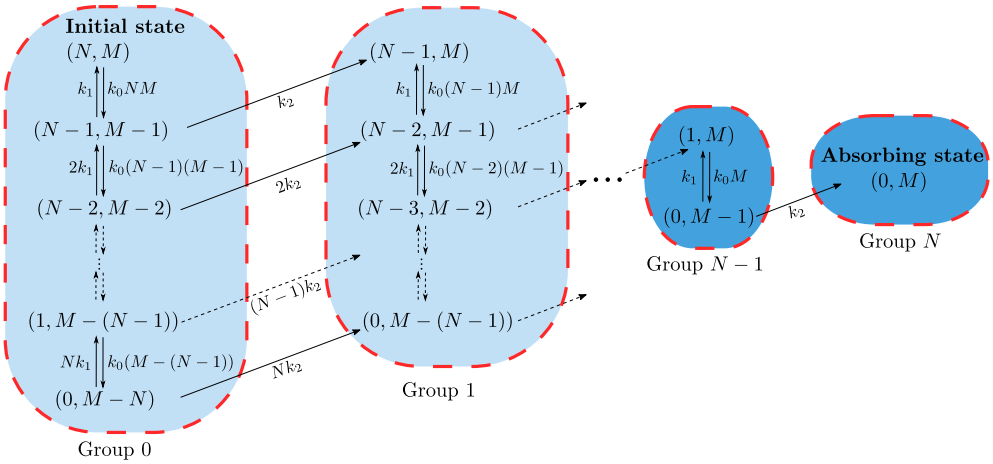

Figure 3.4: Illustration showing the transitions between the discrete microstates of the enzyme system (3.1) with initially M enzymes and N substrate molecules where $M \geq N$. As before, fast internal states are aggregated together into groups. The dynamics of the groups 0 to N can be mapped onto the dynamics of groups $N-M$ to N in the system with $N \geq M$ (shown in Fig. 3.3). See text for discussion.

Now consider a multiple enzyme system which initially contains fewer free substrate molecules than enzymes, i.e., $M \geq N$. The Markov chain describing the transitions between the microstates of this system, shown in Fig. 3.4, has similarities to that for the system with $N \geq M$. Specifically, if we replace N by M in groups 0 to N in the

$M \geq N$ case of Fig. 3.4 then we exactly recover groups $N - M$ to N in the $N \geq M$ case of Fig. 3.3. This mapping implies that the dynamics of the system with $M \geq N$ are correctly described by Eq. (3.38) due to the utility of $g(m)$. Therefore, Eq. (3.38) is a valid solution for any positive integer values of N and M.

As for the single enzyme case, we can make the initial statement that k_2 must be sufficiently small for the derivation to hold, more precise. Suppose we are in the microstate (n, n_e). There are then 3 possible reactions which can occur: (i) $(n, n_e) \rightarrow (n, n_e + 1)$ with rate $k_2(M - n_e)$, (ii) $(n, n_e) \rightarrow (n + 1, n_e + 1)$ with rate $k_1(M - n_e)$ and (iii) $(n, n_e) \rightarrow (n - 1, n_e - 1)$ with rate $k_0\, n\, n_e$. Only the first reaction leads to a transition out of the current group of microstates (since its associated with the product formation step) and hence the probability of exiting the current group is $k_2(M-n_e)/((k_1+k_2)(M-n_e)+k_0\ n\ n_e)$. It is easy to prove that the latter is always less than $k_2/(k_1+k_2)$. Hence quasi-equilibrium of microstates in each group is possible when $k_2/(k_1+k_2) \ll 1$. In other words, generally the closed-form solution for the distribution of substrate numbers will be accurate for all times provided $k_1 \gg k_2$.

In Fig. 3.5A(i-iii) and 3.5B(i-iii) we show agreement between $P(n; t')$ from Eq. (3.38) and the SSA where $k_1 \gg k_2$ is enforced, over times ranging between the initial time, when the number of substrate is $n = N$ and the absorbing state at $n = 0$ for large times, for cases $M \geq N$ and $N \geq M$ respectively. In Fig. 3.5A(iv) and 3.5B(iv) we plot the mean and standard deviation of our analytical distribution ($\langle n \rangle$, σ), the deterministic mean $\langle n \rangle_d$ and the mean predicted by the SSA for $M \geq N$ and $N \geq M$ respectively. The SSA prediction of the mean is shown to be in exact correspondence with $\langle n \rangle$ when the QEA holds. The discrepancy previously seen in Fig. 3.2B(iv) between $\langle n \rangle$ and $\langle n \rangle_d$ at low molecule number is no longer observed in Fig. 3.5A(iv) where $M = \mathcal{O}(N)$, highlighting that the discrepancy seen in Fig. 3.2B(iv) originates from the molecular discreteness of the enzyme species. We additionally note the presence of transient bimodality in Fig. 3.5B(ii) similar to that seen in the single enzyme case from Section 3.4.1; note that the parameter set chosen for Figs. 3.5A(i-iii) does not exhibit transient bimodality. The parameter space of transient bimodality is explored later in more detail in Section 3.4.2. In Fig. 3.6 we demonstrate using stochastic simulations that, as predicted by our theory, the requirement for the stochastic QEA to be a good approximation relies only on satisfying the condition $k_1 \gg k_2$, and does not require any additional constraint on the value of k_0.

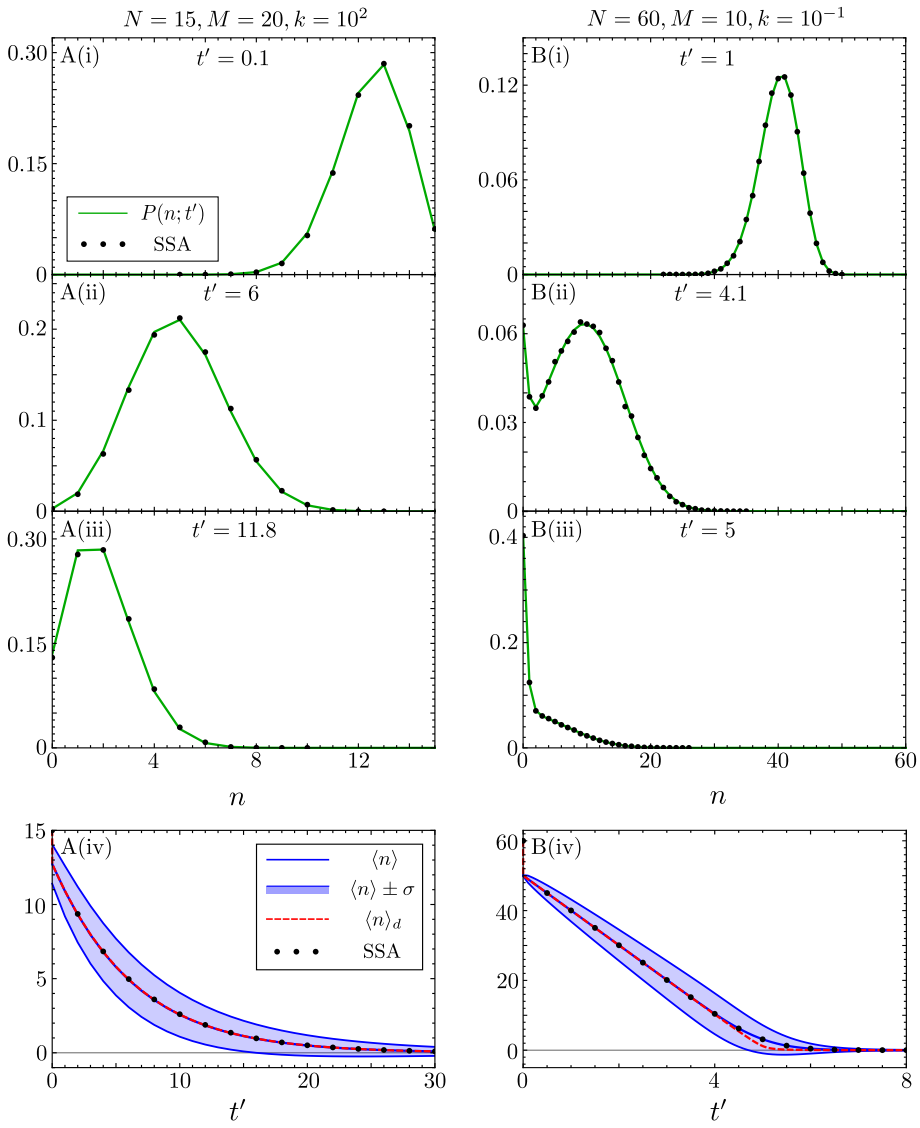

Figure 3.5: Comparison of the closed-form time-dependent probability distribution of substrate molecules, for the enzyme reaction (3.1) with multiple enzyme molecules M, and initial substrate molecules N, to the distribution obtained from the SSA. Note that the closed-form solution is given by Eq. (3.38). In A(i)–(iii), $N = 15, M = 20, k = 10^2$ and we simulate the SSA using $k_0 = 1$, $k_1 = 10^2$ and $k_2 = 1$; the theory (green lines) agrees with the SSA since the quasi-equilibrium assumption is justified, i.e., $k_1/k_2 \gg 1$. In B(i)–(iii), $N = 60, M = 10, k = 10^{-1}$ and we simulate the SSA using $k_0 = 10^3$, $k_1 = 10^2$ and $k_2 = 1$; again the theory is in agreement with the SSA since quasi-equilibrium is justified. Note that these results show that the theory accurately describes both the $N \geq M$ and the $M \geq N$ cases. In A(iv) and B(iv) we show the corresponding plots of the time-evolution of the mean $\langle n \rangle$ and of the standard deviation σ of the distributions of substrate molecules, as predicted by our theory; these are compared with the mean calculated from the SSA and the corresponding mean $\langle n \rangle_d$ obtained from the numerical solution of the deterministic rate equations given by Eq. (3.3). The parameter set in B is shown to be transiently bimodal in B(ii), whereas for the parameter set describing A transient bimodality is not observed. Each SSA probability distribution here is constructed from 10^5 individual reaction trajectories.

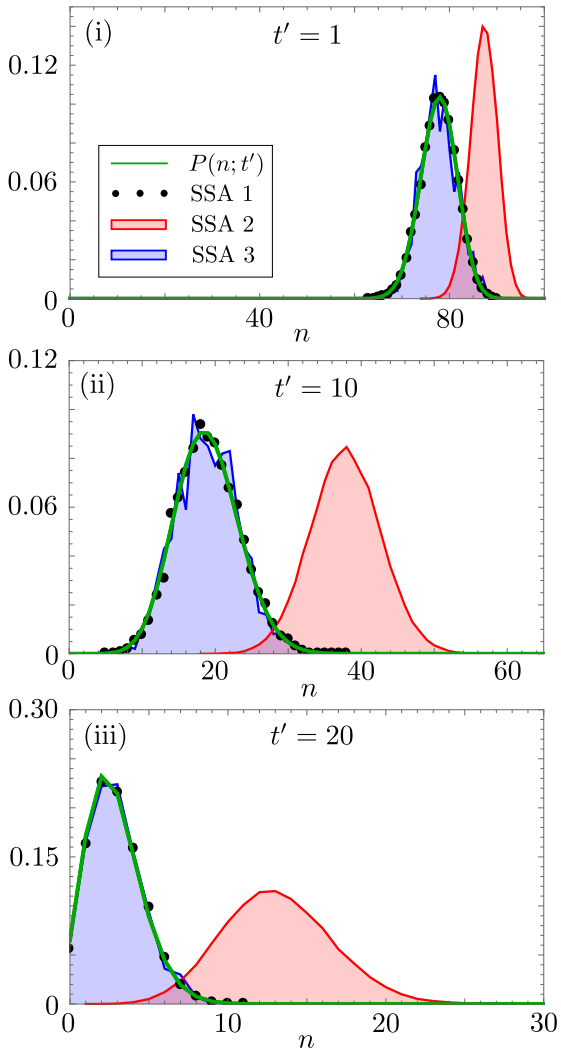

Figure 3.6: Testing the conditions necessary for the accuracy of the stochastic QEA. The three panels (i)–(iii) show how the accuracy of the closed-form time-dependent solution changes with time as we vary k_0/k_2 and k_1/k_2 whilst keeping $k = k_1/k_0$ fixed to 10^2 for the initial substrate number $N = 10^2$ and the total number of enzyme molecules equal to $M = 25$. The green line denotes the stochastic QEA solution from Eq. (3.38); SSA 1 (black dots) denotes the SSA prediction with parameters $k_0/k_2 = 1$, $k_1/k_2 = 10^2$ calculated over 10^4 trajectories; SSA 2 (blocked red region) denotes the SSA prediction with parameters $k_0/k_2 = 10^{-2}$, $k_1/k_2 = 1$ calculated over 10^5 trajectories; SSA 3 (blocked blue region) denotes the SSA prediction with parameters $k_0/k_2 = 10$, $k_1/k_2 = 10^3$ calculated over 10^3 trajectories. It is clear that SSA 2 is poorly predicted by $P(n;t)$, which is expected as $k_1 = \mathcal{O}(k_2)$. Since $P(n;t)$ is in equally good agreement with SSA 1 and SSA 3 it can be seen that the only requirement is $k_1 \gg k_2$, without requiring additional constraints on k_0.

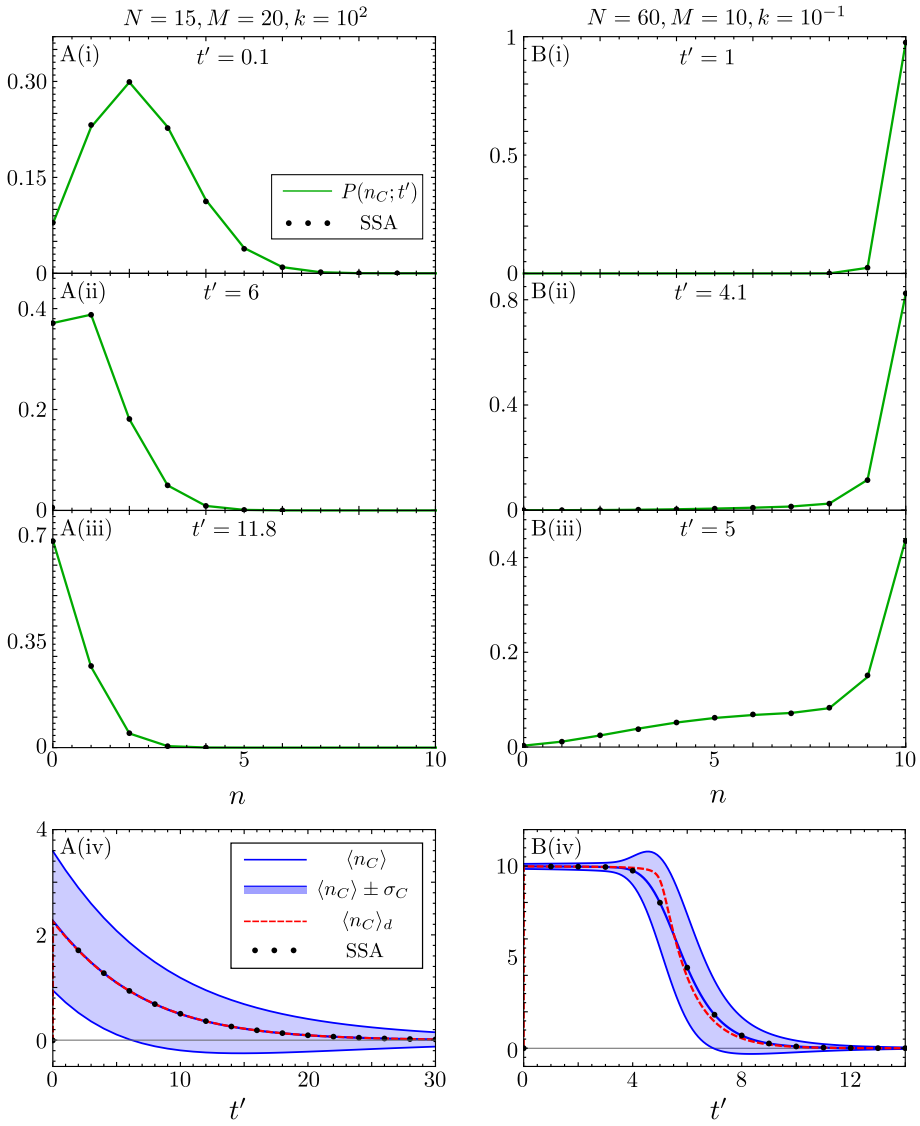

Figure 3.7: Comparison of the closed-form time-dependent probability distribution of enzyme-substrate complexes, for the enzyme reaction (3.1) with multiple enzyme molecules M, and initial substrate molecules N, to the distribution obtained from the SSA. Note that the closed-form solution is given by Eq. (3.40). In A(i)–(iii), $N = 15, M = 20, k = 10^2$ and we simulate the SSA using $k_0 = 1$, $k_1 = 10^2$ and $k_2 = 1$; In B(i)–(iii), $N = 60, M = 10, k = 10^{-1}$ and we simulate the SSA using $k_0 = 10^3$, $k_1 = 10^2$ and $k_2 = 1$ (parameters are the same as in Fig. 3.5). In both cases, the theory (green lines) agrees with the SSA since the quasi-equilibrium assumption is justified, i.e., $k_1/k_2 \gg 1$. In A(iv) and B(iv) we show the corresponding plots of the time-evolution of the mean $\langle n_C \rangle$ and of the standard deviation σ_C of the distributions of enzyme-substrate complex, as predicted by our theory; these are compared with the mean calculated from the SSA and the mean $\langle n_C \rangle_d$ obtained from the numerical solution of the deterministic rate equations given by Eq. (3.3). Each SSA probability distribution here is constructed from 10^5 individual reaction trajectories.

Time-dependent solution for the probability distribution of enzyme molecules

Having solved the master equation for the group dynamics, it is relatively straightforward to extract the time-dependent probability distribution for the number of free enzyme molecules, $P(n_E;t')$, and hence the distribution for the number of enzyme-substrate complexes, $P(n_C;t')$. As previously, we begin by considering the $N \geq M$ system depicted in Fig. 3.3. We observe that the groups $0 \leq m \leq N-M$ all contain a microstate with n_E free enzyme molecules, where $0 \leq n_E \leq M$, as enzymes are saturated with substrate. However, for groups $N-M < m \leq N$, free enzymes become more abundant than free substrate molecules, so that microstates containing $0 < n_E \leq M$ enzymes are found only in groups $N-M < m \leq N-(M-n_E)$. Note that the quasi-equilibrium probability of having n_E free enzymes in group m is $p^{qe}_{M-n_E,m}$, given by Eq. (3.32), and the group probabilities $p^g_m(t')$ are identical to the ones defined for the distribution of substrate number in Eq. (3.38). Therefore, the distribution of free enzymes takes the form:

$$P(n_E;t') = \sum_{j=0}^{N-(M-n_E)} p^{qe}_{M-n_E,j}\, p^g_j(t')\ , \quad 0 \leq n_E \leq M. \tag{3.39}$$

This expression is valid for any positive integer values of N and M, again due to the mapping between the Markov chains of $N \geq M$ and $M \geq N$ systems, described above. Moreover, for the $N \leq M$ system, the definition of an empty sum as zero ensures that non-physical values of n_E are not allowed, i.e., the number of bound enzymes cannot be larger than N given the chosen initial conditions, so that $P(n_E;t') = 0$ for $n_E < M-N$. Finally, as $n_C = M - n_E$, the probability distribution of the enzyme-substrate complex follows trivially:

$$P(n_C;t') = \sum_{j=0}^{N-n_C} p^{qe}_{n_C,j}\, p^g_j(t')\ , \quad 0 \leq n_C \leq M. \tag{3.40}$$

In Fig. 3.7A(i-iii) and 3.7B(i-iii) we confirm that $P(n_C;t')$ from Eq. (3.40) and the SSA are in good agreement for enzyme systems with $M \geq N$ and $N \geq M$ respectively over the whole time-range from near the initial condition to the absorbing state, where again $k_1 \gg k_2$ is enforced (using the same parameters as in Fig. 3.5). Note that the transient bimodality is seemingly not manifest in $P(n_C;t')$ at the points in the parameter space where it is observed for the distribution of substrate number (c.f. Fig. 3.5B(ii) and 3.7B(ii)). In Fig. 3.7A(iv) and Fig. 3.7B(iv) we plot the mean and standard deviation of our analytical distribution for the enzyme-substrate complexes ($\langle n_C \rangle$ and σ_C), the

mean predicted by the SSA and the mean number of complex molecules $\langle n_C \rangle_d$ obtained from the numerical solution of the deterministic rate equations given by Eq. (3.3) for $M \geq N$ and $N \geq M$ respectively. The SSA prediction of the mean matches $\langle n_C \rangle$ for all times further validating our solution, given that the QEA condition holds.

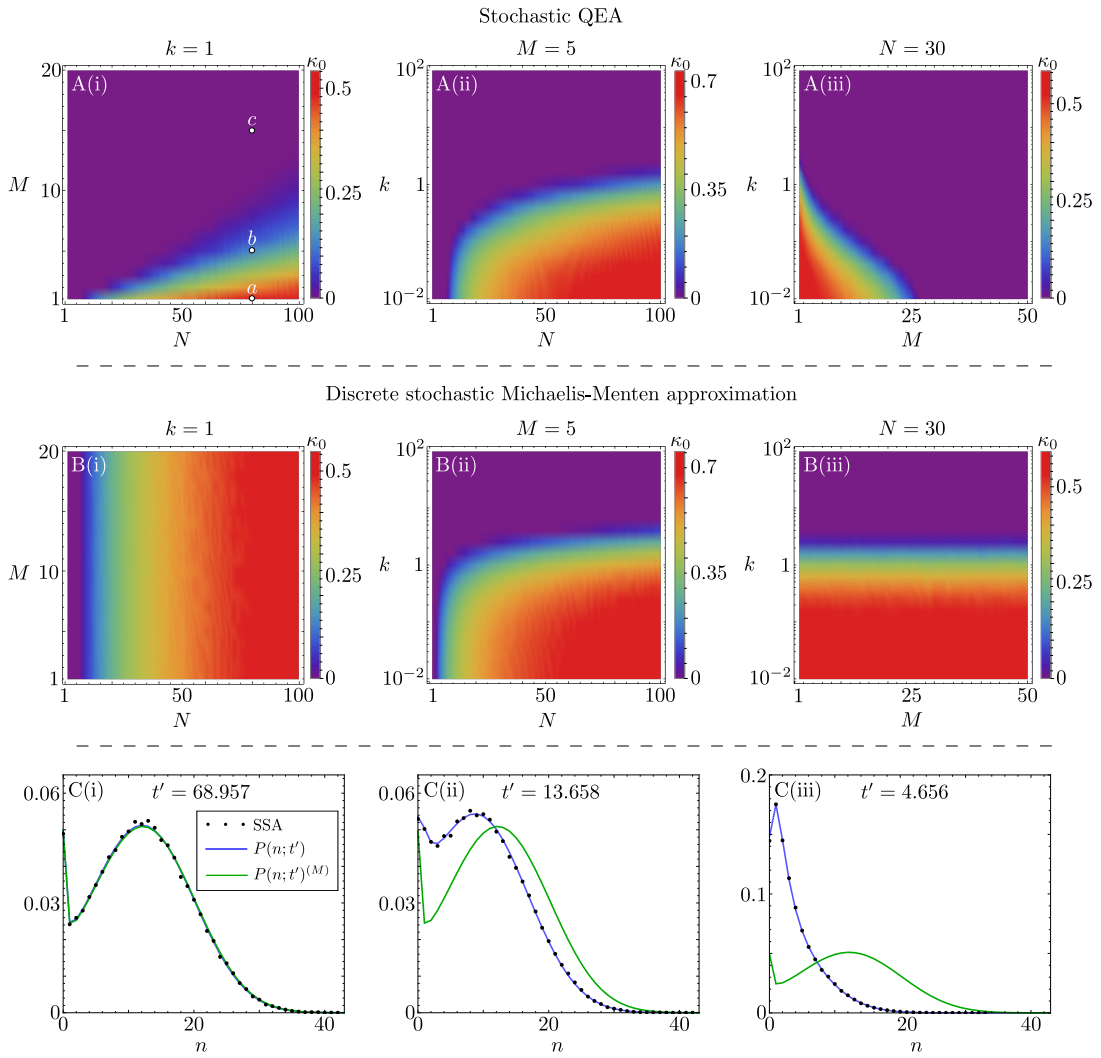

Figure 3.8: Heatmaps elucidating the regions of parameter space where transient bimodality is observed using the stochastic QEA solution (A(i)–(iii)) from Eq. (3.38) and the discrete stochastic MM approximation (B(i)–(iii)) given by Eq. (3.46). Note that κ_0 is a measure of how bimodal is the distribution of substrate molecules across the timecourse of the reaction (see text for details). Three parameter regimes are considered: N vs M with $k = 1$ (left), N vs k with $M = 5$ (middle) and M vs k with $N = 30$ (right). The plots C(i)–(iii) show the closed-form distributions of the stochastic QEA, $P(n;t')$, and the discrete stochastic MM approximation, $P(n;t')^{(M)}$, at the times when the stochastic QEA exhibits maximum bimodality, for cases with $k = 1$, $N = 80$ and (i) $M = 1$, (ii) $M = 5$ and (iii) $M = 15$ (highlighted on the heatmap A(i) as the points a, b and c respectively). The corresponding SSA predictions with $k_0/k_2 = 10^2$ and $k_1/k_2 = 10^2$ are also included (constructed from 10^5 individual reaction trajectories). Note that the two distributions (discrete stochastic MM approximation and stochastic QEA) are almost identical in C(i), but the difference becomes more pronounced in C(ii) and C(iii) with increasing M.

Bimodality

In Fig. 3.8A(i)–(iii) we explore further the transient bimodality observed in Figs. 3.2A(ii), 3.2B(iii) and 3.5B(ii). Namely, we investigate how the strength of the bimodality varies with the parameters N, M and k using the stochastic QEA solution from Eq. (3.38). Each point on the heatmap in Fig. 3.8A(i)–(iii) shows, for a particular parameter set, the maximum of the strength of bimodality calculated over the entire time course from $t' = 0$ to a time near the absorbing state of $n = 0$. We utilise the measure of bimodality strength introduced in [239], which is explicitly given by:

$$\kappa = \frac{H_{\text{low}} - H_{\text{valley}}}{H_{\text{high}}}, \tag{3.41}$$

where H_{low} and H_{high} are the heights of the smallest and largest magnitude modes respectively, and H_{valley} is the height of the valley between the modes. For bimodal distributions κ has a value between 0 (no bimodality) and 1 (maximum bimodality), and for monomodal distributions is defined as zero. This definition of bimodality strength considers the 'most bimodal' distributions to have modes of equal height with a deep valley between them. In order to produce each heatmap we devised a simple algorithm, as follows. For each parameter set $\{N, M, k\}$:

1. Calculate the estimated time to reach the absorbing state which provides us with the time range, T_a, over which the transient bimodality search will be conducted. In order to avoid additional computational burdens of finding the absorption time using stochastic simulations, we use a much simpler but reasonably accurate estimate obtained from the deterministic QEA mean instead, given by solving Eq. (3.5) for $t' = k_2 t$:

$$T_a = \frac{N}{M} - \frac{k}{M} \log\left(\frac{\langle n\rangle_a e^{\frac{\langle n\rangle_a}{k}}}{N}\right), \tag{3.42}$$

   where we set $\langle n\rangle_a = 10^{-2}$, which was chosen small enough such that transient bimodality for all parameter sets was accounted for.
2. Choose the number of iterations, I, over which to check if the distribution is bimodal. In our case we chose $I = 400$. This gives the set of times over which we check for bimodality as $t_i = iT_a/I$ for $1 \leq i \leq I$.
3. Define a variable denoting the maximum bimodality measure κ_0 which is initially set to zero. For each t_i find the number of peaks in the distribution given by Eq. (3.38) for the stochastic QEA, and if two peaks are detected, calculate the bimodality strength κ from Eq. (3.41). If $\kappa > \kappa_0$ then set $\kappa_0 = \kappa$. Do for all t_n.

4. Once all iterations of this process are complete, the value of κ_0 will denote the largest value of the transient bimodality measure for all probability distributions at $t \in t_i$. We take κ_0 as the largest value of transient bimodality encountered on the time course.

The results obtained using this algorithm are summarised by the three heatmaps in Fig. 3.8A(i)–(iii). The distribution of substrate molecules corresponding to the time at which the maximal bimodality strength κ_0 occurs for points a, b, c in Fig. 3.8A(i) are shown by the solid blue lines in 3.8C(i)–(iii), respectively. Note that the bimodality is most pronounced in C(i), less in C(ii) and least in C(iii), in accordance with the value of κ_0 in Fig. 3.8A(i); this validates the use of Eq. (3.41) as an accurate measure of the strength of bimodality. From Fig. 3.8A(i)–(iii), it is clear that bimodality is most pronounced when the initial number of substrate molecules N is significantly larger than the total enzyme number M and also when k is small, i.e., when the frequency of enzyme-substrate binding is much larger than the frequency of complex dissociation into enzyme and substrate. Note that generally the frequency of enzyme-substrate binding is inversely proportional to the volume of the compartment [127] in which the bimolecular reaction occurs and hence the transient bimodality is likely observable inside cells.

3.5 The discrete stochastic Michaelis-Menten approximation

We next consider how the analytical solution that we obtained for the reaction system (3.1) using a combination of averaging and linear algebra techniques in Section 3.4.2 compares with the solution of a commonly used reduced CME for enzyme kinetics.

The reduced CME for single substrate enzyme kinetics can be heuristically justified as follows (for a derivation see [62]). Under the QEA approximation, from the deterministic analysis in Section 3.3, it follows that the rate equation describing the time-evolution of the substrate concentration is given by:

$$\frac{d[S(t)]}{dt} = -\frac{V_{max}[S(t)]}{k + [S(t)]}. \tag{3.43}$$

Note that $V_{max} = k_2 M$, where M is the total number of enzyme molecules. Hence, species S can be seen as changing into P by means of an effective first-order decay reaction with rate given by the right hand side of Eq. (3.43). One common way to approximately describe the enzyme reaction stochastically consists of writing down an effective propensity describing the decay of substrate, i.e., we postulate that if there are m substrate molecules at time t then the probability that a reaction $S \to P$ occurs somewhere in a unit volume in the time interval $[t, t + dt)$ is approximately given by $a_m dt$ where $a_m = V_{max} m/(k + m)$. This is the discrete stochastic Michaelis-Menten

(MM) approximation. Hence if we choose an initial condition of N substrate molecules, it follows that a corresponding effective CME is given by:

$$\partial_t P_{N-m}(t) = a_{m+1} P_{N-(m+1)}(t) - a_m P_{N-m}(t), \tag{3.44}$$

where $P_{N-m}(t)$ is the probability that there are m substrate molecules at time t ($0 \leq m \leq N$). This CME can be conveniently written as:

$$\partial_t \underline{P}(t) = \mathcal{Q} \cdot \underline{P}(t) \tag{3.45}$$

where $\underline{P}(t) = (P_0(t), P_1(t), ..., P_N(t))$ and $\mathcal{Q}$ is a $(N+1) \times (N+1)$ lower bidiagonal matrix whose only non-zero elements are $\mathcal{Q}_{i,i} = -a_{N-(i-1)} = -\frac{(N-(i-1))V_{\max}}{k+(N-(i-1))}$ for $1 \leq i \leq N+1$, and $\mathcal{Q}_{i+1,i} = a_{N-(i-1)} = \frac{(N-(i-1))V_{\max}}{k+(N-(i-1))}$ for $1 \leq i \leq N$. Using the method in [129] that was used to solve the master equation for the group dynamics for the single enzyme, the solution is found to be given by Eq. (3.20), modified to take into account the fact that P_{N-n} is equivalent to the probability of being in the group $N-n$:

$$P_{N-n}(t')^{(M)} = \begin{cases} 0, & n > N, \\ e^{\lambda_1^{(M)} t'}, & n = N, \\ \left\{(-1)^{N-n} \prod_{k=1}^{N-n} \lambda_k^{(M)}\right\} \times \left\{\sum_{k=1}^{N-n+1} \frac{e^{\lambda_k^{(M)} t'}}{\prod_{j=1, j\neq k}^{N-n+1} (\lambda_k^{(M)} - \lambda_j^{(M)})}\right\}, & 0 \leq n < N. \end{cases} \tag{3.46}$$

Note the superscript (M) specifying that the solution is for the CME (3.45) resulting from the discrete stochastic MM approximation. Here, we have again rescaled the time $t' = k_2 t$, and $\lambda_m^{(M)}$ are the eigenvalues of $\mathcal{Q}$, which are simply given by the diagonal elements:

$$\lambda_m^{(M)} = -\frac{M(N-(m-1))}{k+N-(m-1)}, \; 1 \leq m \leq N+1. \tag{3.47}$$

We shall denote the time-dependent mean and standard deviation of the distribution Eq. (3.46) by $\langle n(t') \rangle^{(M)}$ and $\sigma(t')^{(M)}$, respectively. Note that the distributions for the number of free enzymes/enzyme-substrate complexes cannot be obtained under the discrete stochastic MM approximation as the enzyme number fluctuations are not taken into account, in contrast to the Stochastic QEA from which enzyme/enzyme-substrate complex distributions can be obtained (see Section 3.4.2).

3.5.1 Comparison with the stochastic QEA

We used the algorithm described in Section 3.4.2 (with the difference that in step 3 we use Eq. (3.46) instead of Eq. (3.38)) to explore the regions of parameter space where the discrete stochastic MM approximation predicts the distribution of substrate molecules to be bimodal. The results are summarised by the three heatmaps in Fig. 3.8B(i)–(iii). By comparison to the heatmaps generated using the stochastic QEA in Fig. 3.8A(i)–(iii), it is clear that the discrete stochastic MM approximation tends to predict bimodality where in reality there is none. Notably, the bimodality predicted by the discrete stochastic MM approximation is independent of M (see Figs. 3.8B(i) and B(iii)) since M only acts to scale the eigenvalues representing the system's relaxation timescales in Eq. (3.47); in contrast, the stochastic QEA predicts bimodality which is strongly dependent on M (see Figs. 3.8A(i) and A(iii)). These issues with the discrete stochastic MM approximation are also clearly discernible in 3.8C(i)–(iii), where we compare the distribution of substrate molecule numbers predicted by this approximation (green line) with that predicted by the SSA (dots) and the stochastic QEA (blue line).

A different way to contrast the discrete stochastic MM approximation and the stochastic QEA involves comparing the eigenvalues of the transition matrix. In the single enzyme case where $M = 1$, one observes that the eigenvalues predicted by Eq. (3.47) exactly match the eigenvalues predicted by averaging for the group dynamics in the single enzyme case from Eq. (3.10). However, note that the group dynamics is not precisely the same as the substrate dynamics which is determined by two microstates in different groups. For example the averaging technique implies that there are two microstates that contain n substrate molecules: $(n, 0)$ and $(n, 1)$ associated with groups $N-(n+1)$ and $N-n$ respectively. However, this subtlety is not important if $N \gg 1$ and hence the CME resulting from the discrete stochastic MM approximation will practically lead to the same results as averaging for most cases of interest.

The comparison is more complicated in the case of multiple enzymes ($M > 1$) and abundant substrate $N \gg 1$, which we explore in Fig. 3.9 (for $N = 100$ and $k = 1$), showing how the discrete stochastic MM approximate solution differs to that from averaging as the ratio M/N is increased. We first consider the case where $M/N = 1/20$, and we see that $\langle n \rangle^{(M)}$ in Fig. 3.9A(i) is a good approximation of $\langle n \rangle$ for the time range of interest, i.e., from the initial state at $N = 100$ to a time $t' = 30$ where both $\langle n \rangle^{(M)}$ and $\langle n \rangle$ are small quantities. Note that the error in the standard deviation for this parameter set, shown in 3.9A(ii), is also small. The slight difference in the relaxation dynamics is corroborated by small differences in the eigenspectra of λ_m (given in Eq. (3.35) again noting that $\lambda_i = -a_i$) and $\lambda_m^{(M)}$ (given by Eq. (3.47)) which can be appreciated in Fig. (3.9)A(iii). We additionally plot the deterministic mean as predicted by Eq. (3.5) which clearly shows the relaxation dynamics of $\langle n \rangle_a$ accurately approximates $\langle n \rangle$ for short times only.

In Figs. 3.9B(i) and C(i) we see that as M/N increases to 1/5 and 1/2 respectively, $\langle n \rangle^{(M)}$ becomes a worse approximation of $\langle n \rangle$, with $\langle n \rangle^{(M)}$ tending more to $\langle n \rangle_a$ than $\langle n \rangle$. The corresponding error in the standard deviation, as shown in 3.9B(ii) and C(ii), also follows that of the mean, increasing with M/N. There are two main reasons for this disagreement:

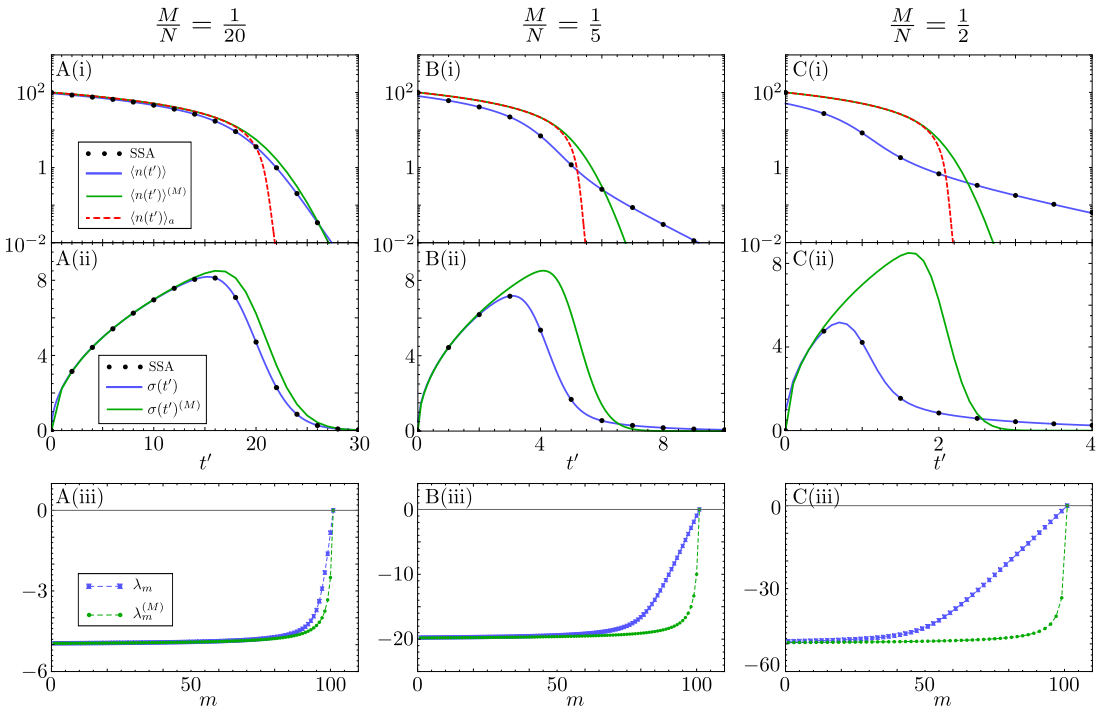

Figure 3.9: Comparison of the discrete stochastic MM approximation and the exact result from averaging in the quasi-equilibrium limit. A(i), B(i) and C(i) show log-scale plots of $\langle n\rangle$, $\langle n\rangle^{(M)}$ and $\langle n\rangle_a$ (from Eq. (3.5)) for $N = 100$, $k = 1$ and $M = 5$ (i.e, $M/N = 1/20$), $M = 20$ (i.e, $M/N = 1/5$) and $M = 50$ (i.e, $M/N = 1/2$) respectively. The corresponding SSA results with $k_0/k_2 = 10^2$ and $k_1/k_2 = 10^2$ are also included (constructed from 10^5 individual reaction trajectories). A(ii), B(ii) and C(ii) are the corresponding plots of the standard deviations $\sigma(t')$, $\sigma(t')^{(M)}$ and that of SSA. A(iii), B(iii) and C(iii) show the eigenspectra for each differing M/N; each symbol corresponds to an individual eigenvalue (since the spectra are discrete) and the dashed lines are only present to aid the reader.

1. If M is comparable to N then initially there will be large fluctuations in the number of enzyme molecules, which are taken into account by the averaging solution (since it allows for switching between microstates in each group) but not by the CME resulting from the discrete stochastic MM approximation (since the total number of enzymes only appears as a constant through V_{max}). This is most clearly seen in Fig. 3.9C(i) where we observe a large discrepancy between $\langle n \rangle$ and $\langle n \rangle^{(M)}$ at $t' = t'_c \ll 1$ (where t'_c is the time over which the initial transient occurs and is indistinguishable from $t' = 0$ in the figure).
2. Where $M/N \approx \mathcal{O}(1)$, the eigenspectra λ_m and $\lambda_m^{(M)}$ show a large disagreement (see Figs. 3.9B(iii) and C(iii)). This leads to the misprediction of the relaxation dynamics of $\langle n \rangle^{(M)}$, which better represents the dynamics predicted by $\langle n \rangle_a$ rather than of $\langle n \rangle$, for both small and large times. This is due to the fact that the effective Michaelis-Menten propensity in the reduced CME Eq. (3.45) is of the same form as the effective rate from the deterministic rate equation given by Eq. (3.43).

In summary, the solution of the CME obtained by the discrete stochastic MM approximation is a good approximation to the solution of the CME derived by averaging provided $N \gg 1$ and $N/M \gg 1$.

3.6 Multi-substrate mechanisms

Thus far we have considered the simple enzyme mechanism shown in (3.1) where an enzyme can catalyze a single type of substrate. However, in nature it is common for one enzyme species to be able to catalyze multiple substrates [223]. Multi-substrate reactions follow various mechanisms that describe how substrates bind and in what sequence. One such common mechanism is that of ternary complex formation, whereby two substrates bind sequentially to an enzyme to form a complex with three molecules. An example is the following mechanism involving two substrate species A and B and two corresponding reaction products, P and Q [223]:

$$\begin{aligned} &E + A \rightleftharpoons EA,\ EA + B \rightleftharpoons EAB,\ EAB \rightarrow EPQ, \\ &EPQ \rightarrow EQ + P,\ EQ \rightarrow E + Q. \end{aligned} \tag{3.48}$$

Note that here we have assumed an ordered binding mechanism, in the sense that binding of A must precede that of B. An alternative is a random binding mechanism, wherein either A or B could first bind the enzyme. We assume that both enzyme-substrate binding reactions and the steps subsequent to complex formation are fast such that we can consider the simpler reaction scheme:

$$E + A + B \underset{k_1}{\overset{k_0}{\rightleftharpoons}} C \xrightarrow{k_2} E + P + Q. \tag{3.49}$$

Note that ordered or random binding mechanisms cannot be distinguished within this reaction scheme. We assume that there are initially N_A molecules of substrate A, N_B molecules of substrate B, where $N_A \geq N_B$, and M free enzymes. There exists a relation between the number of species A and B, denoted n_A and n_B respectively, which we can write as $n_A - n_B = N_A - N_B \equiv \Delta_{AB}$. Hence each microstate of the system is fully specified by (n_B, n_E). Again the

group dynamics where $k_1 \gg k_2$ are given by Eq. (3.20) but the eigenvalues λ_m specific to this mechanism are given by:

$$\lambda_m = -\sum_{n=1}^{M-g(m-1)} np^{qe}_{n,m-1} = k\partial_k(\ln(\mathcal{Z}_{m-1})),\ 1 \leq m \leq N_B + 1, \tag{3.50}$$

where we have now defined

$$p^{qe}_{i,m} = \frac{z_{i,m}}{\mathcal{Z}_m}, \tag{3.51}$$

$$g(m) = \Theta(m - (N_B - M)) \times (m - (N_B - M)), \tag{3.52}$$

$$z_{i,m} = k^{-i}\left\{\prod_{j=1}^{i}((N_A - m) - (j-1))((N_B - m) - (j-1))(M - (j-1))\right\} \times \left\{\prod_{j=i+1}^{M-g(m)} j\right\}, \tag{3.53}$$

$$\mathcal{Z}_m = \sum_{i=0}^{M-g(m)} z_{i,m}. \tag{3.54}$$

Using the results for the group dynamics and quasi-equilibrium probabilities, we can then find the probability distribution for the substrate molecules:

$$P(n_A, n_B; t') = \delta_{n_A - \Delta_{AB}, n_B} \times \left(\sum_{j=0}^{M-g(n_B)} p^{qe}_{j,N_B-(n_B+j)} p^{g}_{N_B-(n_B+j)}(t')\right), \tag{3.55}$$

where $\delta_{i,j}$ is the Kronecker delta symbol. This allows us to find the marginal distributions:

$$P(n_B; t') = \sum_{n_A} P(n_A, n_B; t') = \sum_{j=0}^{M-g(n_B)} p^{qe}_{j,N_B-(n_B+j)} p^{g}_{N_B-(n_B+j)}(t'), \tag{3.56}$$

$$P(n_A; t') = P(n_B + \Delta_{AB}; t'). \tag{3.57}$$

In Fig. 3.10 we compare the analytic marginal distributions against the SSA and as expected we find very good agreement when the rate parameters are consistent with the QEA. As previously for the single substrate mechanism, the distributions of A and B molecules display bimodality at intermediate times.

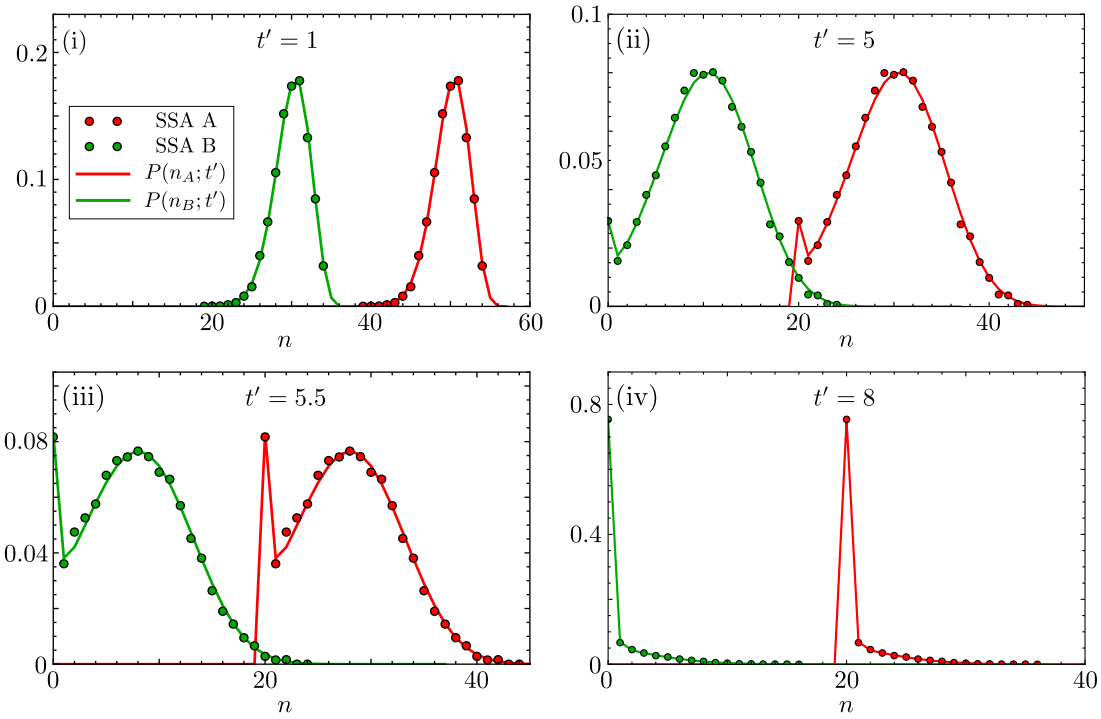

Figure 3.10: Comparison of the analytic distribution of two types of substrate species A and B, involved in the reaction mechanism (3.49), against the distributions obtained using the SSA. Note that SSA A and SSA B denote the SSA predictions for species A (red dots) and species B (green dots), respectively. In the panels (i)–(iv) we plot the probability distributions $P(n_A;t')$ (red line; from Eq. (3.57)) and $P(n_B;t')$ (green line; from Eq. (3.56)) for four different time points from near the initial condition (i) to near the absorbing state (iv) (time is non-dimensional as in previous figures). The initial number of substrate molecules are $N_A = 60$, $N_B = 40$ and the number of enzyme molecules is $M = 5$; the rates are $k_0/k_2 = k_1/k_2 = 10^3$ which enforce the QEA. The analytic distributions are in good agreement with the respective SSA distributions. Note that the absorbing point of A is $n_A = 20$ while that of B is $n_B = 0$; this is dictated by the difference between the initial number of substrate molecules $N_A - N_B = 20$. Each SSA probability distribution is constructed from 10^5 individual reaction trajectories.

3.7 Discussion

In summary, we have shown using averaging that in the limit of quasi-equilibrium between substrate and the enzyme, it is possible to reduce the two variable stochastic description of the MM reaction to that of an effective one variable master equation describing the slow transitions between groups of microstates. This master equation is subsequently solved exactly, using methods from linear algebra and complex analysis, to obtain closed-form solutions for the time-dependent marginal distributions of substrate and enzyme numbers. We have shown theoretically, and verified by means of stochastic simulations, that the solutions for the time-dependent marginal distributions are accurate for all times, provided the probability of complex decay into substrate and enzyme is much larger than the probability of complex decay into product and enzyme. To our knowledge, this is the first systematically derived approximate closed-form solution for the MM reaction for an arbitrary initial number of substrate and enzyme molecules; previous work treated a similar problem but using a heuristic approach [65] or derived closed-form solutions for the case of a single enzyme molecule [132, 225] or else considered reactions with multiple enzyme molecules focusing on deriving expressions for the turnover rate [69, 189, 233]. We have also shown how the same procedure can be used to obtain the solution of more complex enzyme mechanisms such as those involving the catalysis of multiple types of substrate by the same enzyme species.

For the MM reaction, we have compared our closed-form solution with that obtained by the solution of the CME reduced by means of the widely used discrete stochastic MM approximation [62], where the propensity for substrate decay has a hyperbolic dependence on the number of substrate molecules. If the initial substrate number N is not much larger than the total enzyme number M, but the rate constants satisfy the inequality $k_1 \gg k_2$, then the enzyme numbers fluctuations can be large, even though the rapid equilibrium approximation is valid. In this case, we show that the distribution predicted by the CME reduced using the discrete stochastic MM approximation is significantly different than the one obtained from stochastic simulations, whereas the solution provided by our theory accurately matches the simulations.

Using the closed-form solution for the time-dependent marginal probability distribution for substrate number, we have found that unexpectedly for a delta function (unimodal) initial condition, the distribution of substrate numbers can be bimodal at intermediate times, if the initial number of substrate molecules is significantly larger than the total number of enzyme molecules and provided the rate of complex decay into substrate and enzyme is much less than the rate of substrate and enzyme binding. We note that the latter rate in the CME formulation is inversely proportional to the compartment volume (since the encounter rate of two molecules decreases with increasing volume [127]), and hence our results imply that in the limit of small volumes (taken at constant initial number of substrate and enzyme molecules), bimodality of the distribution of substrate molecules is observable. This result is of particular relevance to understanding enzyme dynamics inside cells where the volume is very small. Our system with the initial conditions used, can then be interpreted as modelling the enzyme-mediated decay of substrate molecules, following the production (via translation) of a short burst of substrate molecules N at time $t = 0$, provided there is not another burst of substrate expression before the

substrate decays; these conditions are common for many cells where protein production occurs sporadically in bursts of short duration [248, 249]. We emphasise that the presence of transient bimodality in the MM reaction system is particularly interesting since it has no deterministic counterpart.

Chapter 4

MomentClosure.jl: automated moment closure approximations in Julia

The contents of this chapter are adapted from the application note [2] entitled *MomentClosure.jl: automated moment closure approximations in Julia* (published in *Bioinformatics*) and its documentation accessible online on GitHub, which includes a comprehensive review of moment closure approximations and tutorials on using the software package.

4.1 Abstract

MomentClosure.jl is a Julia package providing automated derivation of the time-evolution equations of the moments of molecule numbers for virtually any chemical reaction network using a wide range of moment closure approximations. It extends the capabilities of modelling stochastic biochemical systems in Julia and can be particularly useful when exact analytic solutions of the chemical master equation are unavailable and when Monte Carlo simulations are computationally expensive. MomentClosure.jl is freely accessible under the MIT license. Source code and documentation are available at https://github.com/augustinas1/MomentClosure.jl.

4.2 Introduction

The stochastic dynamics of biochemical systems under well-mixed conditions are governed by the chemical master equation (CME). The CME cannot be solved analytically except for simple systems and its exact stochastic simulation [152] can be computationally expensive, in turn motivating the development of more efficient approximation methods [125].

One approach is to approximate the whole probability distribution solution of the CME in terms of its first few moments: starting from the CME, we can derive a set of ordinary differential equations (ODEs) describing the time-evolution of moments for the molecule numbers of each species in a system, e.g., means and (co)variances. However, if a chemical reaction network is non-linear, we end up with an infinite hierarchy of coupled moment equations that cannot be solved directly as each moment will depend on higher order moments. Nevertheless, this problem can be tackled using one of many moment closure approximations (MAs) that express all moments above a certain order in terms of lower order moments using various (usually distributional) assumptions, effectively closing the hierarchy and enabling a numerical solution [180, 181, 250].

Deriving the moment equations and applying MAs manually can be a cumbersome and error prone process, especially when large systems and high-order MAs are considered. For this reason, a number of software tools have been developed over the years, allowing automatic derivation of the closed moment equations for a specified chemical reaction network [180, 251–255]. However, these packages are either outdated and unmaintained, offer limited functionality (implementing only few types of MAs and restricted to mass action reactions) or require proprietary expensive software, limiting the potential user base.

We present MomentClosure.jl, the first Julia [204] package to automatically derive the closed time-evolution equations of moments up to an arbitrary order for any chemical reaction network supporting both non-polynomial and time-dependent propensity functions, and implementing a variety of MAs commonly used in stochastic biochemical kinetics. Julia's high performance, flexibility and ease of use make it an attractive programming language for biologists [205], hence we believe that MomentClosure.jl is a timely addition to Julia's package ecosystem that further extends its capabilities of modelling and analysing chemical reaction networks.

The structure of this chapter is as follows. In Section 4.3 we overview the relevant background theory, namely: (1) the method of moment expansion and its generalisation applicable to chemical reaction networks with non-polynomial propensity functions; (2) a variety of moment closure schemes commonly used in literature and implemented in MomentClosure; (3) the novel methodology behind the linear mapping approximation [182] that is also supported in the package. In Section 4.4 we describe the implementation details of MomentClosure and its integration within the Julia modelling ecosystem. In Section 4.5 we present a simple example demonstrating the usage of MomentClosure, where we construct and solve the moment equations for the Brusselator. We conclude the chapter in Section 4.6, touching upon the limitations of moment closure approximations in general and discussing some of the potential future additions to the package.

4.3 Background

4.3.1 Moment expansion

In this section, we present a derivation of the raw moment equations starting from the general-form CME given by Eq. (2.2) for any chemical reaction network with polynomial propensity functions. Next, we discuss a more general method applicable to models with non-polynomial propensities, based on Taylor-expanding the CME in terms of its central moments about the mean. In both cases, we arrive at computationally convenient expressions for the moment expansion up to arbitrary order, which are implemented in MomentClosure. We note that MomentClosure also supports moment expansion for continuous-state models described by stochastic differential equations (SDEs). Although our focus here is the moment expansion for discrete CME-based models, the moment equations for systems of SDEs can be derived starting from Itô's lemma, as described in [256] and [257] (more details can be found in the online documentation).

Raw moment equations

From the CME given by Eq. (2.2) we can obtain a system of ODEs governing the time-evolution of the *raw* moments of the system up to specified *moment expansion order* m, given by $\mu_{\boldsymbol{i}} = \mu_{i_1,\ldots,i_N} = \langle n_1^{i_1} \cdots n_N^{i_N} \rangle$, where the indices $\boldsymbol{i} = (i_1, \ldots, i_N)$ are restricted such that $|\boldsymbol{i}| = \sum_{j=1}^{N} i_j \leq m$. Note that the first order ($|\boldsymbol{i}| = 1$) raw moments are simply the means. For example, when N = 3, the mean molecule number of second species in the system is given by $\mu_2 = \mu_{0,1,0} = \langle n_2 \rangle$. We choose to relax the notation throughout so that the means can be indicated by a single index of the molecular species (used in this section) or by the corresponding one-hot vector (as implemented in the source code).

To find the first moment equations, we multiply the CME by n_i and sum over all possible states:

$$\begin{aligned}\sum_{\boldsymbol{n}} n_i \frac{dP(\boldsymbol{n},t)}{dt} &= \sum_{n_1}^{\infty}\sum_{n_2}^{\infty}\cdots\sum_{n_N}^{\infty} n_i \frac{dP(\boldsymbol{n},t)}{dt} \\ &= \sum_{r}\sum_{\boldsymbol{n}} n_i a_r(\boldsymbol{n}-\boldsymbol{S}_r)P(\boldsymbol{n}-\boldsymbol{S}_r,t) - n_i a_r(\boldsymbol{n})P(\boldsymbol{n},t)\ .\end{aligned}$$

Applying a transformation $\boldsymbol{n} - \boldsymbol{S}_r \to \boldsymbol{n}$ on the first term in the sum leads to

$$\begin{aligned}\frac{d\mu_i}{dt} &= \sum_{r}\sum_{\boldsymbol{n}} (n_i + \boldsymbol{S}_{ir}) a_r(\boldsymbol{n})P(\boldsymbol{n},t) - n_i a_r(\boldsymbol{n})P(\boldsymbol{n},t) \\ &= \sum_{r}\sum_{\boldsymbol{n}} \boldsymbol{S}_{ir} a_r(\boldsymbol{n})P(\boldsymbol{n},t) \\ &= \sum_{r} \boldsymbol{S}_{ir}\langle a_r(\boldsymbol{n})\rangle\ .\end{aligned} \tag{4.1}$$

The derivation can be extended to the multivariate case of higher order moments $\mu_{\boldsymbol{i}}$ with $|\boldsymbol{i}| \leq m$:

$$\begin{aligned}
\frac{d\mu_{\boldsymbol{i}}}{dt} &= \sum_{\boldsymbol{n}} n_1^{i_1} \cdots n_N^{i_N} \frac{dP(\boldsymbol{n},t)}{dt} \\
&= \sum_r \sum_{\boldsymbol{n}} \left[(n_1 + \boldsymbol{S}_{1r})^{i_1} \cdots (n_N + \boldsymbol{S}_{2r})^{i_N} - n_1^{i_1} \cdots n_N^{i_N} \right] a_r(\boldsymbol{n}) P(\boldsymbol{n},t) \\
&= \sum_r \sum_{\boldsymbol{n}} \left[\sum_{j_1=0}^{i_1} \binom{i_1}{j_1} n_1^{j_1} \boldsymbol{S}_{1r}^{i_1-j_1} \cdots \sum_{j_N=0}^{i_N} \binom{i_N}{j_N} n_N^{j_N} \boldsymbol{S}_{Nr}^{i_N-j_N} - n_1^{i_1} \cdots n_N^{i_N} \right] a_r(\boldsymbol{n}) P(\boldsymbol{n},t) \\
&= \sum_r \sum_{\boldsymbol{n}} \sum_{|\boldsymbol{j}|=0}^{|\boldsymbol{i}|-1} \binom{i_1}{j_1} \cdots \binom{i_N}{j_N} \boldsymbol{S}_{1r}^{i_1-j_1} \cdots \boldsymbol{S}_{Nr}^{i_N-j_N} n_1^{j_1} \cdots n_N^{j_N} a_r(\boldsymbol{n}) P(\boldsymbol{n},t) \\
&= \sum_r \sum_{|\boldsymbol{j}|=0}^{|\boldsymbol{i}|-1} \binom{\boldsymbol{i}}{\boldsymbol{j}} \boldsymbol{S}_r^{\boldsymbol{i}-\boldsymbol{j}} \langle \boldsymbol{n}^{\boldsymbol{j}} a_r(\boldsymbol{n}) \rangle \,,
\end{aligned} \tag{4.2}$$

where we have introduced multi-index notation:

$$|\boldsymbol{i}| = \sum_{j=1}^{N} i_j, \tag{4.3}$$

$$\sum_{|\boldsymbol{j}|=0}^{|\boldsymbol{i}|-1} = \sum_{\substack{0 \leq j_1 \leq i_1, \ldots, 0 \leq j_N \leq i_N \\ 0 \leq j_1 + \cdots + j_N \leq |\boldsymbol{i}|-1}} \tag{4.4}$$

$$\binom{\boldsymbol{i}}{\boldsymbol{j}} = \binom{i_1}{j_1} \cdots \binom{i_N}{j_N} \tag{4.5}$$

$$\boldsymbol{n}^{\boldsymbol{j}} = n_1^{j_1} \cdots n_N^{j_N}, \tag{4.6}$$

$$\boldsymbol{S}_r^{\boldsymbol{i}-\boldsymbol{j}} = \boldsymbol{S}_{1r}^{i_1-j_1} \cdots \boldsymbol{S}_{Nr}^{i_N-j_N} \,. \tag{4.7}$$

We stress three key aspects:

- Throughout all derivations presented in this section we assume that the components of the net stoichiometry matrix S are constant values. However, in certain cases the reaction product may itself be a stochastic variable so that the corresponding expectation values, $\langle \boldsymbol{S}_r^{\boldsymbol{i}-\boldsymbol{j}} \rangle$, must be taken into account. For example, models of this type are encountered in the context of gene expression, where, under the assumption of fast mRNA decay, an autoregulatory genetic feedback loop produces proteins in bursts that follow a geometric burst size distribution [116, 250]. Support for such biochemical systems where the components of S can be independent geometrically distributed variables is implemented in MomentClosure.jl—a tutorial detailing this software functionality is provided online.
- Generation of raw moment equations is only possible if the kinetics of the system at hand are governed by the law of mass action where *all* propensity functions are *polynomials* in $\boldsymbol{n}$, otherwise the expectation value terms $\langle a_r(\boldsymbol{n}) \rangle$ and $\langle n_{i_1}^{j_1} \cdots n_{i_N}^{j_N} a_r(\boldsymbol{n}) \rangle$ are ill-defined. This key issue can be overcome by the general central moment expansion method presented below.

- The order of the polynomials $a_r(\boldsymbol{n})$ determines the order of moments encountered in the generated system of ODEs. If the system is *linear* (contains only zeroth and first order reactions), the m^{th} order moments will depend only on the moments of order m or lower, hence constituting a *finite* hierarchy of moment ODEs that can be readily solved numerically or otherwise without any approximations. However, if the system is *non-linear* (involves second or higher order reactions), moment equations will depend on higher order moments. For example, if the reaction network contains bimolecular reactions, the corresponding propensity functions will be second order polynomials and hence m^{th} order moment equations will now depend on $(m+1)^{th}$ order moments. This leads to an infinite hierarchy of coupled moment equations where each moment will depend on higher order moments—it cannot be solved directly and needs to be truncated. This can be achieved using one of many *moment closure approximations* that express $(m+1)^{\text{th}}$ order moments in terms of m^{th} and lower order moments using different distributional assumptions, effectively closing the hierarchy and enabling one to solve the moment equations up to m^{th} order [125].Details of all closure methods currently implemented within the published package MomentClosure.jl can be found in Section 4.3.2.

Central moment equations

As noted in the previous section, the moment equations of $P(\boldsymbol{n}, t)$ can be obtained in a straightforward manner if the kinetics of the system are governed by the law of mass action where all propensity functions are polynomials in $\boldsymbol{n}$ [125]. Similarly, given all propensities are rational functions, a polynomial form can also be recovered [258]. However, problems arise when the propensities are more complicated non-polynomial functions. This can nevertheless be overcome by considering a more general method of moment expansion that enables us to obtain mean and central moment equations up to arbitrary order for virtually any chemical reaction network with any type of *smooth* (infinitely differentiable) propensity functions. Such framework was first independently formulated by Lee [259] and Ale et al. [179]—the derivation provided below closely follows these works.

We start by Taylor-expanding the propensity functions around the mean $\boldsymbol{\mu} = \langle \boldsymbol{n} \rangle = (\mu_1, \ldots, \mu_N)$. This allows us to consider any general propensity function under the assumption that it is infinitely differentiable. The expansion leads to

$$\begin{aligned} a_r(\boldsymbol{n}) &= a_r(\boldsymbol{\mu}) + \sum_{j_1}^{N} \frac{\partial a_r(\boldsymbol{\mu})}{\partial n_{j_1}} (n_{j_1} - \mu_{j_1}) \\ &\quad + \frac{1}{2!} \sum_{j_1, j_2} \frac{\partial^2 a_r(\boldsymbol{\mu})}{\partial n_{j_1} \partial n_{j_2}} (n_{j_1} - \mu_{j_1})(n_{j_2} - \mu_{j_2}) + \cdots \\ &\quad + \frac{1}{q!} \sum_{j_1, j_2, \ldots, j_q} \frac{\partial^q a_r(\boldsymbol{\mu})}{\partial n_{j_1} \partial n_{j_2} \cdots \partial n_{j_q}} (n_{j_1} - \mu_{j_1})(n_{j_2} - \mu_{j_2}) \cdots (n_{j_q} - \mu_{j_q}) + \cdots \\ &= \sum_{|\boldsymbol{j}|=0}^{q} \frac{1}{\boldsymbol{j}!} D^{\boldsymbol{j}} a_r(\boldsymbol{\mu}) (\boldsymbol{n} - \boldsymbol{\mu})^{\boldsymbol{j}} + \cdots , \end{aligned}$$

where q controls the expansion order and we have simplified the expression by using Eqs. (4.3-4.7) together with:

$$\begin{aligned}\boldsymbol{j}! &= j_1! \cdots j_N!, \\ D^{\boldsymbol{j}} f &= \frac{\partial^{|\boldsymbol{j}|}}{\partial n_1^{j_1} \cdots \partial n_N^{j_N}} f \ .\end{aligned}$$

We can now obtain the equations governing the time-evolution of the means μ_i and the central moments $M_{\boldsymbol{i}} = M_{i_1,\ldots,i_N} = \langle (n_1 - \mu_1)^{i_1} \cdots (n_N - \mu_N)^{i_N} \rangle$, where again $|\boldsymbol{i}| \leq m$. We first consider the equations for the means by immediately starting from Eq. (4.1):

$$\begin{aligned}\frac{d\mu_i}{dt} &= \sum_r \boldsymbol{S}_{ir} \langle a_r(\boldsymbol{n}) \rangle \\ &= \sum_r \boldsymbol{S}_{ir} \Big(\sum_{|\boldsymbol{j}|=0}^{q} \frac{1}{\boldsymbol{j}!} D^{\boldsymbol{j}} a_r(\boldsymbol{\mu}) \langle (\boldsymbol{n} - \boldsymbol{\mu})^{\boldsymbol{j}} \rangle + \cdots \Big) \\ &= \sum_r \boldsymbol{S}_{ir} \Big(\sum_{|\boldsymbol{j}|=0}^{q} \frac{1}{\boldsymbol{j}!} D^{\boldsymbol{j}} a_r(\boldsymbol{\mu}) M_{\boldsymbol{j}} + \cdots \Big) \ .\end{aligned}$$

We continue to derive the equations for central moments by taking the time derivative of $M_{\boldsymbol{i}}$ and using the CME as formulated in Eq. (2.2):

$$\begin{aligned}\frac{dM_{\boldsymbol{i}}}{dt} &= \sum_{\boldsymbol{n}} (n_1 - \mu_1)^{i_1} \cdots (n_N - \mu_N)^{i_N} \frac{dP(\boldsymbol{n},t)}{dt} \\ &- \sum_{\boldsymbol{n}} \sum_{j=1}^{N} i_j \frac{d\mu_j}{dt} (n_1 - \mu_1)^{i_1} \cdots (n_j - \mu_j)^{i_j - 1} \cdots (n_N - \mu_N)^{i_N} P(\boldsymbol{n},t) \\ &= \sum_r \sum_{\boldsymbol{n}} \Big[(n_1 - \mu_1 + \boldsymbol{S}_{1r})^{i_1} \cdots (n_N - \mu_N + \boldsymbol{S}_{Nr})^{i_N} \\ &- (n_1 - \mu_1)^{i_1} \cdots (n_N - \mu_N)^{i_N} \Big] a_r(\boldsymbol{n}) P(\boldsymbol{n},t) - \sum_j i_j \frac{d\mu_j}{dt} M_{i_1,\ldots,i_j - 1,\ldots,i_N} \\ &= \sum_r \sum_{\boldsymbol{n}} \left[\sum_{|\boldsymbol{j}|=0}^{|\boldsymbol{i}|-1} \binom{i_1}{j_1} \cdots \binom{i_N}{j_N} \boldsymbol{S}_{1r}^{i_1 - j_1} \cdots \boldsymbol{S}_{Nr}^{i_N - j_N} (n_1 - \mu_1)^{j_1} \cdots (n_N - \mu_N)^{j_N} \right] \\ &\times \left[\sum_{|\boldsymbol{j}|=0}^{q} \frac{1}{\boldsymbol{j}!} D^{\boldsymbol{j}} a_r(\boldsymbol{\mu}) (\boldsymbol{n} - \boldsymbol{\mu})^{\boldsymbol{j}} + \cdots \right] - \sum_j i_j \frac{d\mu_j}{dt} M_{\boldsymbol{i}_{j-}} \\ &= \sum_r \sum_{|\boldsymbol{j}|=0}^{|\boldsymbol{i}|-1} \left[\binom{i_1}{j_1} \cdots \binom{i_N}{j_N} \boldsymbol{S}_{1r}^{i_1 - j_1} \cdots \boldsymbol{S}_{Nr}^{i_N - j_N} \sum_{|\boldsymbol{k}|=0}^{q-|\boldsymbol{j}|} \frac{1}{\boldsymbol{k}!} D^{\boldsymbol{k}} a_r(\boldsymbol{\mu}) M_{\boldsymbol{j}+\boldsymbol{k}} + \cdots \right] \\ &- \sum_j i_j \frac{d\mu_j}{dt} M_{\boldsymbol{i}_{j-}} \\ &= \sum_r \sum_{|\boldsymbol{j}|=0}^{|\boldsymbol{i}|-1} \left[\binom{\boldsymbol{i}}{\boldsymbol{j}} \boldsymbol{S}_r^{\boldsymbol{i}-\boldsymbol{j}} \sum_{|\boldsymbol{k}|=0}^{q-|\boldsymbol{j}|} \frac{1}{\boldsymbol{k}!} D^{\boldsymbol{k}} a_r(\boldsymbol{\mu}) M_{\boldsymbol{j}+\boldsymbol{k}} + \cdots \right] - \sum_j i_j \frac{d\mu_j}{dt} M_{\boldsymbol{i}_{j-}} \ ,\end{aligned}$$

where we have also defined

$$M_{\boldsymbol{i}_{j-}} = M_{i_1,\ldots,i_j-1,\ldots,i_N},$$
$$M_{\boldsymbol{j}+\boldsymbol{k}} = M_{j_1+k_1,\ldots,j_N+k_N}\ .$$

Although raw moment equations for non-linear mass-action systems already require approximate treatment using moment closure, here we have an additional complication: if a system contains *non-polynomial* propensity functions, the equations for both means and central moments will in principle depend on an *infinite* number of higher order central moments. Hence the *Taylor expansion order* q is of utmost importance as it controls the degree of approximation: the propensity functions are expanded up to q^{th} order so that m^{th}-order central moment equations will depend on central moments of order $q > m$ and lower. Finally, moment closure approximations can be applied similarly as in the case of raw moment equations.

4.3.2 Moment closure approximations

In Section 4.3.1, we have shown that for a non-linear system an infinite hierarchy of coupled moment equations is obtained that cannot be solved directly and, therefore, needs to be truncated in an approximate way. This can be achieved using moment closure approximations (MAs), in which all moments above a certain order m are expressed in terms of m^{th} and lower order moments using various (usually distributional) assumptions [125]. Doing so enables us to effectively *close* the moment hierarchy, leading to a finite set of ODEs which can then be solved numerically. In this section, we present some the commonly used MA methods that are implemented in MomentClosure. A simple application of MAs using the software is provided in Section 4.5 and more tutorials showcasing different MAs can be found in the online documentation.

Zero closure

The simplest MA is the "central-moment-neglect" MA (CMN-MA) [180], also referred to as "zero-closure" [181] or "low dispersion moment closure" [177], where CMN-MA at m^{th} order means that the moment equations are truncated by setting all *central moments* above order m to zero. For example, in the simple case of 2nd order truncation, the moment equations for the means μ_i and covariances C_{ij} become:

$$\begin{aligned}
\frac{d\mu_i}{dt} &= \sum_r \boldsymbol{S}_{ir}\Big(a_r(\boldsymbol{\mu}) + \frac{1}{2}\sum_{i_1,i_2}\frac{\partial^2 a_r(\boldsymbol{\mu})}{\partial n_{i_1}\partial n_{i_2}}M_{i_1,i_2}\Big),\\
\frac{dC_{ij}}{dt} &= \frac{dM_{\mathbf{0}_{i+,j+}}}{dt}\\
&= \frac{d\langle(n_i-\mu_i)(n_j-\mu_j)\rangle}{dt} =\\
&= \sum_r\Big[\boldsymbol{S}_{ir}\sum_k\frac{\partial a_r(\boldsymbol{\mu})}{\partial n_k}C_{jk} + \boldsymbol{S}_{jr}\sum_k\frac{\partial a_r(\boldsymbol{\mu})}{\partial n_k}C_{ik}\\
&+ \boldsymbol{S}_{ir}\boldsymbol{S}_{jr}\Big(a_r(\boldsymbol{\mu}) + \frac{1}{2}\sum_{k,l}\frac{\partial^2 a_r(\boldsymbol{\mu})}{\partial n_k\partial n_l}C_{kl}\Big)\Big]\ .
\end{aligned}$$

Normal closure

Another popular MA is the "normal moment-closure", pioneered by Goodman [174] and Whittle [175], where all *cumulants* $\kappa_{\boldsymbol{i}}$ above order m are set to zero, approximating the probability distribution of the system with the normal distribution [180]:

$$\kappa_{\boldsymbol{i}} = 0, \quad \text{for } |\boldsymbol{i}| > m.$$

In order to truncate the higher order central or raw moments $M_{\boldsymbol{i}}$ using normal closure, we can express them in terms of cumulants $\kappa_{\boldsymbol{i}}$ using the multivariate moment and cumulant relationships formalised by Balakrishnan et al. [260]. We use the normal MA in Section 4.5 to close the moment equations of the Brusselator model, where we demonstrate the basic functionality of MomentClosure.

Note that a different implementation of normal closure can be found in literature [181], where the higher order central moments are expressed in terms of a sum of product of covariances using Isserlis' theorem [261]. However, one could argue that such formulation is not advisable as it assumes stronger "Gaussianity" of the underlying distribution than setting the higher order cumulants to zero which is less of an approximation on the form of the distribution and hence can be considered to be preferable in the development of MAs. For example, in case we are truncating the moment equations at the 4th order, the truncation-order central moments would be expressed only in terms of covariances whereas our formulation using cumulants would explicitly include information about the computed values of third central moments, which is expected to improve numerical stability and lead to more accurate moment estimates.

Poisson closure

Although the Poisson distribution lacks a general formulation for multiple variables [181], "Poisson MA" has been formulated [176, 180] assuming that the joint multivariate distribution is a product of univariate Poisson distributions, i.e., $n_i \sim \text{Poisson}(\mu_i)$. The cumulants of a univariate Poisson distribution are equal to the mean, hence in Poisson closure we set all the higher order diagonal cumulants to the corresponding mean values and the mixed cumulants to zero [180], which in our notation can be expressed as:

$$\begin{aligned}
&\kappa_{\boldsymbol{i}} = \mu_j, \quad \text{if } i_j > m \text{ and } i_{j\neq k} = 0 \quad \text{for some } j,k \in \{1,\dots,N\},\\
&\kappa_{\boldsymbol{i}} = 0, \quad \text{if } |\boldsymbol{i}| > m \text{ and } i_j \neq i_k \text{ (where } i_j \neq 0, i_k \neq 0) \quad \text{for some } j,k \in \{1,\dots,N\}
\end{aligned}$$

Similarly to normal closure, the higher order central or raw moments can be expressed in terms of cumulants as described in [260].

Log-normal closure

"Log-normal" MA, first applied by Keeling [262], allows one to truncate the moment equations under the assumption that the distribution of the underlying stochastic process is log-normal. A positive multi-dimensional random variable $\boldsymbol{n}$ follows a log-normal distribution if its logarithm is normally distributed, so that $\boldsymbol{y} = \ln \boldsymbol{n}$, and $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\nu}, \Sigma)$, where $\boldsymbol{\nu}$ and Σ denote the vector of means and the covariance matrix respectively. By considering the moment generating function of the normal distribution, $\mathcal{N}(\boldsymbol{\nu}, \Sigma)$, one can show that the raw moments are given by [181, 263]:

$$\mu_{\boldsymbol{i}} = \exp\left(\boldsymbol{i}^\top \boldsymbol{\nu} + \frac{1}{2}\boldsymbol{i}^\top \Sigma \boldsymbol{i}\right).$$

It follows that

$$\begin{aligned}
\nu_i &= \ln \mu_i - \frac{1}{2}\Sigma_{ii}, \\
\Sigma_{ij} &= \ln\left(1 + \frac{\langle (n_i - \mu_i)(n_j - \mu_j)\rangle}{\exp\left(\nu_i + \nu_j + \frac{1}{2}\left(\Sigma_{ii} + \Sigma_{jj}\right)\right)}\right) \\
&= \ln\left(1 + \frac{C_{ij}}{\exp\left(\nu_i + \nu_j + \frac{1}{2}\left(\Sigma_{ii} + \Sigma_{jj}\right)\right)}\right), \\
\Sigma_{ii} &= \ln\left(1 + \frac{C_{ii}}{\mu_i^2}\right).
\end{aligned}$$

Note that central moments can be obtained from raw moments by utilising their general multivariate relationship [264].

Gamma closure

The method of "gamma closure" was originally implemented by Lakatos et al. [181], where the authors acknowledged the ambiguity arising in defining multivariate gamma distributions, and, building upon previous definitions in the literature (e.g. [265] and [266]), proposed a new formulation of a multivariate gamma distribution. Here we reproduce the definition by closely following the description in [181] and elucidating some of the derivation steps.

We denote a random variable drawn from gamma distribution with shape α and scale β as $n \sim \text{Gamma}(\alpha, \beta)$. The probability density function of the univariate gamma distribution with the corresponding shape-scale parameterisation is

$$f(n; \alpha, \beta) = \frac{1}{\Gamma(\alpha)\beta^\alpha} n^{\alpha-1} e^{-\frac{n}{\beta}},$$

where Γ is the gamma function. The i^{th} raw moment of n is given by

$$\mu_i = \frac{\Gamma(\alpha + i)\beta^i}{\Gamma(\alpha)} = (\alpha)_i \beta^i, \tag{4.8}$$

where $(\alpha)_i = \alpha(\alpha + 1)\cdots(\alpha + i - 1)$. Note that the moment generating function of X is

$$G_n(k) = \langle e^{kn} \rangle = (1 - \beta k)^{-\alpha}.$$

In order to construct a multivariate gamma distribution, we start by considering independent gamma variables Y_{kl}, $k, l = 1, \ldots, N$, with shape and scale parameters α_{kl} and β_{kl} respectively. Here we define Y_{kl} to be symmetric in indices, i.e., $Y_{kl} = Y_{lk}$. Now consider an N-dimensional random vector $\boldsymbol{n} = (n_1, n_2, \ldots, n_N)$, where n_i is a linear combination of independent gamma variables:

$$n_i = \sum_{j=1}^{N} \frac{\beta_{ii}}{\beta_{ij}} Y_{ij}.$$

The i^{th} marginal moment generating function of the joint distribution of $\boldsymbol{n}$ is given by:

$$\begin{aligned} G_{n_i}(k_i) &= \left\langle e^{k_i \sum_{j=1}^{N} \frac{\beta_{ii}}{\beta_{ij}}} \right\rangle \\ &= G_{Y_{i1}}\left(k_i \frac{\beta_{ii}}{\beta_{i1}}\right) G_{Y_{i2}}\left(k_i \frac{\beta_{ii}}{\beta_{i2}}\right) \cdots G_{Y_{iN}}\left(k_i \frac{\beta_{ii}}{\beta_{iN}}\right) \\ &= (1 - \beta_{ii} k_i)^{\sum_{j=1}^{N} \alpha_{ij}}, \end{aligned}$$

so that $n_i \sim \text{Gamma}(\alpha_i, \beta_i)$, where $\alpha_i = \sum_{j=1}^{N} \alpha_{ij}$ and $\beta_i = \beta_{ii}$. Therefore, we have obtained an N-variate gamma distribution, which can be denoted as $\boldsymbol{n} \sim MG(\boldsymbol{\alpha}, \boldsymbol{\beta})$, where the vectors of shape and scale parameters are given by $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)$ and $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_N)$ respectively.

We can now readily obtain the raw m^{th} order moment of n_i:

$$\begin{aligned} \langle n_i^m \rangle &= \left\langle \left(\sum_{j=1}^{N} \frac{\beta_{ii}}{\beta_{ij}} Y_{ij} \right)^m \right\rangle \\ &= \left\langle \sum_{k_1 + k_2 + \cdots + k_N = m} \frac{m!}{k_1! k_2! \cdots k_N!} \prod_{j=1}^{N} \left(\frac{\beta_{ii}}{\beta_{ij}} Y_{ij} \right)^{k_j} \right\rangle \\ &= \sum_{|\boldsymbol{k}| = m} \frac{m!}{\boldsymbol{k!}} \prod_{j=1}^{N} \left\langle \left(\frac{\beta_{ii}}{\beta_{ij}} Y_{ij} \right)^{k_j} \right\rangle \\ &= \beta_i^m \sum_{|\boldsymbol{k}| = m} \frac{m!}{\boldsymbol{k!}} \left(\prod_{j=1}^{N} (\alpha_{ij})_{k_j} \right). \end{aligned}$$

Note that we have used Eq. (4.8) to get to the last line. The mixed raw moments are computed in a similar fashion:

$$\begin{aligned}
\mu_{\boldsymbol{i}} &= \left\langle \left(\sum_{j=1}^{N} \frac{\beta_{11}}{\beta_{1j}} Y_{1j} \right)^{i_1} \cdots \left(\sum_{j=1}^{N} \frac{\beta_{NN}}{\beta_{Nj}} Y_{Nj} \right)^{i_N} \right\rangle \\
&= \boldsymbol{\beta^i} \left\langle \sum_{|\boldsymbol{k_1}|=i_1} \cdots \sum_{|\boldsymbol{k_N}|=i_N} \frac{\boldsymbol{i!}}{\boldsymbol{k_1!} \cdots \boldsymbol{k_N!}} \prod_{j=1}^{N} \left(\frac{Y_{1j}}{\beta_{1j}} \right)^{k_{1j}} \cdots \prod_{j=1}^{N} \left(\frac{Y_{Nj}}{\beta_{Nj}} \right)^{k_{Nj}} \right\rangle \\
&= \boldsymbol{\beta^i} \left\langle \sum_{|\boldsymbol{k_1}|=i_1} \cdots \sum_{|\boldsymbol{k_N}|=i_N} \frac{\boldsymbol{i!}}{\boldsymbol{k_1!} \cdots \boldsymbol{k_N!}} \prod_{q=1}^{N} \left(\frac{Y_{qq}}{\beta_{qq}} \right)^{k_{qq}} \prod_{r=q+1}^{N} \left(\frac{Y_{qr}}{\beta_{qr}} \right)^{k_{qr}+k_{rq}} \right\rangle \\
&= \boldsymbol{\beta^i} \sum_{|\boldsymbol{k_1}|=i_1} \cdots \sum_{|\boldsymbol{k_N}|=i_N} \frac{\boldsymbol{i!}}{\boldsymbol{k_1!} \cdots \boldsymbol{k_N!}} \prod_{q=1}^{N} (\alpha_{qq})_{k_{qq}} \prod_{r=q+1}^{N} (\alpha_{qr})_{k_{qr}+k_{rq}},
\end{aligned}$$

where we have taken into account the symmetry in the indices and defined each $\boldsymbol{k_i}$ as an N-dimensional vector $\boldsymbol{k_i} = (k_{i_1}, \ldots, k_{i_N})$ and $\boldsymbol{\beta^i} = \beta_1^{i_1} \cdots \beta_N^{i_N}$. Note that the mean and variance of n_i can be obtained from:

$$\begin{aligned}
\mu_i &= \sum_{j=1}^{N} \alpha_{ij} \beta_i, \\
C_{ii} &= (\alpha_i)_2 \beta_i^2 - \alpha_i^2 \beta_i^2 \\
&= \sum_{j=1}^{N} \alpha_{ij} \beta_i^2.
\end{aligned} \tag{4.9}$$

Similarly, from Eq. (4.8) it follows that

$$\langle n_i n_j \rangle = \beta_i \beta_j \left(\sum_{\substack{k,l \\ (k,l) \neq (j,i)}}^{N} \alpha_{ik} \alpha_{jl} + (\alpha_{ij})_2 \right),$$

which together with Eq. (4.9) allows us to express the covariance as:

$$\begin{aligned}
C_{ij} &= \langle n_i n_j \rangle - \beta_i \beta_j \left(\sum_{k,l} \alpha_{ik} \alpha_{jl} \right) \\
&= \alpha_{ij} \beta_i \beta_j.
\end{aligned}$$

Finally, from the equations above we can obtain all shape and scale parameters:

$$\begin{aligned}
\beta_i &= \frac{C_{ii}}{\mu_i}, \\
\alpha_{ij} &= \frac{C_{ij}}{\beta_i \beta_j}, \\
\alpha_{ii} &= \frac{\mu_i}{\beta_i} - \sum_{\substack{k \\ k \neq i}} \alpha_{ik}.
\end{aligned}$$

Derivative matching

The derivative matching MA [267, 268] is based on expressing the moments above order m in terms of the lower order moments in such a way that their time derivatives match those of the *exact* moments at some initial time with the given initial conditions. We outline the approach below, closely following the complete exposition found in the original papers of Singh and Hespanha [267, 268].

The raw moment equations up to order m for any mass-action reaction network containing *at most* bimolecular (second order) reactions can be written down concisely in the matrix form:

$$\frac{d\boldsymbol{\mu}}{dt} = \hat{\boldsymbol{a}} + A\boldsymbol{\mu} + B\bar{\boldsymbol{\mu}} \ ,$$

where $\boldsymbol{\mu}$ is a vector containing all raw moments of the system up to order m and $\bar{\boldsymbol{\mu}}$ consists of all $(m+1)^{\text{th}}$ order raw moments which the equations depend on. The constant vector $\hat{\boldsymbol{a}}$ and constant matrices A and B are chosen appropriately for the system at hand. In this case, an MA can be defined as a procedure where each moment in $\bar{\boldsymbol{\mu}}$, $\bar{\mu}_{\boldsymbol{i}}$, is approximated by a *moment closure function* $\varphi_{\boldsymbol{i}}(\boldsymbol{\mu})$ of moments up to order m. Then the moment equations can be rewritten as

$$\frac{d\boldsymbol{\nu}}{dt} = \hat{\boldsymbol{a}} + A\boldsymbol{\nu} + B\bar{\boldsymbol{\phi}}(\boldsymbol{\nu}) \ ,$$

where the state of the system is now denoted by $\boldsymbol{\nu}$ instead of $\boldsymbol{\mu}$, stressing the fact that we are considering the approximation of the true moment dynamics, and $\bar{\boldsymbol{\phi}}(\boldsymbol{\nu})$ is the corresponding vector of moment closure functions.

The idea behind derivative matching is to determine a map $\bar{\boldsymbol{\phi}}$ so that the time derivatives between the exact moments, $\boldsymbol{\mu}(t)$, and the approximate moments, $\boldsymbol{\nu}(t)$, would match at some initial time t_0 under the initial condition $\boldsymbol{\mu}(t_0) = \boldsymbol{\nu}(t_0)$:

$$\left.\frac{d^i\boldsymbol{\mu}}{dt}\right|_{t=t_0} = \left.\frac{d^i\boldsymbol{\nu}}{dt}\right|_{t=t_0}$$

If these conditions hold, one can expect, based on a Taylor series approximation argument, that $\boldsymbol{\mu}(t)$ and $\boldsymbol{\nu}(t)$ will stay close at least locally in time and hence the MA will be sufficiently accurate.

In order to move forward with the derivation, Singh and Hespanha present what can be understood as essentially an ansatz. Firstly, moment closure functions for each $\boldsymbol{i}$, where $|\boldsymbol{i}| > m$, are chosen to have a separable form given by

$$\phi_{\boldsymbol{i}}(\boldsymbol{\mu}) = \prod_{1 \leq j_1 + \cdots + j_N \leq m} (\mu_{\boldsymbol{j}})^{\gamma_{\boldsymbol{j}}} = \prod_{|\boldsymbol{j}|=1}^{m} (\mu_{\boldsymbol{j}})^{\gamma_{\boldsymbol{j}}} \ ,$$

where $\gamma_{\boldsymbol{j}}$ are constants (unique for each vector $\boldsymbol{i}$) that can be determined by solving the following linear equation system:

$$C_{\boldsymbol{j}}^{\boldsymbol{i}} = \sum_{|\boldsymbol{k}|=1}^{m} \gamma_{\boldsymbol{k}} C_{\boldsymbol{j}}^{\boldsymbol{k}} \,, \quad \text{for each } \boldsymbol{j} \text{ where } |\boldsymbol{j}| \leq m \,,$$

where we have introduced multi-index scalars

$$C_{\boldsymbol{v}}^{\boldsymbol{u}} = C_{v_1}^{u_1} C_{v_2}^{u_2} \cdots C_{v_N}^{u_N} ,$$

with each element defined as

$$C_b^a = \begin{cases} \frac{a!}{(a-b)!b!}, & a \geq b \\ 0, & a < b \end{cases} .$$

Using the specific construction of $\bar{\boldsymbol{\phi}}$ described above, it can be shown [268] that for every *deterministic* initial condition, i.e., $\boldsymbol{n}(t_0) = \boldsymbol{n}_0$ with probability one, we will have

$$\begin{aligned} \boldsymbol{\mu}(t_0) = \boldsymbol{\nu}(t_0) \implies & \left.\frac{d\boldsymbol{\mu}}{dt}\right|_{t=t_0} = \left.\frac{d\boldsymbol{\nu}}{dt}\right|_{t=t_0} \\ \implies & \left.\frac{d^2\boldsymbol{\mu}}{dt^2}\right|_{t=t_0} = \left.\frac{d^2\boldsymbol{\nu}}{dt^2}\right|_{t=t_0} + \boldsymbol{\epsilon}(\boldsymbol{n}_0) \,, \end{aligned}$$

where all elements of $\boldsymbol{\epsilon}(\boldsymbol{n}_0)$ are zero except the ones corresponding to m^{th} order raw moments—these elements are second order polynomials in $\boldsymbol{n}_0$. Note, however, that these results hold only for mass-action systems containing no higher than second order chemical reactions. While the derivative matching MA can be applied in the same way to systems containing higher order polynomial and non-polynomial propensity functions, it has not been rigorously analysed in such scenarios, where, naturally, we may expect significantly larger approximation errors.

Conditional closures

As the standard MAs often fail to provide sufficiently accurate approximations of chemical reaction networks involving both high- and low-abundance species, some novel approaches suggest using the moments conditioned on the low-copy number species, which can result in a more effective description of the system dynamics [1, 16]. Here we discuss the conditional moment closure proposed by Soltani et al. [250], applicable to networks containing molecular species which copy number can be either zero or one, i.e., a binary/Bernoulli random variable. Such conditional MA can be very useful in the study of gene networks where two-state gene systems are often considered [182, 250]—the gene state itself can be treated as a distinct species which molecule number is a Bernoulli variable.

To illustrate the conditional MAs, consider a fragment of a simple two-state gene regulatory network where the gene promoter switches between the two states G and G^* upon binding/unbinding a protein molecule P in a nonlinear reversible reaction [182]:

$$G + P \underset{\sigma_u}{\overset{\sigma_b}{\rightleftharpoons}} G^*. \tag{4.10}$$

An example two-state gene circuit with such promoter switching dynamics is a negative autoregulative genetic feedback loop that takes into account the transcription of mRNA M and its subsequent translation into protein P, where the negative feedback is implemented via protein binding to the gene promoter, as fully described by the following set of reactions (also incorporating the degradation of both mRNA and protein):

$$G + P \underset{\sigma_u}{\overset{\sigma_b}{\rightleftharpoons}} G^*, \quad G \xrightarrow{k_m} M, \quad M \xrightarrow{k_p} P, \quad M \xrightarrow{d_m} \emptyset, \quad P \xrightarrow{d_p} \emptyset\,.$$

The schematic of the gene regulatory network is shown in Fig. 4.1, where we also use it as an example chemical reaction network for applying different MAs and comparing their accuracy. Furthermore, some other models of gene expression are described in detail in Chapter 5 (in a different context of neural network-based approximations), such as a a genetic toggle switch in Section 5.5.2 or a model of mRNA turnover in Secion 5.5.3. Note that the discussion in this section is only concerned with the nonlinear gene-switching reaction upon protein binding given by Eq. (4.10) and otherwise does not depend on any model specifics.

Considering Eq. (4.10), we denote the protein number (high-abundance species) in the two-state gene circuit by p and the binary gene state by g so that the unbound promoter state G corresponds to $g = 1$ and the bound state G to $g = 0$. The conditional MA is based on conditioning the higher order moments of high-abundance species on the binary species being in state 1 (instead of 0) and then applying standard MAs on the conditional moments. Firstly, note that g is a Bernoulli variable and hence the following identities hold

$$\begin{aligned}\langle g^j p\rangle &= \langle g\rangle, \quad j \in 2, 3, \ldots,\\ \langle g^j p^k\rangle &= \langle g p^k\rangle, \quad j, k \in 1, 2, 3, \ldots\end{aligned}$$

Therefore, we only need to concern ourselves with moments of the form $\langle p^k\rangle$ and $\langle g p^k\rangle$. The former can be approximated using the standard MAs (no conditioning needed), whereas the latter can be written down as:

$$\langle g p^j\rangle = \langle p^j \,|\, g = 1\rangle\langle g\rangle, \quad j \in \{1, 2, \ldots\}.$$

Now the conditional moment $\langle p^j \,|\, g = 1\rangle$ can be expressed in terms of lower order conditional moments using one of the standard MAs, e.g., normal closure or derivative matchingthe two methods (including the conditioning step) are respectively known as the *conditional gaussian* and *conditional derivative matching* MAs. In the following, for completeness, we show how the two methods are applied to approximate a specific higher order moment

$$\langle g p^3\rangle = \langle p^3 \,|\, g = 1\rangle\langle g\rangle.$$

Conditional Gaussian closure

Under the conditional Gaussian MA, we assume that the number of protein molecules conditioned on the gene being active, $p \,|\, g = 1$, follows a Gaussian distribution. In other words, we apply normal MA on the conditional moment $\langle p^j \,|\, g = 1\rangle$. Hence we obtain (following the example above):

$$\langle p^3 \,|\, g = 1\rangle = 3\langle p^2 \,|\, g = 1\rangle\langle p \,|\, g = 1\rangle - 2\langle p \,|\, g = 1\rangle^3.$$

Using $\langle gp^j\rangle = \langle p^j \,|\, g = 1\rangle\langle g\rangle$, we can rewrite the equation as:

$$\langle p^3 \,|\, g = 1\rangle = 3\frac{\langle gp^2\rangle\langle gp\rangle}{\langle g\rangle^2} - 2\frac{\langle gp\rangle^3}{\langle g\rangle^3}.$$

Plugging this into the previous expression of the higher-order moment $\langle gp^3\rangle$ we finally obtain:

$$\langle gp^3\rangle = 3\frac{\langle gp^2\rangle\langle gp\rangle}{\langle g\rangle} - 2\frac{\langle gp\rangle^3}{\langle g\rangle^2}.$$

Conditional derivative matching

The conditional derivative matching boils down to approximating the higher order conditional moments in terms of lower order conditional moments using the standard derivative matching (described above):

$$\langle p^3 \,|\, g = 1\rangle = \frac{\langle p^2 \,|\, g = 1\rangle^3}{\langle p \,|\, g = 1\rangle^3}.$$

Using $\langle gp^j\rangle = \langle p^j \,|\, g = 1\rangle\langle g\rangle$ again, we find:

$$\langle gp^3\rangle = \frac{\langle gp^2\rangle^3\langle g\rangle}{\langle gp\rangle^3}.$$

Note that conditional moment closure is fully applicable to systems containing multiple binary species, such as a repressor-activator circuit model described in [250] or a genetic toggle switch discussed in Section 5.5.2. For example, given two two-state genes, g_1 and g_2, the same Bernoulli variable properties hold and the higher-order moments can be expressed as:

$$\langle g_1 g_2 p^j\rangle = \langle p^j \,|\, g_1 = g_2 = 1\rangle\langle g_1 g_2\rangle, \quad j \in \{1, 2, \dots\}.$$

Such higher order conditional moments can again be closed using normal closure or derivative matching.

We emphasise that the description here follows [250] and we encourage the reader to see the paper for more details. Tutorial examples of applying conditional MAs on genetic feedback loops (reproducing some of the published results) can be found in the online documentation.

4.3.3 Linear Mapping Approximation

The Linear Mapping Approximation (LMA) provides a novel way of approximating the solution of the CME and has been shown to be accurate for a variety of models of gene regulatory networks (GRNs) [182]. It is based on mapping a *nonlinear* GRN onto an equivalent *linear* GRN so that the exact solution of the linear system gives an approximate solution of the nonlinear system. For example, the gene-protein binding reaction is replaced by an effective first-order reaction by approximating the protein number by its mean conditional on the gene state—the reaction propensity resulting from such mean-field approximation can be seen as optimal in the sense of minimising the Kullback-Leibler divergence between the full (nonlinear) model and the reduced (linear) model [269]. Below we provide a short overview of the LMA procedure and urge the reader to see the original paper [182] for a more comprehensive description.

We begin by considering a simple example of a two-state GRN exactly as described in Section 4.3.2 for conditional MAs, assuming that there is a single nonlinear reversible reaction in the network given by

$$G + P \underset{\sigma_u}{\overset{\sigma_b}{\rightleftharpoons}} G^*,$$

where P is the protein and the promoter can be either in state G ($g = 0$) or G^* ($g = 1$). Our aim is to find an approximate time-dependent probability distribution of protein numbers p at time t. The steps of applying the LMA can then be described as follows:

1. Find the linear network by replacing any reversible nonlinear reaction in the nonlinear network by a reversible pseudo first-order reaction between the binary species' states. In our example, we replace the reaction above with $G \underset{\sigma_u}{\overset{\bar{\sigma}_b}{\rightleftharpoons}} G^*$, noting that the rate parameter is changed from σ_b to $\bar{\sigma}_b$. Moreover, this approach is just as applicable in the case of cooperativity, e.g., $G + nP \rightleftharpoons G^*$ would be similarly replaced with $G \rightleftharpoons G^*$, where where n is an integer indicating the cooperative order. The LMA is restricted in its applicability to chemical reaction networks where *only one* of the substrates in each nonlinear reaction is a molecular species which copy number can be either zero or one (a Bernoulli random variable). While a chemical reaction network is allowed to have an arbitrary number of such species, more than one of them cannot be involved in any nonlinear reaction.
2. Approximate the changed rate parameters of the linearised reactions by their expectation values. As noted in [182], the first-order reaction $G \overset{\bar{\sigma}_b}{\to} G^*$ maps onto the second-order reaction $G + P \overset{\sigma_b}{\to} G^*$ if we choose $\bar{\sigma}_b = \sigma_b \langle p \,|\, g = 1 \rangle$, where $p \,|\, g = 1$ indicates the instantaneous protein number given the gene is in the state G. In LMA, we use the mean-field approximation taking the expectation value of the rate so that

$$\bar{\sigma}_b = \sigma_b \langle p \,|\, g = 1 \rangle = \sigma_b \frac{\langle pg \rangle}{\langle g \rangle}.$$

   The same procedure can be extended to the general nonlinear reaction where n proteins bind cooperatively. The effective parameter is then given by

$$\bar{\sigma}_b = \sigma_b \frac{\left\langle \prod_{i=0}^{n-1} (p - i)\, g \right\rangle}{\langle g \rangle}.$$

3. Write down the moment equations for the linear network using the approximated stochastic rates. Note that the moment equations must be generated up to the order given by the highest order nonlinear reaction in the network. If the only nonlinear reaction is the second-order reaction $G + P \xrightarrow{\sigma_b} G^*$, we need to consider only moments up to the second order (as hinted by the functional form of $\bar{\sigma}_b$ above). However, as the moment hierarchy is otherwise closed, no additional moment closure approximations need to be performed and therefore we can solve the moment equations in a straightforward manner.
4. Solve the moment ODEs numerically up to the time t and plug in the resulting moment values into the equations for the effective parameters. Then proceed to calculate the time-average of these parameters over the time-interval $[0, t]$.
   In our example, plugging the solved-for moment values into the equation for $\bar{\sigma}_b$ allows us to interpret the effective parameter as a time-dependent function $\bar{\sigma}_b(t)$. However, as the time-dependent probability distribution solution of the CME for a nonlinear network with a general time-dependent $\bar{\sigma}_b$ is most likely intractable, we transform $\bar{\sigma}_b(t)$ into a time-independent constant by taking its time-average $\bar{\sigma}_b^* = \int_0^t \bar{\sigma}_b(t')dt'/t$. The validity of this approach is justified in [182] by considering the Magnus expansion of the time-dependent solution of the master equation.
5. Obtain the time-dependent probability distribution solution of the CME of the linear network assuming that the rate parameters of the linearised reactions are time-independent constants. Note that this step is the major limitation of the LMA as closed-form solutions are available only for a handful of systems [182].
6. Finally, construct the approximate probability distribution of the nonlinear network at time t by replacing the respective rate parameters with their time-averaged equivalents obtained in step 4.

In MomentClosure.jl we provide automated generation of the closed moment equations using LMA given a nonlinear chemical reaction network and its linear equivalent. This encapsulates the *first three* steps of the LMA procedure outlined above which are general and can be seen as an original moment closure approximation. Nevertheless, in the online documentation we outline how the subsequent LMA steps can be manually performed in Julia to compute the approximate time-dependent probability distribution of a given nonlinear network on a case-by-case basis.

4.4 Implementation

In what follows, we discuss the main implementation details of MomentClosure.jl and its integration within the broader Julia package ecosystem, enabling a streamlined moment-based modelling workflow in which we can easily define a biochemical system, generate the corresponding moment equations using MAs and solve the resulting system of ODEs numerically. In Fig. 4.1, we summarise the workflow and compare the accuracy of different MAs applied to a simple stochastic model of an auto-regulatory genetic feedback loop.

4.4.1 Model definition

Modelling of chemical reaction networks in Julia is made easy by Catalyst.jl [270] that leverages a powerful symbolic-numeric modelling framework provided by ModelingToolkit.jl [271] and Symbolics.jl [272]. Namely, using Catalyst a chemical reaction network can be constructed by simply specifying the reaction stoichiometry and the corresponding reaction propensities—the model is stored in a symbolic intermediate representation as implemented by ModelingToolkit, which is based on the Symbolics computer algebraic system (more details can be found in the paper presenting the current iteration of Catalyst software [270]). MomentClosure is fully compatible with models defined through Catalyst and ModelingToolkit, allowing chemical reaction networks containing any number of molecular species and reactions with any type of smooth propensity functions. The only assumption made is that reactions occur in a single compartment of fixed volume. We note that systems involving multiple dynamically interacting compartments can be considered using Compartor [273], albeit the software is restricted to reactions with polynomial rate laws and supports only mean-field and gamma MAs.

4.4.2 Moment equations

Using MomentClosure, we can automatically obtain a system of ODEs describing the time evolution of moments up to any order. Internally, the *raw* moment equations are derived in a straightforward manner from the CME when the kinetics of a system are governed by the law of mass action. The derivation becomes more involved if the propensity functions take a non-polynomial form: here we adopt a general moment expansion framework based on Taylor-expanding the propensities around the mean, allowing us to obtain equations for the means and higher order *central* moments, as first formulated by [259] and [179]. The theory behind the method of moment expansion is described in Section 4.3.1.

The moment equations generated up to the specified order m can then be decoupled from all the higher order moments they depend on using one of the implemented MAs, i.e, expressing the moments above order m in terms of m^{th} and lower order moments using MA-specific assumptions. MomentClosure currently supports zero, normal, Poisson, log-normal, gamma, derivative matching, conditional Gaussian and conditional derivative matching closures [180, 181, 250]. We note that conditional MAs are available in only one other package (Cerena, see [255]) and have been found to be particularly effective in modelling gene networks with promoter switching dynamics [116, 250]. The details of all the MAs implemented in the package are discussed in Section 4.3.2. Furthermore, since the original publication of MomentClosure in Bioinformatics [2], we have implemented the Linear Mapping Approximation (LMA) [182] within the package. The LMA is a powerful technique developed in the context of gene regulatory networks, based on mapping a gene network with nonlinear protein-gene interactions onto an approximately equivalent linear model with no binding reactions, as described in more detail in Section 4.3.3. While the LMA provides closed moment equations for the given system and hence it can be considered as a novel moment closure method, we note that it is not strictly an MA in the usual sense as it additionally provides approximate time-dependent probability distributions for an array of different gene regulatory networks (given the corresponding linear model has a known analytical solution to the CME) [182].

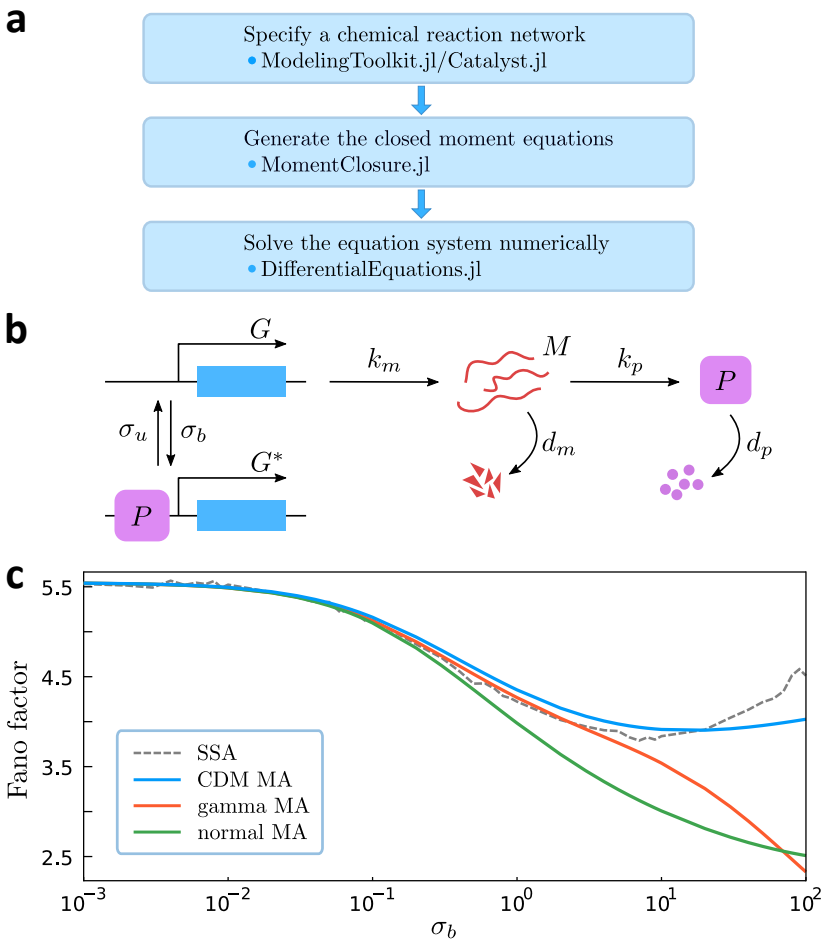

Figure 4.1: (**a**) The general workflow of moment-based modelling in Julia using MomentClosure and related packages. (**b**) Model of a negative auto-regulative genetic feedback loop. If a gene is active (G), an mRNA molecule (M) is produced with rate k_m which can be subsequently translated into proteins (P) with rate k_p (both degrade with rates d_m and d_p respectively). The negative feedback is introduced via protein binding to the gene with rate σ_b, switching the promoter OFF (G^*) and preventing the transcription (in contrast, switching ON occurs with rate σ_u). (**c**) Fano factor of the steady-state protein number as a function of σ_b where we have truncated the moment hierarchy at the second order using normal, gamma and conditional derivative matching (CDM) MAs, and compared the results to the true values predicted by the stochastic simulation algorithm [152] (averaged over 100k realisations). The initial condition is zero protein and mRNA molecules (set to 0.001 for MAs to ensure numerical stability) in state G, and the parameters are fixed as $\sigma_u = 10, k_m = 3, k_p = 50, d_m = 10$ and $d_p = 1$.

Finally, the closed moment equations can be solved numerically using any ODE solver implemented in DifferentialEquations.jl [203], which offers some of the most flexible and fastest implementations of a wide range of state-of-the-art numerical solvers. In addition, DifferentialEquations can be used in tandem with other Julia numerical analysis tools, enabling even further study of the resulting system of moment equations. For example, one can perform a bifurcation analysis using BifurcationKit.jl [274] or do model calibration/parameter estimation together with a sensitivity analysis using SciMLSensitivity.jl [275].

4.5 Example application

In this section, we demonstrate how to use MomentClosure together with other Julia packages to define a chemical reaction network, generate the corresponding moment equations, apply moment closure approximations and finally solve the resulting system of ODEs. This section is adapted from an online tutorial example available on GitHub and is included in this chapter for completeness.

We consider a special case of an oscillatory chemical system known as the Brusselator [276], characterised by the reactions

$$\begin{aligned} 2X + Y &\xrightarrow{c_1} 3X, \\ X &\xrightarrow{c_2} Y, \\ \emptyset &\underset{c_4}{\overset{c_3}{\rightleftharpoons}} X. \end{aligned} \tag{4.11}$$

We have chosen this particular model as it has been studied using different moment closure methods before in [180] and thus serves as a useful reference point. In this simple example, our goal is to obtain the closed second-order moment equations for the Brusselator using normal MA and solve them in order to visualise the system dynamics (trajectories of the means) in time.

Model Initialisation

Catalyst.jl [270] provides a comprehensive interface for modelling chemical reaction networks in Julia and can be used to construct models fully compatible with MomentClosure. More details on how to do so can be found in the official Catalyst's documentation. This way, the Brusselator can be defined as:

```
rn = @reaction_network begin
    # also include system size parameter
    @parameters c c c c
    (c/^2), 2X + Y  3X
    (c), X  Y
    (c*, c), 0  X
end
```

The returned `rn` is an instance of `Catalyst.ReactionSystem`. Note that we explicitly incorporate the system size Ω into the reaction rate parameters. The net stoichiometry matrix and an array of the corresponding propensities, if needed, can be extracted directly from the model using `Catalyst.netstoichmat` and `MomentClosure.propensities` functions respectively.

Moment Equations

The Brusselator model follows the law of mass action, i.e., all propensity functions are polynomials in molecule numbers $X(t)$ and $Y(t)$, and hence we can obtain either *raw* or *central* moment equations (as described in Section 4.3.1). We start by generating the raw moment equations up to the second order ($m = 2$):

```
using MomentClosure
raw_eqs = generate_raw_moment_eqs(rn, 2, combinatoric_ratelaws=false)
```

Note that we set `combinatoric_ratelaws=false` to ignore the combinatorial scaling factors in the mass-action reaction propensities which Catalyst otherwise includes (these are made implicit following the convention in [125]). The function `generate_raw_moment_eqs` returns an instance of `RawMomentEquations` that contains a `ModelingToolkit.ODESystem` object composed of all the moment equations (accessible as `raw_eqs.odes`).

We can use Latexify.jl to print out the generated moment equations in LaTeX format:

```
using Latexify
latexify(raw_eqs)
```

The code above results in the following system of ODEs:

$$
\begin{aligned}
\frac{d\mu_{10}}{dt} &= c_3\Omega + c_1\Omega^{-2}\mu_{21} - c_2\mu_{10} - c_4\mu_{10} - c_1\Omega^{-2}\mu_{11} \\
\frac{d\mu_{01}}{dt} &= c_2\mu_{10} + c_1\Omega^{-2}\mu_{11} - c_1\Omega^{-2}\mu_{21} \\
\frac{d\mu_{20}}{dt} &= c_2\mu_{10} + c_3\Omega + c_4\mu_{10} + 2c_3\Omega\mu_{10} + 2c_1\Omega^{-2}\mu_{31} - 2c_2\mu_{20} - 2c_4\mu_{20} - c_1\Omega^{-2}\mu_{11} - c_1\Omega^{-2}\mu_{21} \\
\frac{d\mu_{11}}{dt} &= c_2\mu_{20} + c_3\Omega\mu_{01} + c_1\Omega^{-2}\mu_{11} + c_1\Omega^{-2}\mu_{22} - c_2\mu_{10} - c_2\mu_{11} - c_4\mu_{11} - c_1\Omega^{-2}\mu_{12} - c_1\Omega^{-2}\mu_{31} \\
\frac{d\mu_{02}}{dt} &= c_2\mu_{10} + 2c_2\mu_{11} + c_1\Omega^{-2}\mu_{21} + 2c_1\Omega^{-2}\mu_{12} - c_1\Omega^{-2}\mu_{11} - 2c_1\Omega^{-2}\mu_{22}\,.
\end{aligned}
$$

The raw moments are defined as

$$\mu_{ij}(t) = \langle X(t)^i Y(t)^j \rangle$$

where the angle brackets $\langle\rangle$ denote the expectation value and we have explicitly included the time-dependence for completeness (made implicit in the formatted moment equations). Note that the ordering of species (X first and Y second) is consistent with the order these variables appear within the `Catalyst.@reaction_network` macro which we used to define the model initially.

The corresponding central moment equations can also be easily generated:

```
central_eqs = generate_central_moment_eqs(rn, 2,
                                          combinatoric_ratelaws=false)
```

Note that if any reactions in a chemical network are associated with non-polynomial propensity functions, the Taylor expansion order q (defined in Section 4.3.1) must be specified. However, the Brusselator contains only mass-action reactions and hence q is automatically determined by the highest order (polynomial) propensity. The function `generate_central_moment_eqs` returns an instance of `CentralMomentEquations`. We do not print out the resulting central moment ODEs for conciseness, noting that in this case they are functionally identical to the raw moment equations.

Coming back to the raw moment equations generated above, we observe that they depend on the higher-order moments. For example, the ODE for μ_{02} depends on third order moments $_{12}$ and $_{21}$ and the fourth order moment μ_{22}. Consider the general case of raw moment equations as described in Section 4.3.1: if a chemical reaction network involves reactions that are polynomials (in molecule numbers) of *at most* order k, then its m^{th} order moment equations will depend on moments up to order $m+k-1$. Hence the relationship seen above is expected as the Brusselator involves a tri-molecular reaction whose corresponding propensity is a third order polynomial in $X(t)$ and $Y(t)$. The number denoting the highest order of moments encountered in the generated `RawMomentEquations` can also be accessed as `raw_eqs.q_order` (returning a value of `4` in this case).

Applying Moment Closure

As observed above, the moment equations of the Brusselator are coupled and depend on the higher order moments—we have an infinite hierarchy of ODEs which cannot be solved directly and hence requires an approximate treatment. One way of approaching the problem is to apply moment closure approximations (MAs), in which higher order moments are expressed as functions of lower order moments, thus effectively truncating the hierarchy and enabling a numerical solution. A variety of MAs have been proposed in literature and are implemented in MomentClosure.jl, see Section 4.3.2 for more details.

We apply normal closure to the raw moment equations (generated above as `raw_eqs`) using `moment_closure` function:

```
closed_raw_eqs = moment_closure(raw_eqs, "normal")
```

The function returns `ClosedMomentEquations` that consists of both the `ODESystem` containing all closed moment equations as well as the specific closure functions for each higher order raw moments. Similarly, we close the central moment equations using

```
closed_central_eqs = moment_closure(central_eqs, "normal")
```

We can use Latexify again to print the closed ODEs, as well as the specific closure functions, in LaTeX format. For example, the moment functions used to close the central moment equations (expressions for the third and fourth order moments under the normal MA) can be displayed by calling `latexify` function with an additional `:closure` argument as follows:

```
latexify(closed_central_eqs, :closure)
```

Using the above we generate the following set of equations:

$$
\begin{aligned}
M_{30} &= 0 \\
M_{21} &= 0 \\
M_{12} &= 0 \\
M_{31} &= 3M_{11}M_{20} \\
M_{22} &= M_{02}M_{20} + 2{M_{11}}^2 .
\end{aligned}
$$

As discussed in Section 4.3.2, all the higher-order cumulants are set to zero under the normal MA, and hence the higher-order central moments expressed in terms of the corresponding cumulants and the lower-order central moments take a rather simple mathematical form.

Solving Moment Equations

The closed moment equations can be solved numerically using DifferentialEquations.jl, an extensive Julia library that provides a variety of highly-efficient numerical solvers and analysis tools. In order to do so, first we need to specify the values of all system parameters, the initial condition and the timespan to solve over. Then the `ModelingToolkit.ODESystem` corresponding to the moment equations can be directly converted into an `ModelingToolkit.ODEProblem` which can finally be solved. We go through the procedure step-by-step for the closed raw moment equations (`closed_raw_eqs`) obtained previously.

We start by creating a vector of reaction rate parameter values (note that the parameters are ordered as they appear in the `@reaction_network` macro):

```
# parameters [c, c, c, c, ]
p = [0.9, 2, 1, 1, 100]
```

Next, we specify the initial condition. Usually in the case of moment equations we consider *deterministic* initial conditions so that the molecule numbers at the initial time $t = 0$ take the specified integer values, $X(t = 0) = X_0$ and $Y(t = 0) = Y_0$, with probability one. This implies that initially the means will be equal to the molecule numbers, i.e., ${}_{10}(t = 0) = X_0$ and ${}_{01}(t = 0) = Y_0$, whereas all the higher order raw moments will be products of the corresponding powers of the means, e.g., ${}_{21} = X_0^2 Y_0$. Note that all central moments would be set to zero in this case. In MomentClosure, we can use the function `deterministic_IC`, which, given the initial molecule numbers, automatically constructs the variable mapping under deterministic initial conditions:

```
# initial molecule numbers [X, Y]
u = [1, 1]
umap = deterministic_IC(u, closed_raw_eqs)
```

Finally, the time interval to solve on can be specified simply as:

```
tspan = (0., 100.)
```

Now we are able to create the corresponding `ODEProblem` and solve the raw moment equations using any ODE solver implemented within DifferentialEquations.jl:

```
using DifferentialEquations
oprob = ODEProblem(closed_raw_eqs, umap, tspan, p)
# using the default Tsit5 ODE solver
sol = solve(oprob, Tsit5(), saveat=0.1)
```

In order to clearly establish how well the second order moment expansion using normal MA performs for the Brusselator with the given set of parameters, we can compare the resulting first-order moment trajectories to the true moment estimates obtained using the SSA. The Brusselator reaction network defined via Catalyst can be simulated using JumpProcesses.jl (a component package of DifferentialEquations.jl). In the following code snippet, we build a `JumpProblem` to simulate the system and compute the mean values of molecule numbers for both species X and Y averaged over 10^4 SSA realisations:

```
# convert ReactionSystem into JumpSystem
jsys = convert(JumpSystem, rn, combinatoric_ratelaws=false)
# encode that the molecule numbers are integer-valued
dprob = DiscreteProblem(jsys, u, tspan, p)
# create a JumpProblem; use Gillespie's Direct Method (SSA) as the solver
jprob = JumpProblem(jsys, dprob, Direct(), save_positions=(false, false))
# define an EnsembleProblem to simulate multiple trajectories
ensembleprob = EnsembleProblem(jprob)
# simulate 10k SSA trajectories
sol_SSA = solve(ensembleprob, SSAStepper(), saveat=0.1,
                trajectories=10000)
# compute the averages over all SSA trajectories
using DiffEqBase.EnsembleAnalysis
means_SSA = timeseries_steps_mean(sol_SSA)
```

The resulting moment trajectories in time are displayed in Fig. 4.2. The moment dynamics show damped oscillations which is the expected averaged behaviour of the Brusselator in a vast swathe of parameter space [166], as the individual oscillating trajectories become dephased over time due to random fluctuations. The comparison to the SSA reveals that the second-order moment expansion using normal closure captures the correct qualitative behaviour of the Brusselator and provides reasonably accurate moment estimates given this particular parameter set.

4.6 Discussion

MomentClosure.jl provides automated moment equation generation and closure approximations up to any desired expansion order. It is easily applicable to chemical reaction networks of any size containing reactions with smooth linear and nonlinear propensity functions. Moreover, utilising the popular Catalyst, ModelingToolkit and DifferentialEquations packages, MomentClosure makes the stochastic modelling of biochemical reaction kinetics using MAs efficient and accessible for Julia newcomers and experts alike.

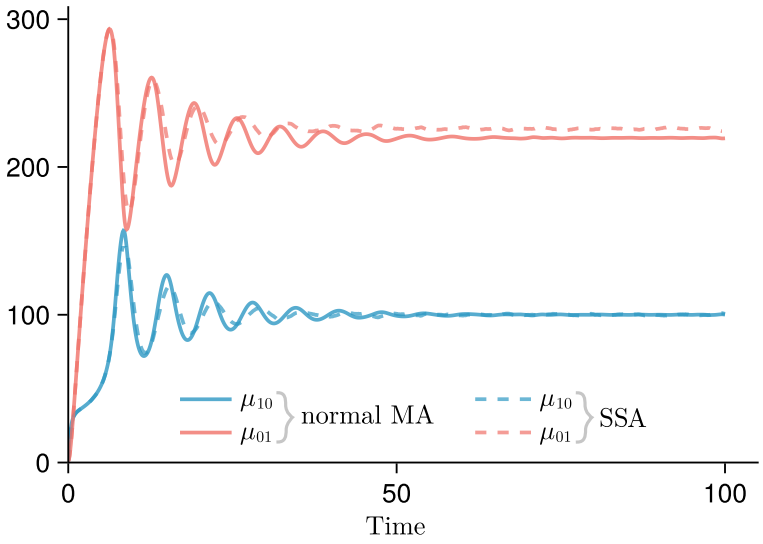

Figure 4.2: The temporal dynamics of the mean molecule numbers of X (μ_{10}, in blue) and Y (μ_{01}, in red) chemical species of the Brusselator model given Eq. (4.11), obtained by numerically solving the moment equations expanded to the second order using the normal MA. The reaction rate parameters are $c_1 = 0.9, c_2 = 2, c_3 = 1, c_4 = 1$, and the system size is set as $\Omega = 100$. The moment estimates are compared to the corresponding SSA values obtained by averaging over 10k simulation trajectories.

Modelling using moment equations is highly scalable and computationally efficient, making it commonly used in practice, especially for inference studies [116, 277–281]. However, the performance of moment-based inference can be inferior compared to more sophisticated approaches which consider full empirical distributions [116, 119, 120]. Moreover, MAs are based on ad hoc assumptions and no rigorous predictions can be made on whether the recovered moments will be accurate or even physically meaningful. In fact, moment equations truncated using MAs are prone to numerical instabilities and can lead to unphysical moment dynamics, such as divergent trajectories and non-physical oscillations or multi-stability [166, 180, 181]. Despite the potential drawbacks of moment-based approaches, this class of methods remains an important stochastic modelling tool. We hope that the unifying moment-based modelling framework provided by MomentClosure and other Julia packages will streamline the use of MAs in different scientific contexts and facilitate the development of new and performant moment closure methods.

MomentClosure is actively maintained and we plan to extend its functionality in the future. This includes the method of conditional moments (available in CERENA [255]), based on constructing a differential-algebraic system of equations where the low-copy number species are described in terms of their marginal probabilities, whereas the high abundance species are modelled in terms of their moments conditioned on the state of the low-copy number species [282, 283]. Similarly, the *selected-node stochastic simulation algorithm*—a hybrid approach that combines the conditional moment equations with the exact simulation of the selected chemical species—could also be implemented [284, 285]. In addition, we plan to add other automated moment expansion methodologies that are based, for example, on the binomial moments [286] or the so-called convergent moments [287], as well as the support for computing the stationary moment bounds [288].

A different type of feature we hope to integrate is the automated reconstruction of the probability distribution from the corresponding moments based on the *maximum entropy principle* [125, 198, 289]. While conceptually simple, the maximum entropy moment problem in multiple dimensions with high-order moment constraints requires careful numerical implementation [289, 290]. This functionality would also allow us to incorporate the *zero-information* closure [291, 292], based on reconstructing the joint distribution at each timestep, and its more efficient variants [293]. Novel closure approaches based on systematic polynomial approximation methods could also be considered [294, 295], as well as compartmentalised modelling [273].

Chapter 5

Approximating solutions of the Chemical Master Equation using neural networks

This chapter has been published as [3] entitled *Approximating solutions of the Chemical Master Equation using neural networks* in *iScience*. The text has been slightly edited and restructured for its inclusion in this thesis.

5.1 Abstract

The Chemical Master Equation (CME) provides an accurate description of stochastic biochemical reaction networks in well-mixed conditions, but it cannot be solved analytically for most systems of practical interest. While Monte Carlo methods provide a principled means to probe system dynamics, the large number of simulations typically required can render the estimation of molecule number distributions and other quantities infeasible. In this chapter, we aim to leverage the representational power of neural networks to approximate the solutions of the CME and propose a framework for Neural Estimation of Stochastic Simulations for Inference and Exploration (Nessie). Our approach is based on training neural networks to learn the distributions predicted by the CME from relatively few stochastic simulations. We show on biologically relevant examples that simple neural networks with one hidden layer can capture highly complex distributions across parameter space, thereby accelerating computationally intensive tasks such as parameter exploration and inference.

5.2 Introduction

The past decades have seen great progress in our understanding of the complex dynamics that underlie noisy cellular processes, both from an experimental and a theoretical perspective. Modern experimental techniques have shown that mRNA and protein levels can vary enormously at the single-cell level, but building detailed quantitative models that take into account the stochasticity of biochemical systems remains a daunting task. Due to the numerous challenges involved in describing the stochastic dynamics of these systems, modelling frequently relies on deterministic and small noise approximations which do not paint an accurate picture in many situations. Such simplified descriptions are often insufficient to describe how biochemical networks function in the presence of molecular noise [296, 297] and do not capture intricate noise-driven phenomena involved in cell fate decision [16, 78] and phenotypic regulation [298].

The most commonly used formalism for modelling biochemical reaction networks in a fully stochastic framework is the Chemical Master Equation (CME) [125], which describes how the probability distribution over states evolves with time. The CME cannot be solved analytically for most biologically relevant cases, and since the state space is typically infinite, numerical solutions of the CME often involve state space truncation methods such as the Finite State Projection (FSP) [137]. However, due to the combinatorial explosion of the state space in the number of species, using the FSP to solve the CME quickly becomes too computationally intensive for most non-trivial systems [144, 145, 149]. A wide variety of other approximation methods exist for the CME (see [125] for an overview), but these often trade computational efficiency for accuracy and are generally difficult to apply to complex systems involving many species and interactions.

While solving the CME remains challenging, *sampling* realisations of a system is possible thanks to the Stochastic Simulation Algorithm (SSA) [151]. Many physical quantities such as moments of molecule number distributions can be computed to arbitrary accuracy by repeatedly simulating samples from the system. Nevertheless, the SSA can be prohibitively computationally expensive when many repeated simulations are needed for accurate estimation of these quantities. Since simulations have to be performed anew for all parameters of interest, investigating system properties over time and parameter space with this approach can quickly become intractable. Furthermore, likelihoods are hard to estimate reliably using Monte Carlo methods, rendering likelihood-based inference particularly difficult [14].

Given the difficulties inherent in solving the CME exactly, it is natural to explore whether we could tackle this problem using neural networks, which in recent years have found diverse applications in the physical and biological sciences [209–212]. Their ability to detect patterns and learn complex representations given enough data is particularly useful when combined with simulator-based modelling, where such data can often be generated aplenty. In the context of systems biology, neural network-based approaches have been used to perform parameter inference on deterministic models, accelerate parameter exploration for models described by partial and stochastic differential equations [299], and even learn Markovian approximations to non-Markov models, translating them into the CME framework [300], amongst other applications.

Moreover, some recent studies have investigated the use of neural networks to learn various properties of stochastic biochemical reaction networks modelled using the CME [214–218]. In [214] the authors presented DeepCME, an approach that uses reinforcement learning to estimate summary statistics such as means and variances from stochastic simulations. The model abstraction procedure introduced in [215] employs Mixture Density Networks [301] to learn the transition kernel of the CME, and has been further extended into a framework providing automated neural network architecture search [216]. In the same vein, mixture density networks have been used to directly predict the probability distributions characterising the dynamics of an SIR-type model [217]. Finally, in [218] the authors demonstrate how Generative Adversarial Networks can be trained to generate full trajectories resembling the output of the SSA.

In this chapter we introduce Nessie, a framework for Neural Emulation of Stochastic Simulations for Inference and Exploration, based on using neural networks to learn solutions of the CME from stochastic simulations. Using only a moderate number of simulations of the specified reaction system at different parameter values, we train a neural network to learn the marginal probability distributions predicted by the CME over the whole parameter region of interest. We approximate the target distributions using mixtures of negative binomials, a flexible class of distributions particularly well-suited for this task [40, 302]. Once trained, Nessie becomes a surrogate for the CME that can efficiently and accurately predict the solution of the CME at different times for a wide range of parameters, bypassing the need to use further simulations or expensive approximation techniques to analyse the reaction network in question.

Our work differs from related approaches [214–218] in several regards. Unlike [215, 216] or [218], we do not aim to learn the transition kernels or the distributions of trajectories, i.e. the dynamics of the chemical system in question, but to capture the *marginal distributions* directly. In this sense Nessie is also different from DeepCME [214], which focuses on the task of learning summary statistics such as moments of molecule numbers. The relevant expectation values can be computed directly from the distributions predicted by Nessie, and we verify in our examples that Nessie can predict means and variances to a high degree of accuracy. Although our neural network is a variant of mixture density networks, in contrast to [215, 216] we do not use continuum approximations based on mixtures of Gaussians, thereby avoiding training and numerical stability issues that can arise in this context [207, 215, 216, 303]. As our approach directly targets experimentally observable distributions, it can also be used to perform parameter estimation based on comparing the neural network output with experimental data.

The chapter is organised as follows. In Section 5.3 we summarise the basics of artificial neural networks. In Section 5.4 we describe Nessie, providing an overview of the technical details of our neural network implementation and the workflow we use to predict the marginal probability distributions for a given system. In Section 5.5 we test our approach on several biologically relevant examples: an autoregulatory feedback loop, a genetic toggle switch involving mRNA and protein dynamics for two genes, a detailed model of mRNA turnover and a model of the mitogen-activated protein kinase (MAPK) pathway in *S. Cerevisiae*. The results indicate that Nessie can learn the dynamics of biochemical reaction networks for a wide range of parameters, allowing us to investigate physical properties of a given system such as multimodality and parameter identifiability. Furthermore, we demonstrate how Nessie enables us to build upon [302] and

perform efficient parameter inference from population snapshot data, a challenging problem for CME-based models. Further details on the neural network training and hyperparameter tuning specifics are given in Section 5.6. We conclude the chapter by discussing the results and the limitations of our study in Section 5.7.

5.3 Neural networks

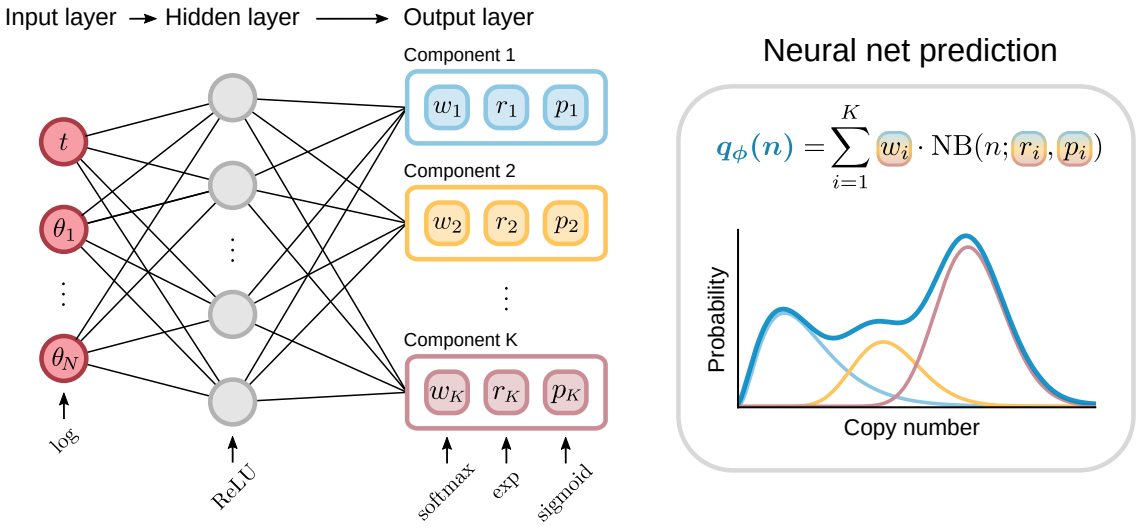

Figure 5.1: Architecture of Nessie, a simple feedforward neural network with one hidden layer. The input layer takes in the time t and the model parameters $\boldsymbol{\theta}$, and the output layer returns the corresponding negative binomial mixture components. Activation functions are indicated at the bottom.

A neural network learns a mapping f between inputs $\boldsymbol{x}$ and outputs $\boldsymbol{y} = f(\boldsymbol{x})$ via a parametric approximation f_ϕ such that $f_\phi(\boldsymbol{x}) \approx \boldsymbol{y}$. The basic building block of a neural network is a single neuron, which performs the mapping $\boldsymbol{x} \mapsto g(\boldsymbol{x} \cdot \boldsymbol{w} + b)$ for a weight vector $\boldsymbol{w}$, a bias b and a nonlinear activation function g. Several neurons arranged in parallel form a layer, and their outputs can be treated as inputs to another layer of neurons. By combining several layers in a row one gets a standard feedforward neural network, illustrated in Fig. 5.1. Here the first layer is called the input layer, the last layer is the output layer and the layers in between are hidden layers. For a comprehensive introduction to neural networks and deep learning we refer to [207].

Using activation functions with each neuron enables neural networks to learn complicated nonlinear mappings. Commonly used activation functions in the Machine Learning community are sigmoid functions, Rectified Linear Units (ReLUs) [304] and variants thereof [207]. For these activation functions one can show that a feedforward neural network with a single hidden layer and a sufficient number of neurons acts as a universal approximator, i.e. it is able to represent any sufficiently smooth function [305] to arbitrary accuracy. In theory, a better approximation could be achieved using a deep neural network with multiple hidden layers, which can compose simpler functions into increasingly more complex ones. Such “deep” neural networks, often combining a

variety of architectures more complex than a simple feedforward neural network, can outperform their shallow counterparts on difficult tasks such as Natural Language Processing, Computer Vision and others [207], which has led to a surge of interest in Deep Learning in recent years [208–212]. As we will see, however, a single hidden layer is enough for our purposes.

The network parameters ϕ (weights and biases for each neuron) that minimise the discrepancy between the mapping f_ϕ represented by the neural network and the true mapping f, are not known and have to be learned from data. This is commonly achieved by constructing a labelled set of training data $\mathcal{D}$ containing $N \gg 1$ different input-output pairs, and minimising a *loss function* $\mathcal{L}(\phi; \mathcal{D})$ that measures the deviation between the neural mapping f_ϕ and the target:

$$\mathcal{L}(\phi; \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\phi; \boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}), \tag{5.1}$$

where $(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})$ denotes the i^{th} example pair in the training dataset. The most appropriate loss function depends on the type of data and the task the neural network is trying to perform: common examples are L2 distances for regression, cross-entropy for classification and negative log-likelihoods for inference problems [207].

As the loss function is often highly nonlinear and nonconvex, minimising it with respect to the network parameters ϕ is a difficult task, most often performed using iterative gradient-based optimizers [306]. This requires the loss function to be differentiable with respect to the weights. Computing the gradients of the loss function is usually done using the backpropagation algorithm [207], which is implemented in most common deep learning frameworks such as Flux [307] or PyTorch [308]. Once the gradients have been computed one can use an optimization algorithm such as stochastic gradient descent or Adam [309] to minimise the loss. Note that the training set is generated once and reused for every gradient descent iteration.

In practice the behaviour of a neural network and its training procedure are determined by a number of hyperparameters, such as the number and size of hidden layers, the activation functions applied on each layer, the choice of the optimiser (as well as its associated parameters) and the convergence criterion. There is no universal formula for determining the best hyper-parameter choices for each task, and hence one has to resort to heuristics and hyperparameter tuning to find the best setup, which can be one of the most time-consuming aspects of training complicated neural networks. We discuss these practical considerations in connection to our approach below in Section 5.6.

Note that minimising the loss function over the training data does not guarantee that the neural network will be able to *generalise*, i.e. accurately learn the mapping for previously unobserved inputs. For this reason, the network's generalisation ability is usually evaluated on a separate validation dataset made up of examples that are not included in the training data [207]. Comparison of the loss on the training and validation datasets during the training procedure allows us to perform effective hyperparameter tuning. Finally the predictive performance of the trained network can be accurately measured on a separate test dataset consisting of yet another set of input examples.

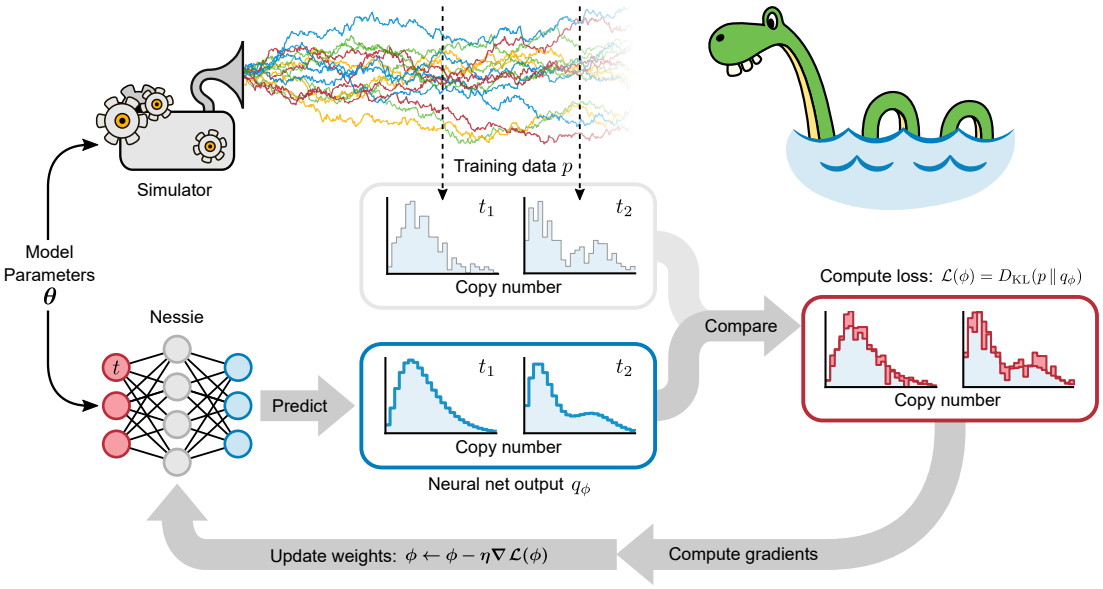

Figure 5.2: Workflow for training Nessie. Given model parameters $\boldsymbol{\theta}$, the reaction network is simulated repeatedly using the SSA to obtain empirical distributions at training time points $t_1, t_2, \ldots$. These are then compared with the output of the neural network and the total loss $\mathcal{L}$ is computed. In order to decrease the loss, the neural weights ϕ are updated iteratively via gradient descent until the loss has converged. In the figure η denotes the learning rate (see Section 5.6.1) and $D_{\mathrm{KL}}(p \,\|\, q_\phi)$ the KL divergence between the true distribution p and the neural network prediction q_ϕ.

5.4 Nessie

Our goal in this work is to learn marginal distributions predicted by the CME for different parameters and measurement times. As such the inputs to our neural network will consist of the chemical reaction network parameters $\boldsymbol{\theta}$ and the time t; as these can span several orders of magnitude we log-transform them first. Although we work with fixed initial conditions for each reaction system, this constraint could be relaxed by adding the molecule numbers at time $t = 0$ as extra inputs to the neural network.

We approximate the marginal distribution of interest by a mixture of negative binomials, a flexible parametric class of distributions that has been shown to be very accurate for a large class of reaction networks [40, 302]. Indeed, it is known that single-time marginal distributions predicted by the CME for many different reaction networks can be modelled as a mixture of negative binomials in the presence of timescale separation [34, 40, 41, 310, 311]. Experimental measurements of mRNA and protein distributions in bacterial, yeast and mammalian cells show that these are often fit well by such mixtures, even when timescale separation is not applicable [12, 312–314]. We remark that a mixture of negative binomials always has a Fano factor (variance over mean) greater than 1, and systems whose Fano factor is significantly less than 1 (see e.g. [51]) would benefit from a different parametric approximation which we shall not consider here.

A mixture of negative binomials can be parameterised as

$$q_\phi(n) = \sum_{i=1}^{K} w_i \cdot \mathrm{NB}(n; r_i, p_i), \tag{5.2}$$

where K is the number of mixture components, w_i is the weight of the i-th component and $\mathrm{NB}(r_i, p_i)$ is a negative binomial distribution with parameters (r_i, p_i). The number of components is fixed *a priori* and the weights are subject to the normalization constraint $w_1 + \ldots + w_K = 1$. Our task is therefore to learn a mapping $(t, \boldsymbol{\theta}) \mapsto (w_i, r_i, p_i)_{i=1}^K$ from the input parameters to those of the output distribution.

Each parameter characterising the output distribution is represented by a single neuron in the output layer. To respect the constraints on the weights w_i we apply the *softmax* activation function to the corresponding neurons: the outputs are exponentiated, then normalised to sum to 1. For the neurons corresponding to the count parameters r_i we choose exponential activation functions, and for the probabilities p_i we use sigmoid activation functions. The architecture of our neural network is shown in Fig. 5.1.

The number of hidden layers and the number of neurons per hidden layer can have a large impact on the representational power of a neural network (see Section 5.3). The networks we build throughout this chapter contain only a single hidden layer as we have found such architecture in our case to be easier to train and provide better predictive performance than "deeper" networks (see Section 5.6.2), an observation corroborated in [300]. We choose the number of neurons in the hidden layer depending on the complexity of the chemical reaction network at hand and use the ReLU activation function as it enables efficient training [304].

In our setup a single training point consists of an input point $\boldsymbol{x} = (t, \boldsymbol{\theta})$ and a reference distribution p of target molecules at time t for the specified reaction network with parameters $\boldsymbol{\theta}$, obtained by averaging over a number of SSA trajectories (or by using the FSP). We build the training set by sampling parameter sets $\boldsymbol{\theta}$ in the parameter region of interest and running simulations at each $\boldsymbol{\theta}$; this can be done in parallel for all parameters. We then use the simulation results at fixed times as training inputs to Nessie, using its ability to interpolate between these times to learn general time-dependent distributions. In order to ensure that the training data evenly cover the entire parameter region we use Sobol sequences [315], which generally provide more uniform coverage than random sampling.

A common method to match distributions in the statistics literature is to minimise the Kullback-Leibler (KL) divergence $D_{\mathrm{KL}}(p \,\|\, q_\phi)$, where p is the target distribution and q_ϕ the prediction; this procedure is mathematically equivalent to maximising the average log-likelihood under q_ϕ of a sample drawn from p [207]. Hence we use the KL divergence as our loss function for each point in the training set:

$$\mathcal{L}(\phi; \boldsymbol{x}^{(i)}, p^{(i)}) = D_{\mathrm{KL}}(p^{(i)} \,\|\, q_\phi(\boldsymbol{x}^{(i)}))\,. \tag{5.3}$$

This is equivalent to the cross-entropy, up to the addition of a constant that does not depend on the network weights ϕ. Computing the mixture of negative binomials q_ϕ for a given input point is straightforward using Eq. 5.2. The complete workflow for training Nessie is shown in Fig. 5.2.

We note that maximising the average log-likelihood for a mixture of Gaussians, as is commonly done with Mixture Density Networks (MDNs) [301], can lead to stability issues for more than one component. For example, the neural network can learn to place a Gaussian component at zero with arbitrarily low variance, which will give an arbitrarily high likelihood if 0 occurs anywhere in the training dataset, irrespectively of the overall quality of fit—an example of overfitting (see Section 5.6.1). This is because in the continuous case one deals with probability *densities*, which can become arbitrarily large in contrast to probabilities. We believe that this phenomenon is responsible for some common numerical issues observed e.g. in [207, 216, 303]. One can attempt to remedy this issue by integrating the density over a finite interval (say, $[-0.5, 0.5]$), or by regularising the precision of each component (thereby adding hyperparameters to the training procedure). In contrast, mixtures of negative binomials were not prone to overfitting in our experiments and did not require any form of regularisation.

In what follows, we quantify the relative accuracy of the trained neural networks by computing the Hellinger distance between the predicted and test distributions. Although the KL divergence is more suited as a loss function [207] to train the neural network for its computational efficiency, the Hellinger distance is a bounded metric and a more interpretable measure of the model's predictive performance.

We used Julia [204] with `Flux.jl` [307] to implement neural networks. Gradients of the loss function were computed directly by Flux using the built-in `Zygote.jl` automatic differentiation system [316]. The training datasets were constructed by defining chemical reaction networks via `Catalyst.jl` [270] and simulating them using `DifferentialEquations.jl` [203] (SSA) and `FiniteStateProjection.jl` (FSP). Minimisation of the Hellinger distance in the MAPK inference example was performed using `BlackBoxOptim.jl`. All numerical experiments were performed on a Intel Xeon Silver 4114 CPU (2.2 GHz) using 16 threads.

5.5 Results

5.5.1 Autoregulatory feedback loop

We first consider a simple autoregulatory feedback loop illustrated schematically in Fig. 5.3A, consisting of the following reactions:

$$G_u + P \underset{\sigma_u}{\overset{\sigma_b}{\rightleftharpoons}} G_b, \quad G_u \xrightarrow{\rho_u} P, \quad G_b \xrightarrow{\rho_b} P, \quad P \xrightarrow{1} \emptyset\,.$$

This system contains a single gene with two promoter states G_u and G_b, each associated with different protein P production rates ρ_u and ρ_b (mRNA dynamics are not modelled explicitly). The feedback is introduced via reversible binding of a protein molecule to the promoter region with binding rate σ_b and unbinding rate σ_u, which causes switching between the two promoter states. Finally, protein degradation is modelled by an effective first-order reaction. This system is a rudimentary example of stochastic self-regulation in a gene: the model functions as a positive feedback loop if $\rho_b > \rho_u$, and a negative feedback loop if $\rho_b < \rho_u$.

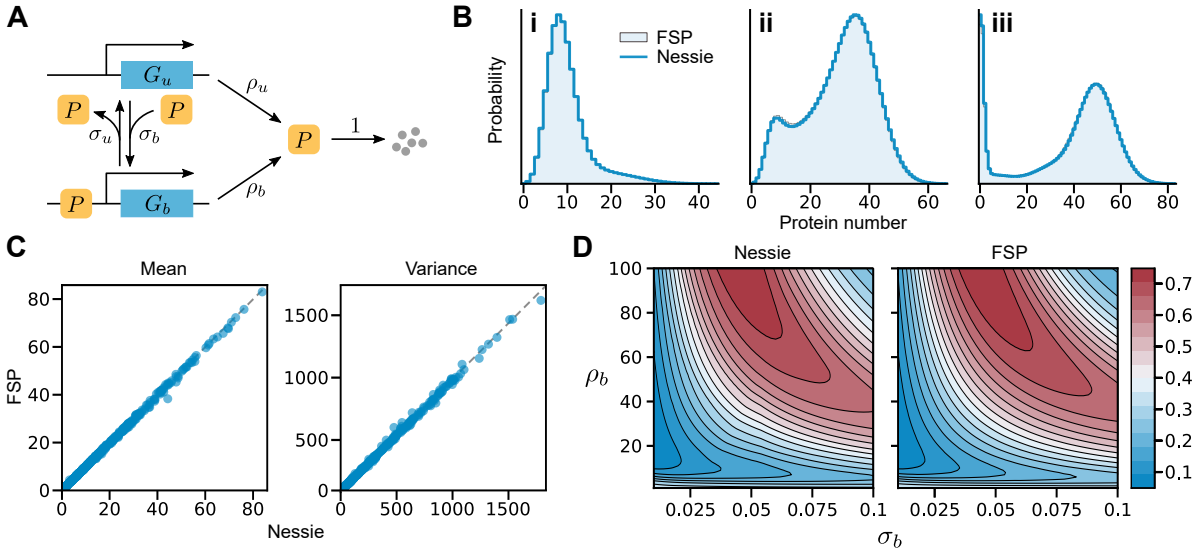

Figure 5.3: Nessie applied to an autoregulatory genetic feedback loop. (A) Schematic of the reaction network. We assume mass action kinetics for all reactions. (B) Protein distributions for three different test parameter values (indicated in Table B.1). The ground truth distributions were computed using the FSP. (C) Comparison of true and predicted means and variances of protein numbers at time $t = 100$ for the test set containing 500 parameter values. True means and variances were again computed using the FSP. (D) Exact and predicted bimodality coefficients as a function of ρ_b and σ_b, where we set $\sigma_u = \rho_u = 1$ and $t = 100$. Here the bimodality coefficients predicted by Nessie closely agree with their ground truth values.

Although the CME of the autoregulatory feedback loop has only been solved analytically in the steady-state [109] (although a complete semi-analytical solution has been recently proposed in [136]), in this specific case an efficient time-dependent numerical solution can be obtained with the FSP as the model contains few molecular species and chemical reactions. Estimating the probability distributions for the autoregulatory feedback loop via the FSP is much faster than using the SSA. This makes the autoregulatory feedback loop an ideal toy model for our initial experiments as we can relatively quickly build arbitrarily large training datasets with the FSP, calibrate the neural network, and probe the performance of Nessie in capturing the marginal distributions of protein numbers.

We use a training set of size 1k, a validation set of size 100 and a test set of size 500, sampled using a Sobol sequence in the parameter region indicated in Table B.1. For each datapoint we take four snapshots at times $t = \{5, 10, 25, 100\}$ and construct the corresponding histograms using the FSP. Our neural network consists of a single hidden layer with 128 neurons and outputs 4 negative binomial mixture components; we use a batch size of 64 for training. More details on the training procedure and the hyperparameter choices are given in Section 5.6.

In Fig. 5.3B we show the protein distributions for three test parameters sets, comparing the predicted distribution with the FSP results. Our approach provides highly accurate fits for the different distribution shapes obtained at these points in the parameter space, showcasing the flexibility of negative binomial mixtures in approximating the CME of the autoregulatory feedback loop.

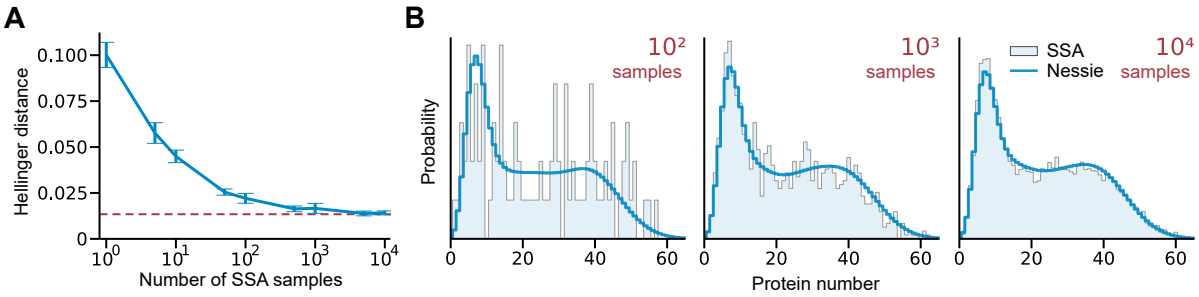

Figure 5.4: Training Nessie using the SSA. (A) Mean Hellinger distance computed over the validation dataset versus the number of SSA trajectories used to construct the histogram of each training datapoint. Here the validation dataset consists of 100 different parameter values at 4 time snapshots, constructed with the corresponding number of SSA trajectories. The error bars are obtained by averaging over 10 independent Nessie training runs, where for each run we resample the training and validation datasets and train a new model, as done for the manual tuning of other hyperparameters discussed in Section 5.6.2 and shown in Figure 5.9. The red dashed line indicates the accuracy of Nessie trained on the FSP data. (B) Example distributions constructed with 100, 1k and 10k SSA samples (indicated in red at the top) compared to the neural network predictions. We retrain Nessie for each plot using the respective number of trajectories per training point. Parameter values and ranges are given in Table B.1.

Having learned the output distributions for this chemical system we can compute various quantities of interest from these. In Fig. 5.3C, we compare the means and variances of the protein number predicted by Nessie to the true values computed using the FSP for all points in the test dataset. Furthermore, in Fig. 5.3D we analyse the bimodality coefficient as defined by [317]. The bimodality coefficient of a distribution is defined as $1/(\kappa - \gamma^2)$, where κ and γ are the skewness and kurtosis respectively, and is a measure of bimodality with higher values corresponding to strongly bimodal distributions. We see that Nessie provides a good approximation to this quantity and closely matches the FSP results.

Numerically estimating the bimodality coefficient for many different parameters is a computationally intensive task, whereas predicting it using the neural network, once trained on its 1k datapoints, is very quick: using Nessie we can produce the plotted heatmap in 0.03 s, in contrast to the FSP which takes 240s. This example illustrates how we can apply the neural network to rapidly and efficiently analyse large swathes of parameter space and, in turn, to determine the regions of bimodality in the system. Note that in evaluating the model performance we did not take into account the time required to generate the initial training data and to train the neural network, which are given in Table B.5. Although this does incur a notable overhead, it becomes largely insignificant in comparison to the computational gains provided by the trained neural network in large parameter exploration studies. Similarly, our approach scales to more complicated chemical reaction networks that are not amenable to study using the FSP, allowing us for example to perform global sensitivity analysis for the genetic toggle switch presented in the next section.

Finally, in Fig. 5.4 we consider using the SSA as an alternative to the FSP for constructing the training dataset. As each training histogram is built from a number of SSA samples, we investigate how many samples per training point are required to accurately train the network. Note that the numerical FSP solution is virtually indistinguishable from the exact CME solution as its approximation error can be systematically reduced by increasing the size of the truncated

state space [137], and hence it is effectively equivalent to an infinite number of SSA trajectories. We see in Fig. 5.4A that for the autoregulatory feedback loop using as few as 100 trajectories per training point enables Nessie to produce good approximations of the true distributions. In contrast, obtaining similar quality fits using the SSA alone would require two orders of magnitude more samples per parameter, as demonstrated in Fig. 5.4B where we compare the histograms obtained using 100, 1k and 10k SSA trajectories. One may expect this effect to be amplified for larger system sizes and more widely spread out distributions, where many more simulations are typically needed to obtain smooth histograms.

5.5.2 Genetic toggle switch

For our next experiment we consider a stochastic model of the genetic toggle switch, one of the first synthetic biological circuits [103]. The reaction network, introduced in [110], is composed of two mutually interacting genes (which we label A and B) and takes into account transcription, translation and the subsequent degradation of the produced mRNA and proteins (see Fig. 5.5A), consisting altogether of the following reactions:

$$
\begin{aligned}
&G_u^A + P^B \underset{\sigma_u^A}{\overset{\sigma_b^A}{\rightleftharpoons}} G_b^A, \quad G_u^B + P^A \underset{\sigma_u^B}{\overset{\sigma_b^B}{\rightleftharpoons}} G_b^B, \\
&G_u^A \xrightarrow{\rho_u^A} G_u^A + M^A, \quad G_b^A \xrightarrow{\rho_b^A} G_b^A + M^A, \quad M^A \xrightarrow{\gamma^A} M^A + P^A, \quad M^A \xrightarrow{\delta_m^A} \varnothing, \quad P^A \xrightarrow{\delta_p} \varnothing, \\
&G_u^B \xrightarrow{\rho_u^B} G_u^B + M^B, \quad G_b^B \xrightarrow{\rho_b^B} G_b^B + M^B, \quad M^B \xrightarrow{\gamma^B} M^B + P^B, \quad M^B \xrightarrow{\delta_m^B} \varnothing, \quad P^B \xrightarrow{1} \varnothing, \\
&M^B + P^A \underset{\sigma_u^M}{\overset{\sigma_b^M}{\rightleftharpoons}} M^B P^A \xrightarrow{\delta_{pm}} \varnothing \,.
\end{aligned}
$$

The translated protein A can bind to the promoter region of the gene that produces protein B, and vice versa with protein B binding to the gene promoter of protein A. This results in effective transcriptional regulation: depending on the mRNA production rate associated with each promoter state, the process can either lead to repression or activation of transcription for each species. In addition, the system contains post-transcriptional regulation mediated by protein A binding to the mRNA of species B and modulating its translation accordingly [318].

The toggle switch is noticeably more intricate than the autoregulatory feedback loop considered previously. It exhibits rich dynamics highlighted by diverse protein distributions that can be highly multimodal [110]. Due to the considerable number of reaction parameters, the frequent occurrence of high copy numbers (> 1000) and the complexity of the observed distributions, studying this system poses significant problems both analytically and computationally. This makes it a good challenge for Nessie.

Our aim is to predict the probability distributions of the target protein B in the genetic toggle switch. Note that we could similarly consider the distribution of protein A or, alternatively, the neural network architecture itself could be extended to predict both marginal distributions for the two proteins in parallel. If however joint distributions are required, the univariate mixture of negative binomials we use needs to be replaced by a multivariate equivalent; we briefly mention possible approaches in the Discussion.

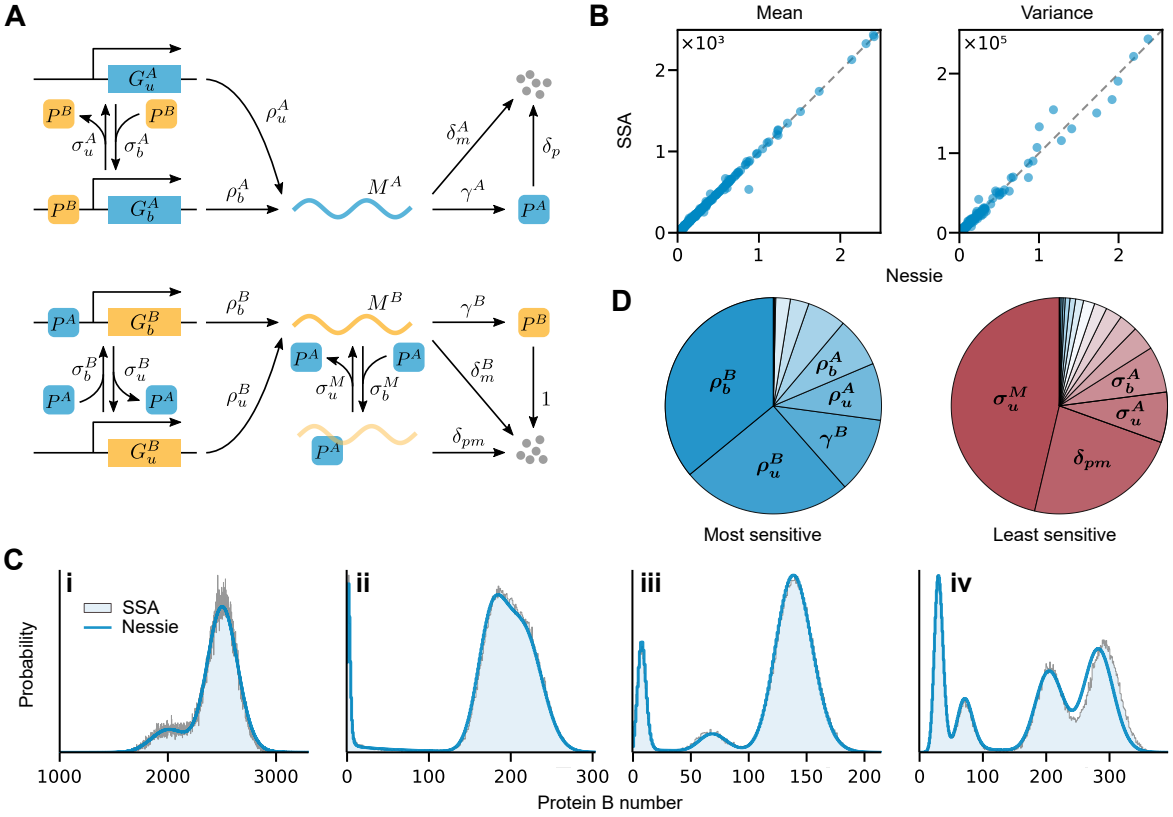

Figure 5.5: Nessie applied to a genetic toggle switch with post-transcriptional regulation. (A) Schematic of the reaction network (assuming mass action kinetics for all reactions). (B) Comparison of true and predicted means and variances of the protein B numbers for the test set consisting of 1k different parameter values at time $t = 100$ constructed using 100k SSA trajectories. (C) Protein B distributions for four different test parameter values (specified in Table B.2). The SSA distributions were computed by averaging over 100k trajectories. (D) Sensitivity of the Fano factor of protein B to parameter perturbation at time $t = 100$, where the pie charts show the most and least sensitive parameters. The results are obtained by using Nessie to compute the logarithmic sensitivity of the Fano factor to the 16 model parameters for 100k parameter values drawn from a Sobol sequence covering the training range given in Table B.2). We observe that only a few reaction parameters can be identified as typically the most/least sensitive (indicated in bold).

We draw 40k, 100 and 1k parameters for the training, validation and test datasets respectively using a Sobol sequence in the parameter region indicated in Table B.2). For each parameter set we take 8 snapshots at times $t = \{2, 4, 10, 16, 32, 50, 74, 100\}$. The complexity of the system prevents us from using the FSP to construct the reference histograms and hence we resort to the SSA. As discussed previously, a relatively small number of SSA samples can be used to successfully train the neural network. For this reason, we use 1k simulations for each training datapoint and 100k simulations for the validation and test data (in order to ensure a more accurate comparison to the true distributions). In this case, we use a neural network with a single hidden layer of 1024 neurons and 6 output mixture components, and fix the batch size to 1k for the training procedure (see Section 5.6.1 for more details). The remainder of our setup is the same as in the previous example. The data generation and neural network training times are given in Table B.5.

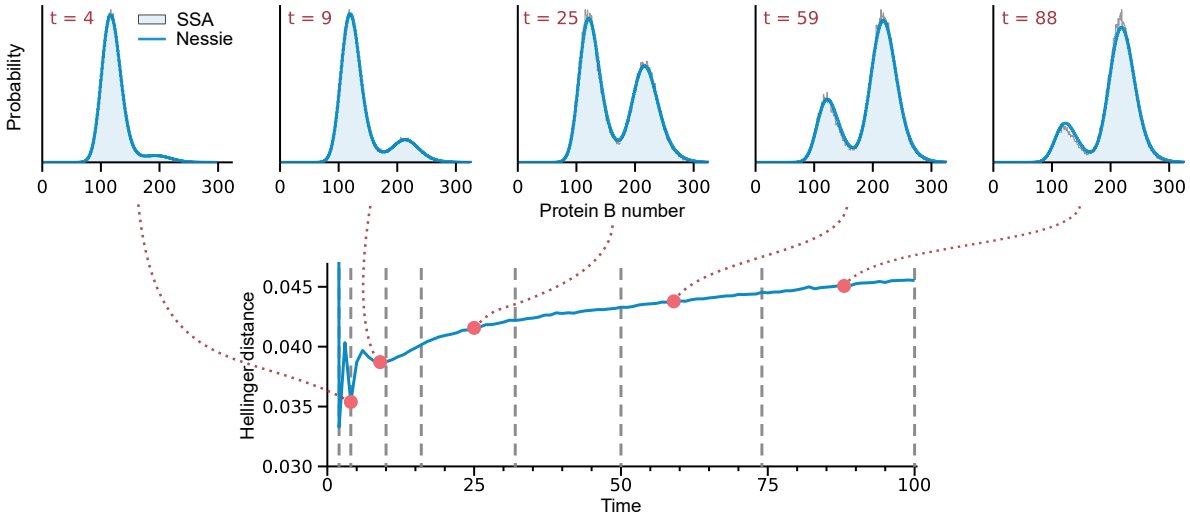

Figure 5.6: Using Nessie to interpolate and predict protein B dynamics in time. Bottom: Mean Hellinger distance computed over the test dataset made up of 1k different parameter values (constructed using 100k SSA trajectories) evaluated at times $t = \{1, 2, \ldots, 100\}$. The vertical gray dashed lines indicate the time snapshots used for training the neural network, showing that the predictive error does not notably increase in between the training points. Top: Time evolution of the protein distribution predicted by Nessie and the SSA (averaged over 100k trajectories) for an example parameter set, demonstrating Nessie's ability to accurately capture the time evolution of the protein B distribution.

In Fig. 5.5B we verify that the moments (means and variances) of the protein B numbers predicted by Nessie closely match those computed using the SSA for all test datapoints. Furthermore, in Fig. 5.5C, we compare the predicted protein distributions to the true distributions constructed by averaging over 100k SSA realisations. Notably, the trained neural network is able to reconstruct the complex distributions and provides a good approximation to the CME solution of the genetic toggle switch. The promising performance of Nessie highlights the usefulness of neural emulators for dealing with stochastic chemical reaction networks that go beyond the more tractable examples that are typically studied in the literature; we demonstrate this further by applying Nessie to a detailed model of mRNA turnover in the next section.

Next we explore the sensitivity of the genetic toggle switch to noise. The Fano factor, which is defined as the ratio of the variance of molecule numbers to the mean molecule number, is a commonly used measure of deviations from Poisson noise and the extent of transcriptional/translational bursting. We investigate how the Fano factor of the protein B number changes upon parameter perturbation. Using the trained neural network we have computed the logarithmic sensitivity [319] of the Fano factor of protein B to all reaction parameters over a wide parameter range and identified the most and least sensitive parameters on average, as shown in the pie charts in Fig. 5.5D. In particular, we can identify a few parameters that are the most or least sensitive to noise in the majority of cases. For example, for over 60% of the parameter space, the mRNA production rates ρ_u^B and ρ_b^B are the most sensitive, and hence tweaking these parameters is usually the optimal way to control the fluctuations in the protein B number. Note that performing such global sensitivity analysis is highly computationally expensive using the SSA, whereas with Nessie it can be approximated within minutes, making it possible to significantly accelerate further parameter exploration studies.

Although we trained Nessie with distribution snapshots only at a few fixed time points, an obvious question of interest is whether the neural network can capture the temporal dynamics of the chemical system over its whole trajectory. In Fig. 5.6 we plot the Hellinger distance between the predicted and true (SSA) distributions at times $t = \{1, 2, \ldots, 100\}$ averaged over the 1k test parameter sets. Remarkably, the predictive accuracy is largely similar throughout the whole time range, indicating that the neural network is able to effectively interpolate between the time points it has seen during training. Note that the worse performance at $t < 2$ is expected as we do not train on the initial transient during which the dynamics rapidly evolve from the initial condition.

5.5.3 Model of mRNA turnover

In this section we consider a detailed model of eukaryotic mRNA turnover, first proposed in [320] and consisting of the following reactions:

$$
\begin{aligned}
&G_{ON} \underset{\sigma_u}{\overset{\sigma_b}{\rightleftharpoons}} G_{OFF}, \quad G_{ON} \xrightarrow{1} A, \quad A \xrightarrow{k_2} B, \\
&B \xrightarrow{r_1} BC_1, \quad BC_1 \xrightarrow{r_2} BC_2, \quad BC_2 \xrightarrow{r_3} BC_3, \quad BC_3 \xrightarrow{r_4} BC_4, \quad BC_4 \xrightarrow{r_5} BC_5, \quad BC_5 \xrightarrow{r_6} C, \\
&C \xrightarrow{r_7} E, \quad C \xrightarrow{k_3} D, \quad E \xrightarrow{k_8} G, \quad E \xrightarrow{k_9} F, \quad D \xrightarrow{r_7} F, \quad D \xrightarrow{k_4} L, \\
&G \xrightarrow{k_{10}} M, \quad G \xrightarrow{k_{11}} \varnothing, \quad F \xrightarrow{k_8} M, \quad F \xrightarrow{k_4} I_1, \quad L \xrightarrow{r_8} I_2, \\
&M \xrightarrow{k_{11}} \varnothing, \quad M \xrightarrow{k_4} \varnothing, \quad I_1 \xrightarrow{k_5} \varnothing, \quad I_2 \xrightarrow{k_5} \varnothing.
\end{aligned}
$$

This reaction network contains a single gene with two states G_{ON} and G_{OFF}, which in the active state G_{ON} produces nuclear mRNA molecules A that are then degraded in a complex downstream pathway. The nuclear mRNA A is transported to the cytoplasm where it undergoes deadenylation followed by decapping and exonucleolytic degradation either in $3' \to 5'$ or $5' \to 3'$ direction [320], modelled as a complex sequence of first-order reactions (see Fig. 5.7A). Note that although in the original model mRNA production is described as occurring constitutively, we have extended it to include gene state switching in order to account for transcriptional bursting [321].

The mRNA degradation model contains even more species and reaction parameters than the genetic toggle switch presented in the Main Text, and approximating it using our framework is a further example of how Nessie can be applied to study large reaction networks. Our focus here is predicting the probability distributions of the total number of full-length mRNA segments (FL) that have not yet been exposed to exonucleolytic degradation, given by the sum $A + B + BC_1 + \ldots + BC_5 + C + D + E + F$.

We draw 100k, 100 and 1k parameters for the training, validation and test datasets respectively using a logarithmic Sobol sequence in the parameter region indicated in Table B.3. For each parameter set we take 8 snapshots at times $t = \{100, 230, 360, 500, 620, 750, 880, 1000\}$. We generate 1k SSA trajectories for each training datapoint and 100k simulations for the validation and test data. The number of hidden neurons is 1024 neurons, and we used 4 output mixture components, using a batch size of 1k for the training procedure. The times required to generate the data and train the neural network are given in Table B.5.

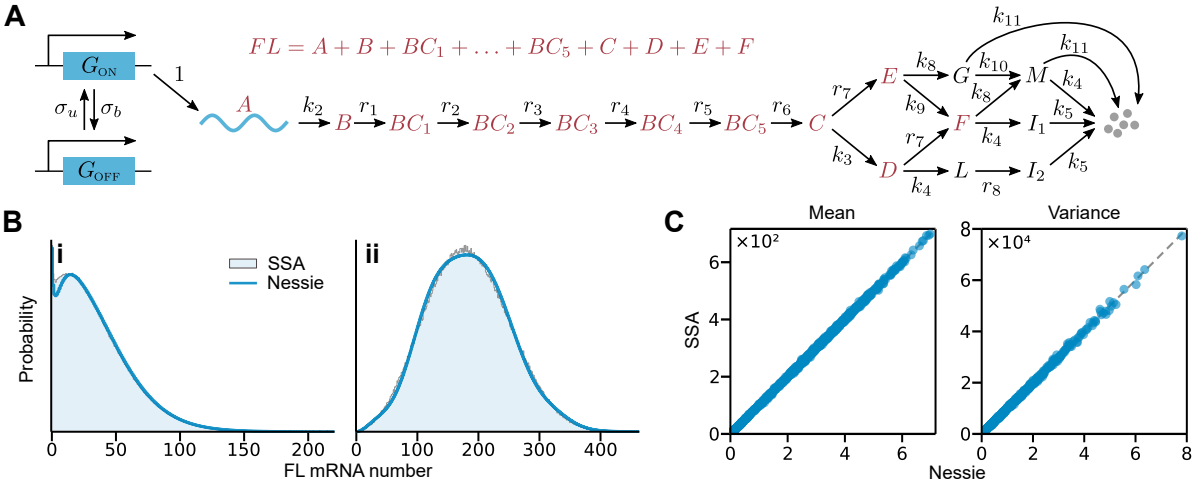

Figure 5.7: Nessie applied to a detailed model of mRNA turnover. (A) Schematic of the reaction network (assuming mass action kinetics for all reactions). We are interested in modelling the sum of full-length mRNA segments (the FL mRNA number), highlighted in red. Note that the mRNA fragment G is not related to the gene states G_{ON} and G_{OFF} (the notation is kept consistent with the original model [320]). (B) FL mRNA number distributions for two different test parameter values. The SSA distributions were computed by averaging over 1M trajectories. (C) Comparison of true and predicted means and variances of the FL mRNA numbers for the test set consisting of 1k different parameter values at time $t = 500$ constructed using 100k SSA trajectories. Parameter values and ranges are given in Table B.3.

In Fig. 5.7B we show that the predicted FL mRNA distributions closely match the true distributions constructed using 1M SSA realisations. Furthermore, in Fig. 5.7C, the means and variances of the FL mRNA numbers predicted by Nessie accurately compare with those computed using the SSA thus demonstrating that Nessie can perform well even in larger-scale applications.

5.5.4 MAPK pathway

We finally apply Nessie to a biological model of the MAPK pathway in *S. Cerevisiae* with the aim of inferring system parameters using experimental data from [277]. The reaction network can be seen in Fig. 5.8A and is modified from [277], removing extrinsic noise contributions. It consists of the following reactions:

$$G^{**} \underset{\eta_u}{\overset{\eta_b(t)}{\rightleftharpoons}} G^*, \quad G^* + CR \underset{\sigma_u}{\overset{\sigma_b}{\rightleftharpoons}} G, \quad G \xrightarrow{\rho_m} G + M, \quad M \xrightarrow{\rho_p} M + P, \quad M \xrightarrow{\delta_m} \varnothing, \quad P \xrightarrow{\delta_p} \varnothing.$$

where $\eta_b(t)$ depends on the current Hog1 concentration, which was measured experimentally, via the formula

$$\eta_b(t) = \frac{V_{\max}(\text{hog1}(t) + b)^h}{K_m^h + (\text{hog1}(t) + b)^h}$$

The number of ribosomes and chromatin remodellers in [277] were treated as constant and absorbed into the reaction rates ρ_p.

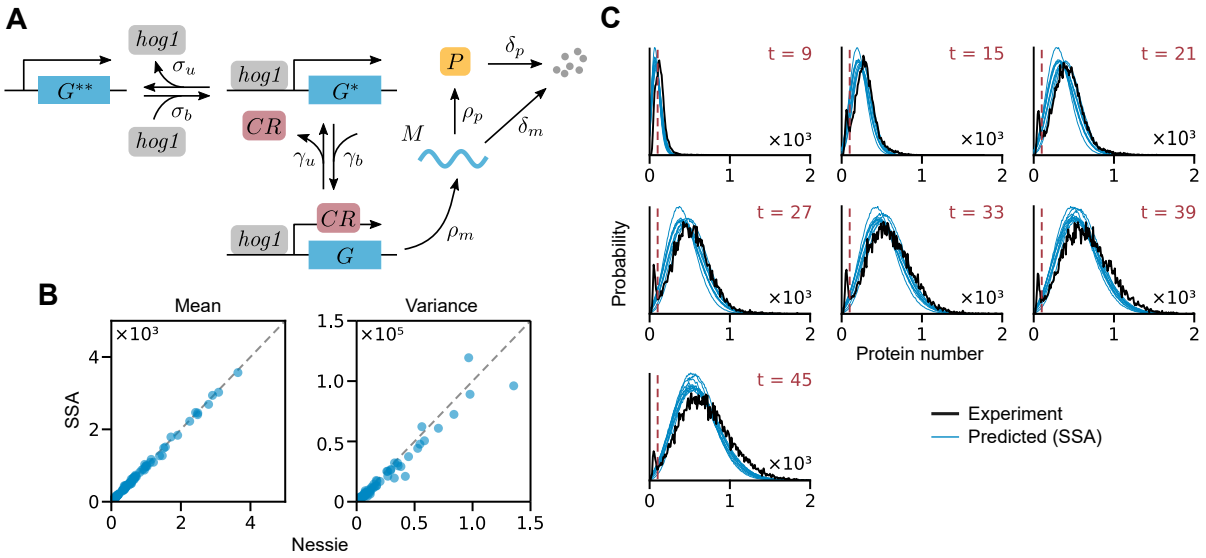

Figure 5.8: Nessie applied to a model of the MAPK pathway. (A) Schematic of the reaction network (see main text for details). The *hog1* concentration over time is taken from [277]. (B) True and predicted means and variances of the protein distribution for the test set consisting of 100 different parameter values at time $t = 27$ constructed using 100k SSA trajectories. (C) Comparison of the experimental distribution (black) with those predicted by the CME for the parameters inferred using Nessie (blue). To probe parameter uncertainty we performed 10 independent estimation rounds. SSA distributions were computed by averaging over 1M trajectories. Parameter values and ranges are given in Table B.4.

This reaction network describes the pSTL1 gene and includes activation due to a time-varying *hog1* signal, chromatin remodelling, transcription and translation. When yeast is subjected to external osmotic pressure, activation of the MAPK signalling cascade results in doubly phosphorylated *hog1* molecules entering the nucleus. These bind to the pSTL1 promoter, which is initially in an inactive state (G^{**}). Upon binding of *hog1* to the gene, subject to chromatin structure remodelling via the chromatin remodelling complex (CR), starts transcribing mRNA, which after translocation into the cytosol is translated into protein. In [277] the authors used flow cytometry to measure protein number distributions for the MAPK pathway, for which they proposed the above model. Protein distributions predicted by this model tend to be bimodal with a sharp peak at 0, as depending on the parameters a sizeable fraction of cells never starts transcribing mRNA before the *hog1* signal decays.

The data measured in [277] consists of intensity measurements (in arbitrary units, AU) at times $t = \{3, 9, 15, 21, 27, 33, 39, 45\}$ (in minutes) after salt was added to the solution to induce osmotic shock in the cells, which triggers the MAPK pathway. Since the fluorescence intensity per protein (I/P) was not measured in these experiments we assumed a value of 1AU per protein and rounded the estimated protein number to the nearest integer (in [277] it was noted that identifying all parameters is not possible from these experiments).

Observing that the experimental distributions at most times had a peak near 0 whose width was consistent across time points, we binned all observations less than 100AU, the approximate width of the peaks. The observed peaks are best explained by measurement noise that does not allow us to identify the exact protein numbers in the low copy number regime. Binning in this

case, while potentially losing some information, renders the procedure more reliable than the alternative of discarding observations below the threshold [121, 322]. Since measurements at time $t = 3$ were almost entirely below the threshold we discarded that time point for inference purposes.

Our goal is to find parameters consistent with the data by minimising the discrepancy between the experimentally observed distributions and the model output as predicted by Nessie. The training, validation and test sets consist of 15k, 250 and 100 points, respectively, which were generated from a logarithmic Sobol sequence in the parameter region indicated in Table B.4, chosen around the maximum a posteriori estimates reported in [277]. We use 1k simulations for each training datapoint and 100k simulations for the validation and test data. The size of the hidden layer was set to 2048 and the number of mixture components was 5. Since a significant fraction of the simulated trajectories had 0 proteins we added a sixth component that was set to be a Dirac delta at 0; this was performed by adding a single output neuron predicting the weight of this peak. To speed up training we split the procedure into two rounds, first training with 10% of the training data for 100 rounds and then with the entire training set for 400. We used a batch size of 1k throughout. Fig. 5.8B shows that the means and variances predicted by the resulting neural network are close to those obtained using the SSA.

Once Nessie is trained we estimate the model parameters by minimising the sum of Hellinger distances between the experimental distributions and Nessie's predictions, treating all observations below 100 AU as lying in one bin as discussed above. Since the outputs of the neural network are fully differentiable with respect to its inputs, including the model parameters, we can perform minimisation using any standard gradient-based algorithm. This is similar to the training procedure discussed in Section 2, except that we fix the neural weights and vary the model parameters instead. To evaluate the accuracy of the estimation scheme we ran the SSA at the predicted parameters, verifying that the results match the experimentally observed distribution (see Fig. 5.8C for the results using 10 estimated parameter sets). As can be seen in the figure our results do not reproduce the peak near 0 found in the experimental input, and an extensive parameter search did not lead to any parameters which exactly reproduce the experimental data. We therefore suspect that the model we used does not fully describe the dynamics of *hog1*-mediated gene expression and that obtaining better results will require a more detailed model than the one we are using.

While neural networks could be used to perform maximum likelihood estimation or Bayesian inference [323], we did not pursue likelihood-based approaches in this chapter. Due to the large number of datapoints (over 100k), any small approximation error in the distributions predicted by Nessie will get amplified by several orders of magnitude: as the likelihoods for each datapoint add up to form the total likelihood of the data, the errors in the likelihood will, too. This leads to highly fluctuating likelihood values for similar parameters that are an artefact of the neural approximation and not present in the true model. This results in the predicted posterior being concentrated tightly around one parameter set where these fluctuations result in a marginally higher likelihood for the experimental data than the others, and the resulting uncertainty estim-

ates reflect the approximation error incurred by Nessie instead of true parameter uncertainty. Such concentration of the estimated posterior due to randomness is a common problem in using MCMC with "tall data" [324], where estimating likelihoods for large datasets becomes very difficult.

Since fitting parameters using Hellinger distances does not directly provide uncertainty information, we estimated uncertainty by repeatedly fitting parameters to the experimental data; Table B.4 shows the results of 10 fits. As can be seen in Fig. 5.8B, these results produce similar distributions under the CME, yet some parameters such as σ_b and δ_p are spread over an order of magnitude. Such parameter unidentifiability is common with the type of experimental data measured in biological experiments and should be taken into account when interpreting results. In particular the Hill coefficient, which was allowed to range from 1 to 10, could not be narrowed down within this range.

Once the network is trained, globally optimising the Hellinger distance within the targeted parameter region takes a few minutes. Our approach should therefore be particularly suited for scenarios where distribution data is available for many copies of one network, e.g. when using a single gene expression model to analyse many different genes in an organism.

5.6 Methods

5.6.1 Training neural networks

We train our neural networks using the Adam optimiser [309], one of the most popular optimisation algorithms for this purpose. The gradients of the loss function with respect to the network parameters ϕ are calculated over minibatches of m training points, which are then used to update ϕ using the optimiser [207]. One training *epoch* is completed by iterating over all minibatches in the training dataset and hence performing many gradient steps (which can lead to faster convergence). Before training we initialise the network weights using the Glorot Uniform method [325].

The batch size and the learning rate are two optimiser hyperparameters that may significantly affect the results of the training procedure. We adjust these hyperparameters and define our stopping criterion using heuristic arguments outlined below.

Batch Size: It has been noted that large batch size m may reduce the model's ability to generalise, whereas small m can lead to more reliable results [326]. In our experiments we observed that very small batch sizes did little to improve results while significantly increasing training time. To balance these observations, we usually choose m to be 2-10% the size of the training dataset, which consistently gave good performance.

Learning Rate: The learning rate η of the optimiser controls the step size of each gradient update and should be chosen appropriately: too low a choice can lead to slow convergence, whereas a large learning rate can overshoot the target minimum. In our experiments we usually initialise $\eta = 0.01$ and decrease it as training progresses. Namely, we periodically monitor the loss function over the validation dataset and halve η if the loss has improved by less than 0.5% over the last 25 epochs on average.

Stopping Criterion: The training procedure is terminated after η has been decreased 5 times, which usually indicates that optimization has stalled. We found this stopping criterion to work well for our examples as training beyond this point did not often lead to significant improvements in accuracy.

The learning rate decay described above is similar to early stopping [327], a regularisation technique that helps to prevent overfitting of the training dataset. Overfitting occurs when the neural network learns, or "memorises", particular features of the training dataset that are not representative of the model as a whole, and loses its ability to generalise to unseen data. This can often be detected by an increase in the validation loss together with a monotonically decreasing training loss. While a number of popular regularisation strategies can be used to prevent this, such as L2 regularisation or dropout [207], we did not find overfitting to be an issue in our experiments. We conjecture that this is due to the rigid nature of the negative binomial distribution which, unlike a Gaussian, cannot overfit single datapoints away from 0.

5.6.2 Hyperparameter tuning

As noted above, training a neural network effectively requires finding a good architecture and hyperparameters. Beyond manual tuning this is classically done using black-box optimization methods such as grid search, random search or Bayesian Optimisation [328]; the related work in [216] uses a more recent differentiable architecture search. For deep neural networks such approaches can be very computationally expensive depending on the number of hyperparameters. In contrast, our approach based on a single hidden layer can feasibly be tuned manually, and using the autoregulatory feedback loop as a testbed we can obtain intuition about the effect of each hyperparameter for our problem.

Hidden Layers: The number and structure of the hidden layers in a neural network greatly affect its capacity, i.e. its ability to represent sufficiently complex functions. In Fig. 5.9A we plot the Hellinger distance between the true and predicted distributions for a single hidden layer with different numbers of neurons. We observe that the network's capacity quickly grows with the number of neurons, reaching peak accuracy at about 128 neurons. Increasing the number further does not have a measurable effect on our model's performance beyond increasing training time.

Network Depth: In Fig. 5.9B we compare the predictive performance for neural networks with multiple hidden layers of different sizes. The results suggest that shallow neural networks consisting of a single hidden layer are as effective as deeper ones for our purposes, while being easier to set up and train. Our experiments with the other systems (not shown) corroborated this observation.

Number of Components: Another important hyperparameter is the number of negative binomial components in the output mixture. In Fig. 5.9C we see that at least 4 components are needed to obtain good approximations for the autoregulatory feedback loop. This is not unexpected as the protein distributions of the autoregulatory feedback loop can be distinctly bimodal in certain parameter regimes, as shown in Fig. 5.3. The maximum number of modes in models of gene networks (that we focus in this study) is usually given by the number of gene states times

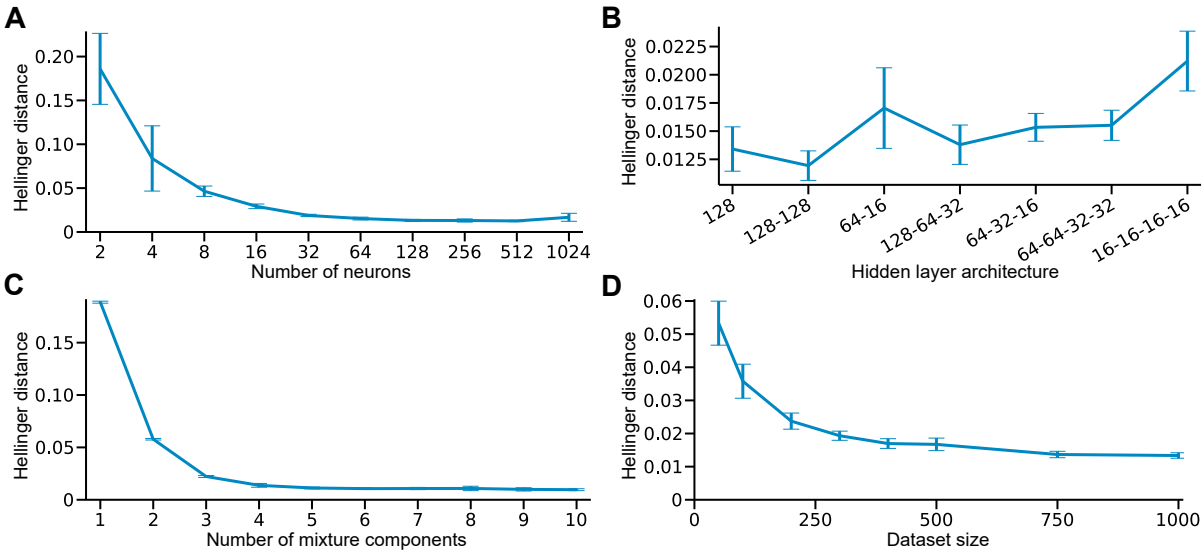

Figure 5.9: Results of manual hyperparameter tuning for the autoregulatory feedback loop. Each point in the figure is the mean Hellinger distance for the validation dataset (consisting of 100 different parameter values at 4 time snapshots constructed with the FSP), computed over 10 independent training runs with a differently initialised neural network. For each run, we reinitialise the network, construct new training and validation datasets by drawing new parameters from a fixed Sobol sequence and train the network again. Note that the validation dataset used to produce the figure is distinct from the validation data used during training.

the total number of gene copies [110], and this acts as an effective lower bound for choosing the number of mixture components. However, increasing the number of mixture components beyond this lower bound may be required to improve the accuracy of the neural network further, as is the case for the autoregulatory feedback loop.

Although adding extra components does not lead to overfitting, as observed by [302] it increases the training time and can also make the network more prone to mistakenly predicting too many modes in the solution for certain parameter regimes, which can be regarded as an unphysical artefact. A viable option to find the optimal number of mixture components would be to initialise the neural network with a relatively high number of components and implement L1 regularisation [207] in order to sparsify the output mixture during training.

Dataset Size: Having enough training samples is essential to train a neural network in a way that allows it to learn to generalise. In Fig. 5.9D we show how increasing the size of the training dataset improves the performance of our network. Although relatively small datasets are sufficient for achieving good accuracy, in this case we have only five neural network inputs. Therefore, due to the curse of dimensionality, significantly larger datasets may be needed for effective training on chemical systems involving more reaction parameters. Adding more training points when the validation loss is significantly higher than the training loss is an effective way to determine the appropriate size.

Number of Simulations: The number of simulations at each training point also affects the accuracy of the fit. In general we suggest considering more SSA samples if the neural network does not provide a good fit on the training data. As fitting negative binomial mixtures to samples has a strong regularising effect, the number of simulations per training point is generally much less than what is required to get an accurate histogram from samples (see Fig. 5.4).

5.7 Discussion

In this chapter we presented Nessie, a framework that allows us to train neural networks on simulation data to accurately estimate the solution of the CME for various biological systems. Our approach is scalable to complex nonlinear reaction networks with over a dozen parameters that exhibit diverse, multimodal dynamics across parameter space. We illustrated the performance of Nessie on four examples: a well-studied autoregulatory feedback loop, a complex genetic toggle switch, a detailed model of mRNA turnover and the MAPK pathway in *S. Cerevisiae*. The latter models pose significant challenges both analytically and computationally due to the number of species and reactions involved, yet Nessie allows us to efficiently emulate them and analyse their properties, with applications for parameter exploration and estimation. While the models we have tested in this work were all Markovian, the simulator-based nature of our approach makes it suitable for non-Markovian models including delays, see e.g. [329, 330].

Nessie can be particularly useful in rapidly exploring large swathes of parameter space, for example to perform a local or global sensitivity analysis. This has many uses, e.g. to guide the tuning of parameters to find a desired phenotype [331], for the design of optimal experiments [332], to provide insights into the robustness and fragility tradeoff in genetic regulatory mechanisms [333] and to find those parameters which most influence the size of transcriptional noise [334]. We note that performing any such analysis using the standard stochastic simulation methods like the SSA can be prohibitively computationally expensive [335].

Following methodology similar to that proposed by [323], Nessie can be used to fit models to data by matching experimentally observed distributions to those predicted by the neural network, as demonstrated in the case of the MAPK pathway model where we recovered model parameters that are mostly consistent with experimental observations. We remark, however, that this approach has to be used with care in the context of likelihood-based inference due to small approximation errors in the likelihood being amplified in the presence of many datapoints. In order to be reliable any such approach, including Bayesian inference, must take into account the bias introduced by the choice of approximation. This could be done e.g. by placing a prior over network weights and treating them as unobserved variables, sampling from the resulting Bayesian neural network using Hamiltonian Monte Carlo methods [336].

As discussed in Section 5.2, our approach differs from other studies that use neural networks to predict the dynamics of stochastic biochemical systems [214–218]. As all of these approaches try to learn different things, comparing them directly is not straightforward and is further complicated by the sensitivity of neural networks to architecture and hyperparameter choices [207]. An advantage of Nessie is that it requires relatively little setup in terms of hyperparameter optimisation. Due to its architectural simplicity a very limited amount of tuning is needed,

without requiring automated neural architecture search techniques [216], and we provide a detailed discussion of the relevant hyperparameter and training considerations in Section 5.6. We hope that this will enable the interested reader to quickly deploy and apply Nessie to their favourite reaction network.

5.7.1 Limitations of the study

Although Nessie can relatively accurately interpolate in time between the training snapshots for such models like the genetic toggle switch, its performance may be inadequate when applied on systems with complex oscillatory behaviour. This is a general limitation of our approach, which uses a simple feedforward network and therefore may not be able to efficiently represent oscillating functions [337]. To remedy this, besides the recently proposed generative adversarial network-based approach in [218] one could consider more sophisticated neural network architectures such as recurrent neural networks [207] and universal differential equations [203]. This would allow us to extract temporal features such as power spectra and first passage times, which, while difficult to measure experimentally, have been shown to provide a wealth of information about the system and significantly aid in model discrimination [338–340].

While in this chapter we have focused on the task of learning one-dimensional marginal distributions predicted by the CME, Nessie can also be extended to capture joint distributions. One way to implement this could be replacing the mixture of univariate negative binomials by a mixture of independent negative multinomials or alternative multivariate distributions. Such generalisation only requires updating the output layer to correctly represent the parameters of the new mixture. However, we expect the computational cost of training such a network to greatly increase as the state space (and hence the number of required gradient computations) grows exponentially. The construction of training datasets may also become significantly more expensive, as generating sufficiently smooth multi-dimensional histograms with the SSA may require many more trajectories than in the one-dimensional case. Another way to learn multivariate distributions was recently proposed by [219] and consists of learning the distribution of one species conditioned on the number of another species, whose marginal distribution is known beforehand (or can be learned). Exploring this and other possible ways to efficiently approximate joint distributions using Nessie remains an interesting avenue for future research.

5.8 Data availability statement

All original code has been deposited at `https://github.com/augustinas1/Nessie` and is publicly available since the original publication [3]. The neural network training data can be generated using the original code, or it can be shared upon request. The MAPK Pathway inference example uses data from [277] which can be obtained from the authors.

Chapter 6

Conclusions and Outlook

In this thesis, we have derived an approximate closed-form time-dependent solution for the Michaelis-Menten enzyme kinetic mechanism, built a software tool for automated moment closure approximations and proposed a neural network-based approach for surrogate modelling of chemical reaction networks. With the present research, we aimed to contribute to the study and development of analytical and computational approximations for the Chemical Master Equation, which can help to better understand stochastic reaction kinetics inside cells. In what follows, we reiterate the main results, highlight some of the key differences between the approximation methods considered in this thesis, and point out several future research directions.

We have demonstrated in Chapter 3 how the CME for the MM reaction system can be systematically reduced upon the rapid equilibrium approximation using averaging [189], and how the resulting single-variable one-step master equation can then be solved exactly using a linear algebraic method [129], allowing us to obtain the time-dependent marginal distributions of substrate and enzyme numbers. We further established the timescale separation conditions under which the solution is valid: the rate of complex dissociation (complex decay into substrate and enzyme) must be much greater than the rate of product formation (complex decay into product and enzyme). Similarly, we have shown that our approximation is accurate over a wider region of parameter space than the commonly used discrete stochastic MM approximation (based on the deterministic quasi-equilibrium assumption). Furthermore, we note that the exact solution for the MM reaction mechanism is only known in the case of a single enzyme [132, 225]. Although an approximate closed-form solution for a system with multiple enzymes has been reported previously in [65], the derivation is based on seemingly heuristic arguments and the solution's validity conditions do not appear to have been rigorously explored. Finally, analysis of the model's transient behaviour revealed the presence of transient bimodality—a phenomenon that has also been reported in other systems, albeit rarely [239, 244, 341–343], and a quantitative explanation for its appearance in terms of system dynamics remains to be found.

In Chapter 4, we have presented MomentClosure.jl, a software tool for automatic generation of moment equations and the application of MA methods. Although a number of conceptually similar tools exist [180, 251–255], neither of them are actively maintained and provide only limited functionality or require proprietary software. MomentClosure can be applied to any chemical reaction network with arbitrary propensity functions to derive the time-evolution equations for the moments up to any desired order, and provides the most comprehensive selection of MA methods to date. Although MomentClosure does not currently support the

method of conditional moments [282] which is available in CERENA [255], we plan to add this functionality in the future. Nevertheless, the conditional Gaussian and derivative matching methods can be seen as special cases of the method of conditional moments when the low copy number species are binary variables, e.g. in two-state models of expression. We also note that the linear mapping approximation is not available in any other tools and can accurately capture the dynamics of gene networks [182]. In addition, the generation of moment equations could be extended to biochemical systems with multiple interacting compartments [344], otherwise available in Compartor [273]. Finally, MomentClosure is the first such package to be implemented in Julia, a highly-performant and versatile language that offers a growing number of state-of-the-art modelling tools [205], which together with MomentClosure provide a unifying moment-based modelling framework that is readily accessible to other researchers in systems biology.

Compared to the timescale separation techniques considered for enzyme kinetics, which require a thorough analysis and identification of the fast and slow timescales in the system, moment closure-based methods are much easier to implement and can be applied in any context. However, this comes at a cost, as most MAs are based on either distributional assumptions [180, 181], matching of the time derivatives of the approximate and exact moments at some initial point [267, 268], or conditioning on the state of the low copy number species [250]—these are ad hoc approximations and their physical validity or even numerical convergence cannot be generally guaranteed [166, 180, 181]. Nevertheless, MomentClosure could be used to build on the previous work [166, 180, 181] and streamline the numerical studies to extensively establish the accuracy of different MAs and their dependence on the moment truncation order for more complex reaction networks.

Moment closure methodology is commonly compared to the system size expansion (SSE) [5], as both types of approximations are computationally efficient and can be used to obtain a system of closed moment equations [125]. The advantage of the SSE is that it is not based on heuristic assumptions and can be derived systematically by expanding the CME around its deterministic rate equations in powers of the compartment volume [5, 125]. Therefore, the SSE can be expected to be accurate for large enough systems (implying large enough molecule numbers) and also provide good approximations for smaller systems if the higher-order expansion terms are included. Moreover, the lowest-order truncation of the SSE, known as the linear noise approximation, yields an expression for the probability distribution, which can also be improved by including the higher-order corrections using an expansion developed in [169] (only applicable for systems involving a single species).

However, because the SSE is an expansion around the deterministic mean, its applicability is limited to deterministically monostable systems, although the conditional system size expansion proposed in [110] could be used to capture the multimodal dynamics under timescale separation. Moment closures, on the other hand, do not have such conditions imposed and could in principle accurately predict the dynamics even for deterministically multistable systems with low abundance species. Moreover, it is possible to reconstruct the probability distributions from the obtained moments using the principle of maximum entropy [198, 289, 290]. While the SSA and MA methods have been compared in the literature [125, 178, 198, 279, 345], previous studies have considered only a few simple systems at a low truncation order with a limited set of MAs.

Therefore, it would be interesting to rigorously extend the numerical comparisons between the two approaches. This could be done by developing a Julia package to automate the SSE up to an arbitrary order, in the same spirit as MomentClosure, making such numerical analysis easy to perform for any chemical reaction network. We note that a computational implementation of the SSE is currently provided (to a limited order) only by the Intrinsic Noise Analyzer [346] and CERENA [255].

In Chapter 5, we have introduced Nessie, a novel approach to approximate the solutions to the CME using neural networks. We have trained Nessie on several biochemical systems and demonstrated that a relatively simple neural network can accurately learn one-dimensional marginal probability distributions predicted by the CME and hence capture complex, multimodal system dynamics over the parameter space. As discussed in Section 5.2, the use of neural networks for the CME-based modelling has been explored previously, but mostly focusing on different learning tasks [214–219].

In contrast to the time-scale separation techniques and moment closure schemes, Nessie does not make any assumptions about the system, except for what is built directly into the neural network architecture. Namely, we expect the specified number of negative binomial mixture components to adequately represent the system dynamics over the parameter space (a different parametric approximation may be required for systems with sub-Poissonian distributions [51, 347]). Although Nessie does not have an accuracy guarantee, it can be iteratively improved: if the prediction error is high in a specific subdomain of the parameter space (including time), it can be reduced by sampling more training points in the problematic region and training the model further. Depending on the model's capacity, its predictive power can also be increased by adding more neurons or mixture components—our proposed neural network architecture is simple and requires relatively little hyperparameter tuning. Although the data generation and the neural network training incurs an overhead, once trained Nessie becomes an effective surrogate model of the CME that can greatly outperform exact simulations [220]. This way, Nessie can be utilised to significantly speed up computationally demanding tasks such as parameter exploration and inference.

One potential application of Nessie could be to emulate the approximate closed-form solution of the CME for the MM reaction mechanism obtained in Section 3. Although we have successfully used the derived solution to explore the system dynamics and detect the parameter regimes where transient bimodality is present, its complicated mathematical form can become difficult to evaluate numerically for large initial substrate and enzyme numbers, whereas its approximation using Nessie could prove much less computationally expensive. We note that this particular issue has also been observed by Lente [348] while studying their own approximate solution of the MM reaction system [65], who suggested a heuristic approximation to the full solution based on constructing a mixture of binomial distributions.

Moreover, the numerical evaluation of the analytical closed-form solutions can be particularly problematic in the development and inference of mechanistic models of gene expression. For example, the exact transient mRNA distribution can be derived for an extended telegraph model that explicitly incorporates the cell cycle dynamics [43]. However, its generating function solution takes a rather complicated form that also needs to be converted back into a probability distribution—its evaluation can be resource intensive and potentially numerically

unstable, depending on the molecule numbers and the specific parameter values. This can make parameter inference for the model intractable as the procedure typically requires many function evaluations, which gets compounded further in the case of transcriptome-wide studies where the model parameters may need to be estimated for thousands of genes. This general issue in the context of single-cell genomics is elucidated by Gorin *et al.* [219, 349] who advocate for the use of neural approximations, variants of which have been practically applied in [45, 350]. We also emphasise that analytical solutions for many detailed mechanistic models are not known, making likelihood-free inference even more important [351]. Therefore, using Nessie to model experimental gene expression data is a particularly promising future research direction.

Since the original publication of Nessie [3], several studies have further explored deep learning applications for the CME. In [352], the authors have shown how a Nessie-like approach can be used to efficiently estimate the Fisher Information matrix and perform fast policy search. In [353], a recurrent neural network (based on the previous work on approximating SDEs [299]) was used to learn the probability distribution of a simple gene network, however, it remains unclear whether the suggested approach can be easily applied to more complex systems. In [354], the authors used Generative Adversarial Networks to generate full stochastic trajectories of a system, further advancing their previous work on model abstraction [218]. Variational Autoregressive Networks were used in [355] to emulate the joint probability distribution of a chemical reaction network, but it appears that the model has to be retrained for each parameter set, limiting its applicability. Another study has utilised a transformer-based language model to learn the joint probability distributions of the CME and demonstrated promising predictive performance [356]. However, the suggested approaches have varying modelling aims and implement very different neural network architectures, making their comparison to Nessie and earlier studies far from straightforward. The field could benefit from a systematic benchmark study to clearly establish the most effective neural network methodologies for different stochastic modelling tasks.

Another possible research direction is to use neural networks to approximately close the moment equations. More specifically, the higher order moments of a chemical reaction network could be represented by a neural network which takes as its input the lower-order moments, hence effectively encoding a moment closure scheme that can be implemented in MomentClosure.jl. A system of moment ODEs with such neural network terms can be modelled using the formalism of universal differential equations [275], where the neural network can be trained on stochastic simulation data, as in the case of Nessie. Alternatively, the closure functions could be constructed in a more interpretable way using SINDy [357], a machine learning algorithm that models the target function as a sparse linear combination of the provided function terms (lower-order moments). However, one cannot guarantee the numerical stability of the resulting moment equations [358] and it remains to be seen how viable such approximations would be for stochastic chemical kinetics. We note that a few similar approaches have been studied in the literature: in [359, 360] the authors approximated the higher-order moments using the SSA trajectories and Kalman filtering, whereas neural network-based moment closures were considered in [361, 362], albeit in a different context of the lattice Lotka-Volterra model.

Finally, the approximation methods developed in this thesis are not limited to biochemical kinetics and could be applied in other fields where master equations are commonly employed. One prominent example can be found in computational neuroscience: the stochastic equivalent of the canonical Wilson-Cowan neural mass model [363, 364] describes the random spiking activity of inter-connected populations of neurons, each of which is modelled as a two-state Markov process, and the system dynamics are governed by a master equation analogous in form to the CME [365–368]. As its exact solution remains infeasible due to the highly non-linear propensity functions that characterise the neural firing rates, a number of studies have explored the parallels to the CME using the system size expansion (usually restricted to the linear noise approximation) [367–374], but the applicability of other methods such as moment closure approximations has received less attention [375–379]. The stochastic neural mass model exhibits a rich variety of dynamical regimes which makes it a useful testing ground for different approximations, and it would be interesting to investigate how the accuracy of different moment closures compares to the system size expansion across the parameter space, or similarly explore the performance of Nessie.

Appendix A

Chapter 3 appendices

A.1 Exact time-dependent solution of single enzyme system

The master equation for a single enzyme molecule (given by Eq. (3.6)) was first solved by Arányi and Tóth [132]. As the original paper is rather difficult to find, we present the solution here. The authors used marginal probability generating functions

$$G_{n_E}(z,t) = \sum_{n=0}^{N-1+n_E} z^n P(n,n_E,t) \qquad (n_E = 0,1;\ t \geq 0) \tag{A.1}$$

to transform Eq. (3.6) into the following first-order partial differential equations:

$$\begin{cases} \dfrac{\partial G_0(z,t)}{\partial t} = -(k_1+k_2)G_0(z,t) + k_0 \dfrac{\partial G_1(z,t)}{\partial z} \ , \\ \dfrac{\partial G_1(z,t)}{\partial t} = -k_0 z \dfrac{\partial G_1(z,t)}{\partial z} + k_1 z G_0(z,t) + k_2 G_0(z,t) \ . \end{cases} \tag{A.2}$$

By a simple substitution one can prove that the solutions have the form:

$$G_0(z,t) = \overline{\Gamma} e^{-\frac{k_1}{k_0}(z-1)} e^{-k_2 t} + \overline{\overline{\Gamma}} \frac{k_1+k_2}{k_1 z + k_2} e^{-(k_0+k_2)t} + \sum_{i=1}^{2} \sum_{m=0}^{\infty} \Gamma_i^{(m)} \left[\frac{k_2 - (k_2+\lambda_i^{(m)})z}{-\lambda_i^{(m)}} \right]^{q_m} e^{\lambda_i^{(m)} t} \ , \tag{A.3}$$

$$G_1(z,t) = \Gamma^{(-1)} - \overline{\Gamma} e^{-\frac{k_1}{k_0}(z-1)} e^{-k_2 t} - \overline{\overline{\Gamma}} e^{-(k_1+k_2)t} - \sum_{i=1}^{2} \sum_{m=0}^{\infty} \Gamma_i^{(m)} \left[\frac{k_2 - (k_2+\lambda_i^{(m)})z}{-\lambda_i^{(m)}} \right]^{q_m+1} e^{\lambda_i^{(m)} t} \ , \tag{A.4}$$

where

$$\lambda_i^{(m)} \neq -k_2 \ , \quad q_m = -\frac{(\lambda_i^{(m)})^2 + (k_1+k_0+k_2)\lambda_i^{(m)} + k_0 k_2}{k_0(k_2+\lambda_i^{(m)})} \ , \quad i = 1,2 \ . \tag{A.5}$$

Since G_0 and G_1 are generating functions of a system with a finite state space, i.e., the number of substrate and enzyme are bounded quantities ($n \in [0,N]$, $n_E \in [0,1]$), they must be polynomials of a finite degree in z. Hence, the summations in Eqs. (A.3) and (A.4) must contain a finite number of terms only, meaning that $\overline{\Gamma} = \overline{\overline{\Gamma}} = 0$ (if $k_1 \neq 0$). By the same reasoning the q_m must be positive integers, i.e., $0 \leq q_m \leq N-1$, ($q_m = m$), then the $\lambda^{(m)}$ are the roots of a quadratic

equation:

$$(\lambda^{(m)})^2 + [k_1 + k_0(m+1) + k_2]\,\lambda^{(m)} + k_0 k_2 (m+1) = 0, \quad (m = 0, 1, ..., N-1). \tag{A.6}$$

The constants Γ can be determined from the initial conditions:

$$\begin{aligned} &G_0(1,t) + G_1(1,t) = 1\ , \\ &G_0(z,0) = 0\ , \\ &G_1(z,0) = z^N\ . \end{aligned} \tag{A.7}$$

The first constraint implies that $\Gamma^{(-1)} = 1$, while the remaining two lead to a linear algebraic system for $\Gamma_i^{(m)}$ by enforcing the constraints explicitly on each coefficient of the polynomials G_0 and G_1 for each power of z. However, solving for $\Gamma_i^{(m)}$ becomes computationally expensive for larger values of N.

To summarise, the solution has the form:

$$\begin{aligned} G_0(z,t) &= \sum_{i=1}^{2} \sum_{m=0}^{N-1} \Gamma_i^{(m)} \left[\frac{k_2 - (k_2 + \lambda_i^{(m)})z}{-\lambda_i^{(m)}} \right]^m e^{\lambda_i^{(m)} t}\ ; \\ G_1(z,t) &= 1 - \sum_{i=1}^{2} \sum_{m=0}^{N-1} \Gamma_i^{(m)} \left[\frac{k_2 - (k_2 + \lambda_i^{(m)})z}{-\lambda_i^{(m)}} \right]^{m+1} e^{\lambda_i^{(m)} t}\ , \end{aligned} \tag{A.8}$$

where

$$\begin{aligned} \lambda_1^{(m)} &= -\frac{k_0(m+1) + k_1 + k_2}{2} + \frac{\sqrt{[k_0(m+1) + k_1 + k_2]^2 - 4k_0k_2(m+1)}}{2}\ ; \\ \lambda_2^{(m)} &= -\frac{k_0(m+1) + k_1 + k_2}{2} - \frac{\sqrt{[k_0(m+1) + k_1 + k_2]^2 - 4k_0k_2(m+1)}}{2}\ . \end{aligned} \tag{A.9}$$

Finally, the probabilities can be calculated from the generating functions according to

$$P(n, n_E, t) = \frac{1}{n!} \frac{\partial^n G_{n_E}(z,t)}{\partial z^n}\bigg|_{z=0}\ . \tag{A.10}$$

A.2 Figure showing the initial transient

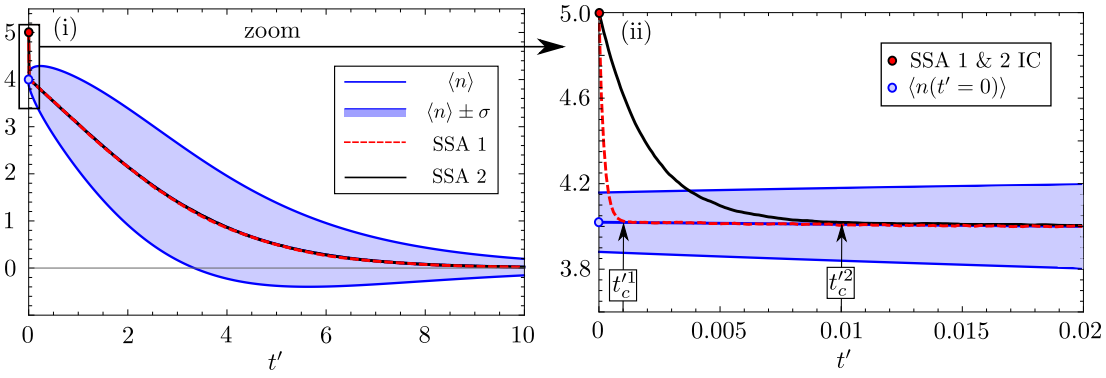

Figure A.1: Exhibition of the initial transient seen explicitly from the SSA. (i) Comparison of two differing SSA means for the same value of $k = 0.1$ against $\langle n \rangle$ from Eq. (3.22). SSA 1 was simulated with parameters $N = 5$, $M = 1$, $k_0 = 10^3$ and $k_1 = 10^2$, and SSA 2 $N = 5$, $M = 1$, $k_0 = 10^2$ and $k_1 = 10$. For most of the time course the SSA means agree with the mean predicted analytically from Eq. (3.22), aside from the initial transient very close to $t' = 0$. (ii) Zoomed in area around the initial transient. There exists some critical time t_c' for both SSA 1 and 2, denoted by $t_c'^1$ and $t_c'^2$ respectively, over which the mean predicted by the SSA relaxes to the mean value predicted by Eq. (3.22). In SSA 1, where k_0 and k_1 are a magnitude of 10 larger than the same parameters in SSA 2, one observes that the initial transient occurs over a much shorter time. *This relaxation of the SSA means to the mean predicted by the quasi-equilibrium analysis is known as the initial transient.* Dots of differing colour, seen in the legend, show the means of the stochastic QEA and SSAs at $t' = 0$ (IC in the legend refers to initial conditions). The mean predicted by the stochastic QEA in Eq. (3.22) reaches the quasi-equilibrium instantaneously at $t' = 0$, unlike that seen in the SSA. In both cases the SSA means were determined as an average over 10^5 individual reaction trajectories.

A.3 Derivation of Eq. (3.35)

In this appendix we prove the resulted stated in Eq. (3.35) of the main text. First consider the sum that defines $\mathcal{Z}_{m-1}$ explicitly:

$$\begin{aligned} \mathcal{Z}_{m-1} &= \sum_{i=0}^{M-g(m-1)} z_{i,m-1} \\ &= \sum_{i=0}^{M-g(m-1)} k^{-i} \left(\prod_{j=1}^{i} ((N-m+1) - -(j-1))(M-(j-1)) \right) \left(\prod_{j=i+1}^{M-g(m-1)} j \right). \end{aligned} \tag{A.11}$$

We now relabel $g(m-1) = Q$ for brevity and consider later the individual cases where $g(m-1) = 0$ for $m \leq N - M + 1$ and $g(m-1) = (m-1) - -(N-M)$ for $m > N - M + 1$. Using the definition of the Pochhammer function, $(x)_n = \prod_{j=0}^{n-1}(x+j)$, one can re-write Eq. (A.11) to give

$$\mathcal{Z}_{m-1} = \sum_{i=0}^{M-Q} k^{-i} (m-N-1)_i (-M)_i (i+1)_{M-Q-i}. \tag{A.12}$$

We now utilise the relation between the Pochhammer function and the Gamma function, namely $(x)_n = \Gamma(x+n)/\Gamma(x)$, which allows us to tactically write Eq. (A.12) as:

$$\mathcal{Z}_{m-1} = \frac{k^{-(M-Q)}\Gamma(-Q)\Gamma(m+M-N-Q-1)}{\Gamma(-M)\Gamma(m-N-1)} \times \mathcal{S}, \tag{A.13}$$

where $\mathcal{S}$ is defined by the sum,

$$\mathcal{S} = \sum_{i=0}^{M-Q} k^{M-Q-i} \frac{\Gamma(m-N-1+i)\Gamma(i-M)}{\Gamma(m+M-N-Q-1)\Gamma(-Q)} (i+1)_{M-Q-i}. \tag{A.14}$$

Our task is now to find an analytic function that is equal to the sum $\mathcal{S}$. Motivated by the Pochhammer and Gamma functions contained within the sum, we look to match this sum to the definition of a generalised hypergeometric function ${}_pF_q(\{\alpha_1, \alpha_2, ..., \alpha_{L_1}\}, \{\beta_1, \beta_2, ..., \beta_{L_2}\}, z)$ defined by:

$${}_pF_q(\{\alpha_1, \alpha_2, ..., \alpha_{L_1}\}, \{\beta_1, \beta_2, ..., \beta_{L_2}\}; z) = \sum_{n=0}^{\infty} \left(\frac{\prod_{l=1}^{L_1} (\alpha_l)_n}{\prod_{l=1}^{L_2} (\beta_l)_n} \times \frac{z^n}{n!} \right). \tag{A.15}$$

We begin by relabelling the summation index in Eq. (A.14) by $j = M-Q-i$ and again utilising the definition of the Pochhammer function in terms of Gamma functions, giving us

$$\mathcal{S} = \sum_{j=0}^{M-Q} k^j (M-Q+1-j)_j (m-N-1+M-Q)_{-j} (-Q)_{-j}. \tag{A.16}$$

Consider now the latter two Pochhammer functions in the summand of Eq. (A.16). Using the relation $(b)_{-n} = (-1)^n/(1-b)_n$ we find that:

$$(m-N-1+M-Q)_{-j} \times (-Q)_{-j} = \frac{1}{(Q+1)_j (Q+N+2-m-M)_j}. \tag{A.17}$$

Now consider the first Pochhammer function in the summand of Eq. (A.16). One can re-write this as:

$$(M-Q+1-j)_j = (-1)^j (Q-M)_j. \tag{A.18}$$

Note that $(Q-M)_j$ has the property $(Q-M)_{j>M-Q} = 0$, which is found trivially from the definition of the Pochhammer function. Using Eqs. (A.17) and (A.18), and the relation $j! = (1)_j$, one can then show that:

$$\begin{aligned} \mathcal{S} &= \sum_{j=0}^{\infty} \left(\frac{(1)_j (Q-M)_j}{(Q+1)_j (Q+N+2-m-M)_j} \times \frac{(-k)^j}{j!} \right), \\ &= {}_2F_2(\{1, Q-M\}, \{Q+1, Q+N+2-m-M\}; -k), \end{aligned} \tag{A.19}$$

using the definition of the generalised hypergeometric function in Eq. (A.15). Note, one is able to extend the upper limit of the sum defining $\mathcal{S}$ to infinity due to the property $(Q-M)_{j>M-Q} = 0$. One finds that $\mathcal{Z}_{m-1}$ is now fully specified by Eqs. (A.13) and (A.19), and we can now return to our original problem of finding the group transition rates a_m in Eq. (3.35).

In order to find a_m we must now compute $a_m = -k\partial_k(\ln(\mathcal{Z}_{m-1}))$, which using the chain rule and the differentiation rules for generalised hypergeometric functions gives:

$$\begin{aligned} a_m =&(M-Q) \\ &- \frac{k(Q-M)\,{}_2F_2(\{2, Q-M+1\},\{Q+2, Q+N+3-m-M\};-k)}{(Q+1)(Q+N+2-m-M)\,{}_2F_2(\{1, Q-M\},\{Q+1, Q+N+2-m-M\};-k)}. \end{aligned} \tag{A.20}$$

Where $m \leq N-M+1$, $Q=0$, and Eq. (A.20) becomes:

$$a_m = -M \times \left(\frac{k\,{}_1F_1(1-M, -m-M+N+3;-k)}{(-m-M+N+2)\,{}_1F_1(-M, -m-M+N+2;-k)} - 1 \right), \tag{A.21}$$

noting that for $Q=0$ the ${}_2F_2(...)$ general hypergeometric function reduces to the ${}_1F_1(...)$ confluent hypergeometric function. And finally, where $m > N-M+1$, $Q=(m-1)--(N-M)$, and Eq. (A.20) becomes:

$$a_m = -(N-m+1) \times \left(\frac{k\,{}_1F_1(m-N, m+M-N+1;-k)}{(m+M-N)\,{}_1F_1(m-N-1, m+M-N;-k)} - 1 \right), \tag{A.22}$$

where again the ${}_2F_2(...)$ general hypergeometric function reduces to the ${}_1F_1(...)$ confluent hypergeometric function. This completes the derivation of Eq. (3.35) from the main text.

Appendix B

Chapter 5 appendices

Model parameters

Parameter	t	σ_u	σ_b	ρ_u	ρ_b
Range	-	0-2	0-0.1	0-10	0-100
Fig. 3B-(i)	10	0.94	0.01	8.40	28.1
Fig. 3B-(ii)	25	0.69	0.07	7.20	40.6
Fig. 3B-(iii)	100	0.44	0.08	0.94	53.1
Fig. 3B	10	1.17	0.05	6.90	47.9

Table B.1: Parameters and parameter ranges for the autoregulatory feedback loop, related to Figure 5.3. The initial conditions are zero proteins with the gene in the unbound state G_u.

Parameter	t	σ_b^B	σ_u^B	σ_b^A	σ_u^A	ρ_u^A	ρ_b^A	γ^A	δ_m^A	δ_p	ρ_u^B	ρ_b^B	γ^B	δ_m^B	σ_u^M	σ_b^M	δ_{pm}
Range	-	0–5×10^{-4}	0–0.1	0–5×10^{-4}	0–0.1	0–500	0–500	0–12	1–20	0–2	0–500	0–500	0–12	1–20	0–2	0–0.5	0–100
Fig. 5B-(i)	35	5×10^{-4}	0.070	1.6×10^{-4}	0.0680	367	119.9	5.3	14.36	1.35	413	472.3	11.9	1.74	0.98	0.43	65.6
Fig. 5B-(ii)	16	2×10^{-4}	0.060	4.8×10^{-4}	0.0095	328	190.2	11.1	15.84	1.01	467	11.3	8.7	13.91	1.14	0.17	92.2
Fig. 5B-(iii)	70	3×10^{-4}	0.018	0.2×10^{-4}	0.0200	222	474.9	4.0	6.96	0.68	360	82.1	4.2	7.13	0.10	0.41	68.4
Fig. 5B-(iv)	100	2×10^{-4}	0.005	3.5×10^{-4}	0.0460	371	69.1	7.1	13.90	0.87	128	452.7	8.4	11.98	1.10	0.20	72.6
Fig. 6	-	4×10^{-4}	0.060	2.2×10^{-4}	0.0034	270	61.3	8.0	9.46	1.78	308	351.2	10.1	12.27	0.09	0.38	85.6

Table B.2: Parameters and parameter ranges for the genetic toggle switch, related to Figure 5.5. The initial conditions are zero proteins/mRNAs with the genes in the unbound states G_u^A and G_u^B respectively.

Parameter	t	σ_u	σ_b	k_2	k_3	k_4	k_5	k_8	k_9	k_{10}
Range	-	0.0001–0.1	0.0001–0.1	0.1–2.0	0.001–0.02	0.5–4.0	0.0001–0.01	0.001–0.02	0.001–0.02	0.01–0.2
Fig. 7B-(i)	750	0.009	0.056	1.22	0.015	0.81	0.006	0.009	0.003	0.78
Fig. 7B-(ii)	500	0.016	0.0214	0.603	0.0021	2.338	0.00037	0.00191	0.0163	0.0128

Parameter	k_{11}	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	
Range	0.001–0.01	0.01–0.1	0.01–0.1	0.01–0.1	0.01–0.1	0.01–0.1	0.01–0.05	0.001–0.01	0.001–0.01	
Fig. 7B-(i)	0.001	0.057	0.036	0.017	0.039	0.069	0.03	0.003	0.007	
Fig. 7B-(ii)	0.003	0.0169	0.0286	0.0151	0.062	0.0333	0.0461	0.0013	0.0015	

Table B.3: Parameters and parameter ranges for the mRNA degradation model, related to Figure 5.7. The initial conditions are zero mRNA fragments with the gene in the active state G_{ON}.

Parameter	b	h	K_m	$V_{\max}$	η_u	σ_b	σ_u	ρ_m	ρ_p	δ_m	δ_p
Range	0.001–0.1	1–10	0.01–1	0.1–10	0.1–10	10^{-4}–10^{-2}	0.001–0.1	0.1–1	0.001–0.01	10^{-4}–10^{-3}	10^{-5}–10^{-3}
Fig. 8B	0.0070	7.3	0.196	3.16	3.12	28.19×10^{-4}	0.054	0.555	0.00661	4.31×10^{-4}	45.4×10^{-4}
	0.0186	3.7	0.184	0.95	2.92	8.17×10^{-4}	0.027	0.353	0.00906	7.68×10^{-4}	28.7×10^{-4}
	0.0101	1.5	0.078	0.71	2.09	2.29×10^{-4}	0.008	0.456	0.00518	9.73×10^{-4}	4.97×10^{-4}
	0.0182	3.1	0.227	1.16	1.21	5.87×10^{-4}	0.036	0.334	0.00982	3.88×10^{-4}	2.01×10^{-4}
	0.0315	8.7	0.190	1.63	0.31	3.81×10^{-4}	0.025	0.368	0.00757	8.48×10^{-4}	45.3×10^{-4}
	0.0056	1.5	0.072	0.86	2.20	2.18×10^{-4}	0.010	0.394	0.00605	6.73×10^{-4}	5.22×10^{-4}
	0.0177	7.5	0.175	0.41	2.24	56.15×10^{-4}	0.047	0.507	0.00634	2.62×10^{-4}	2.19×10^{-4}
	0.0252	5.8	0.232	0.68	0.27	8.06×10^{-4}	0.031	0.459	0.00632	2.66×10^{-4}	4.70×10^{-4}
	0.0108	1.9	0.075	0.55	1.47	2.06×10^{-4}	0.008	0.381	0.00584	7.54×10^{-4}	5.22×10^{-4}
	0.0278	3.6	0.300	3.56	3.40	12.88×10^{-4}	0.055	0.453	0.00982	4.09×10^{-4}	1.45×10^{-4}

Table B.4: Search ranges and 10 estimated parameter sets for the MAPK pathway, related to Figure 5.8. The initial conditions were zero mRNA and proteins, 225 CR molecules with the gene in the deactivated state G^{**}.

Runtime performance comparison

Model	Feedback Loop (FSP)	Feedback Loop (SSA)	Toggle Switch	Model of mRNA Turnover	MAPK Pathway
Training data	38 s	153 s	31 h	5 h	3 h
Validation data	4 s	135 s	7.3 h	0.4 h	1.8 h
Test data	19 s	727 s	83.3 h	4 h	0.7 h
Training	126 s	126 s	6 h	23.7 h	8 h

Table B.5: Execution times required to generate the training, validation and test datasets and train the neural network for each chemical reaction network presented in the paper: the autoregulatory feedback loop (Section 5.5.1), the genetic toggle switch (Section 5.5.2), the MAPK pathway (Section 5.5.4) and the model of mRNA turnover (Section 5.5.3). The details on the dataset size, the number of SSA trajectories used and the neural network architecture are provided for each model in the corresponding text sections. Note that for the autoregulatory feedback loop we show the results for both when the datasets are constructed using the FSP and the SSA (using 10^3 trajectories for the training and 10^4 trajectories for the validation and test datasets).

Bibliography

[1] James Holehouse, Augustinas Sukys, and Ramon Grima. Stochastic time-dependent enzyme kinetics: Closed-form solution and transient bimodality. *The Journal of Chemical Physics*, 153(16):164113, 2020.

[2] Augustinas Sukys and Ramon Grima. MomentClosure.jl: automated moment closure approximations in Julia. *Bioinformatics*, 38(1):289–290, 2021.

[3] Augustinas Sukys, Kaan Öcal, and Ramon Grima. Approximating solutions of the Chemical Master equation using neural networks. *iScience*, 25:105010, 2022.

[4] Vahid Shahrezaei and Peter S. Swain. The stochastic nature of biochemical networks. *Current Opinion Biotechnology*, 19(4):369–374, August 2008.

[5] N.G. Van Kampen. *Stochastic Processes in Physics and Chemistry*. Elsevier, 3rd edition, 2007.

[6] Kerson Huang. *Statistical Mechanics*. John Wiley & Sons, 2 edition, 1987.

[7] Lev S. Tsimring. Noise in biology. *Reports on Progress in Physics*, 77(2):026601, January 2014.

[8] Ramon Grima and Santiago Schnell. Modelling reaction kinetics inside cells. *Essays in Biochemistry*, 45:41, 2008.

[9] Brian P. Ingalls. *Mathematical Modeling in Systems Biology: An Introduction*. MIT Press, London, England, UK, August 2013.

[10] Arjun Raj and Alexander van Oudenaarden. Nature, Nurture, or Chance: Stochastic Gene Expression and Its Consequences. *Cell*, 135(2):216–226, October 2008.

[11] Yasushi Ishihama, Thorsten Schmidt, Juri Rappsilber, Matthias Mann, F Ulrich Hartl, Michael J Kerner, and Dmitrij Frishman. Protein abundance profiling of the *Escherichia coli* cytosol. *BMC genomics*, 9(1):102, 2008.

[12] Yuichi Taniguchi, Paul J. Choi, Gene-Wei Li, Huiyi Chen, Mohan Babu, Jeremy Hearn, Andrew Emili, and X. Sunney Xie. Quantifying E. coli proteome and transcriptome with single-molecule sensitivity in single cells. *Science*, 329(5991):533–538, July 2010.

[13] Lok-hang So, Anandamohan Ghosh, Chenghang Zong, Leonardo A. Sepúlveda, Ronen Segev, and Ido Golding. General properties of transcriptional time series in Escherichia coli. *Nat. Genet.*, 43:554–560, June 2011.

[14] Darren J. Wilkinson. *Stochastic Modelling for Systems Biology*. Chapman and Hall/CRC, Boca Raton, 3rd edition edition, November 2018.

[15] Mukhtar Ullah and Olaf Wolkenhauer. *Stochastic Approaches for Systems Biology*. Springer, New York, NY, USA, 2011.

[16] Michael B. Elowitz, Arnold J. Levine, Eric D. Siggia, and Peter S. Swain. Stochastic gene expression in a single cell. *Science*, 297(5584):1183–1186, August 2002.

[17] Peter S Swain, Michael B Elowitz, and Eric D Siggia. Intrinsic and extrinsic contributions to stochasticity in gene expression. *Proceedings of the National Academy of Sciences*, 99(20):12795–12800, 2002.

[18] Ertugrul M. Ozbudak, Mukund Thattai, Iren Kurtser, Alan D. Grossman, and Alexander van Oudenaarden. Regulation of noise in the expression of a single gene. *Nature Genetics*, 31:69–73, May 2002.

[19] William J Blake, Mads Kærn, Charles R Cantor, and James J Collins. Noise in eukaryotic gene expression. *Nature*, 422(6932):633–637, 2003.

[20] Harley H. McAdams and Adam Arkin. Stochastic mechanisms in gene expression. *Proceedings of the National Academy of Sciences*, 94(3):814–819, February 1997.

[21] Mads Kaern, Timothy C Elston, William J Blake, and James J Collins. Stochasticity in gene expression: from theories to phenotypes. *Nature Reviews Genetics*, 6(6):451–464, 2005.

[22] Ido Golding, Johan Paulsson, Scott M. Zawilski, and Edward C. Cox. Real-Time Kinetics of Gene Activity in Individual Bacteria. *Cell*, 123(6):1025–1036, December 2005.

[23] Arjun Raj, Charles S Peskin, Daniel Tranchina, Diana Y Vargas, and Sanjay Tyagi. Stochastic mRNA synthesis in mammalian cells. *PLoS Biology*, 4(10):e309, 2006.

[24] Jonathan R. Chubb, Tatjana Trcek, Shailesh M. Shenoy, and Robert H. Singer. Transcriptional Pulsing of a Developmental Gene. *Current Biology*, 16(10):1018–1025, May 2006.

[25] Damien Nicolas, Nick E Phillips, and Felix Naef. What shapes eukaryotic transcriptional bursting? *Molecular BioSystems*, 13(7):1280–1290, 2017.

[26] Edward Tunnacliffe and Jonathan R. Chubb. What Is a Transcriptional Burst? *Trends Genet.*, 36(4):288–297, April 2020.

[27] Jean Peccoud and Bernard Ycart. Markovian modeling of gene-product synthesis. *Theoretical population biology*, 48(2):222–234, 1995.

[28] Jonathan M. Raser and Erin K. O'Shea. Control of Stochasticity in Eukaryotic Gene Expression. *Science*, 304(5678):1811–1814, June 2004.

[29] Srividya Iyer-Biswas, Fernand Hayot, and Ciriyam Jayaprakash. Stochasticity of gene products from transcriptional pulsing. *Physical Review E*, 79(3):031911, 2009.

[30] Samuel O Skinner, Leonardo A Sepúlveda, Heng Xu, and Ido Golding. Measuring mRNA copy number in individual *Escherichia coli* cells using single-molecule fluorescent *in situ* hybridization. *Nature protocols*, 8(6):1100–1113, 2013.

[31] Xavier Pichon, Mounia Lagha, Florian Mueller, and Edouard Bertrand. A Growing Toolbox to Image Gene Expression in Single Cells: Sensitive Approaches for Demanding Challenges. *Mol. Cell*, 71(3):468–480, August 2018.

[32] Byungjin Hwang, Ji Hyun Lee, and Duhee Bang. Single-cell RNA sequencing technologies and bioinformatics pipelines. *Exp. Mol. Med.*, 50:1–14, August 2018.

[33] Tim Stuart and Rahul Satija. Integrative single-cell analysis. *Nature Reviews Genetics*, 20:257–272, May 2019.

[34] Vahid Shahrezaei and Peter S. Swain. Analytical distributions for stochastic gene expression. *Proceedings of the National Academy of Sciences*, 105(45):17256–17261, November 2008.

[35] Daniel Zenklusen, Daniel R Larson, and Robert H Singer. Single-RNA counting reveals alternative modes of gene expression in yeast. *Nature structural & molecular biology*, 15(12):1263, 2008.

[36] Jong Kyoung Kim and John C. Marioni. Inferring the kinetics of stochastic gene expression from single-cell RNA-sequencing data. *Genome Biol.*, 14(1):1–12, January 2013.

[37] Keren B Halpern, Sivan Tanami, Shanie Landen, Michal Chapal, Liran Szlak, Anat Hutzler, Anna Nizhberg, and Shalev Itzkovitz. Bursty gene expression in the intact mammalian liver. *Molecular cell*, 58(1):147–156, 2015.

[38] Anton JM Larsson, Per Johnsson, Michael Hagemann-Jensen, Leonard Hartmanis, Omid R Faridani, Björn Reinius, Åsa Segerstolpe, Chloe M Rivera, Bing Ren, and Rickard Sandberg. Genomic encoding of transcriptional burst kinetics. *Nature*, 565(7738):251–254, 2019.

[39] Samuel O Skinner, Heng Xu, Sonal Nagarkar-Jaiswal, Pablo R Freire, Thomas P Zwaka, and Ido Golding. Single-cell analysis of transcription kinetics across the cell cycle. *Elife*, 5:e12175, 2016.

[40] Ruben Perez-Carrasco, Casper Beentjes, and Ramon Grima. Effects of cell cycle variability on lineage and population measurements of messenger RNA abundance. *Journal of the Royal Society Interface Interface*, 17(168):20200360, July 2020.

[41] Zhixing Cao and Ramon Grima. Analytical distributions for detailed models of stochastic gene expression in eukaryotic cells. *Proceedings of the National Academy of Sciences*, 117(9):4682–4692, March 2020.

[42] Gennady Gorin, John J. Vastola, Meichen Fang, and Lior Pachter. Interpretable and tractable models of transcriptional noise for the rational design of single-molecule quantification experiments. *Nature Communications*, 13(7620):1–13, December 2022.

[43] Chen Jia and Ramon Grima. Coupling gene expression dynamics to cell size dynamics and cell cycle events: Exact and approximate solutions of the extended telegraph model. *iScience*, 26(1):105746, January 2023.

[44] X. Fu, H. P. Patel, S. Coppola, L. Xu, Z. Cao, T. L. Lenstra, and R. Grima. Quantifying how post-transcriptional noise and gene copy number variation bias transcriptional parameter inference from mRNA distributions. *eLife*, 11, 2023.

[45] Wenhao Tang, Andreas Christ Sølvsten Jørgensen, Samuel Marguerat, Philipp Thomas, and Vahid Shahrezaei. Modelling capture efficiency of single-cell RNA-sequencing data improves inference of transcriptome-wide burst kinetics. *Bioinformatics*, 39(7):btad395, July 2023.

[46] Ulysse Herbach, Arnaud Bonnaffoux, Thibault Espinasse, and Olivier Gandrillon. Inferring gene regulatory networks from single-cell data: a mechanistic approach. *BMC systems biology*, 11(1):105, 2017.

[47] Chen Jia, Peng Xie, Min Chen, and Michael Q Zhang. Stochastic fluctuations can reveal the feedback signs of gene regulatory networks at the single-molecule level. *Scientific reports*, 7(1):1–9, 2017.

[48] James Holehouse, Zhixing Cao, and Ramon Grima. Stochastic modeling of autoregulatory genetic feedback loops: A review and comparative study. *Biophysical Journal*, 118(7):1517–1525, 2020.

[49] Lucy Ham, David Schnoerr, Rowan D Brackston, and Michael PH Stumpf. Exactly solvable models of stochastic gene expression. *The Journal of Chemical Physics*, 152(14):144106, 2020.

[50] Chen Jia and Youming Li. Analytical Time-Dependent Distributions for Gene Expression Models With Complex Promoter Switching Mechanisms. *SIAM Journal on Applied Mathematics*, August 2023.

[51] Svitlana Braichenko, James Holehouse, and Ramon Grima. Distinguishing between models of mammalian gene expression: telegraph-like models versus mechanistic models. *Journal of The Royal Society Interface*, 18(183):20210510, 2021.

[52] Tatiana Filatova, Nikola Popović, and Ramon Grima. Statistics of nascent and mature RNA fluctuations in a stochastic model of transcriptional initiation, elongation, pausing, and termination. *Bulletin of Mathematical Biology*, 83(1):1–62, 2021.

[53] Juraj Szavits-Nossan and Ramon Grima. Mean-field theory accurately captures the variation of copy number distributions across the mRNA life cycle. *Physical Review E*, 105(1):014410, January 2022.

[54] Leonor Michaelis and Maud L Menten. Die kinetik der invertinwirkung. *Biochem Z*, 49:333–369, 1913.

[55] Santiago Schnell and Philip K Maini. A century of enzyme kinetics. Should we believe in the K_M and v_{max} estimates? *Comments on Theoretical Biology*, 8:169–187, 2003.

[56] Athel Cornish-Bowden. One hundred years of Michaelis–Menten kinetics. *Perspectives in Science*, 4:3–9, 2015.

[57] H. Peter Lu, Luying Xun, and X. Sunney Xie. Single-Molecule Enzymatic Dynamics. *Science*, 282(5395):1877–1882, December 1998.

[58] X. Sunney Xie and H. Peter Lu. Single-molecule Enzymology∗. *J. Biol. Chem.*, 274(23):15967–15970, June 1999.

[59] Hong Qian and Elliot L. Elson. Single-molecule enzymology: stochastic Michaelis–Menten kinetics. *Biophysical Chemistry*, 101-102:565–576, December 2002.

[60] S. C. Kou, Binny J. Cherayil, Wei Min, Brian P. English, and X. Sunney Xie. Single-Molecule Michaelis-Menten Equations. *The Journal of Physical Chemistry B*, 109(41):19068–19081, October 2005.

[61] Brian P English, Wei Min, Antoine M van Oijen, Kang T Lee, Guobin Luo, Hongye Sun, Binny J Cherayil, SC Kou, and X Sunney Xie. Ever-fluctuating single enzyme molecules: Michaelis-Menten equation revisited. *Nature chemical biology*, 2(2):87–94, 2006.

[62] Christopher V Rao and Adam P Arkin. Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the Gillespie algorithm. *The Journal of Chemical Physics*, 118(11):4999–5010, 2003.

[63] Ramon Grima. Investigating the robustness of the classical enzyme kinetic equations in small intracellular compartments. *BMC systems biology*, 3(1):101, 2009.

[64] Kevin R Sanft, Daniel T Gillespie, and Linda R Petzold. Legitimacy of the stochastic Michaelis-Menten approximation. *IET systems biology*, 5(1):58–69, 2011.

[65] Éva Dóka and Gábor Lente. Stochastic mapping of the Michaelis-Menten mechanism. *The Journal of Chemical Physics*, 136(5):054111, 2012.

[66] Ramon Grima, Nils G Walter, and Santiago Schnell. Single-molecule enzymology à la Michaelis–Menten. *The FEBS journal*, 281(2):518–530, 2014.

[67] Jeffrey R Moffitt and Carlos Bustamante. Extracting signal from noise: kinetic mechanisms from a Michaelis–Menten-like expression for enzymatic fluctuations. *The FEBS journal*, 281(2):498–517, 2014.

[68] Shlomi Reuveni, Michael Urbakh, and Joseph Klafter. Role of substrate unbinding in Michaelis–Menten enzymatic reactions. *PNAS*, 111(12):4391–4396, March 2014.

[69] Ramon Grima and André Leier. Exact product formation rates for stochastic enzyme kinetics. *The Journal of Physical Chemistry B*, 121(1):13–23, 2017.

[70] Hye-Won Kang, Wasiur R KhudaBukhsh, Heinz Koeppl, and Grzegorz A Rempała. Quasi-steady-state approximations derived from the stochastic model of enzyme kinetics. *Bulletin of Mathematical Biology*, 81(5):1303–1336, 2019.

[71] Dawn Fraser and Mads Kærn. A chance at survival: gene expression noise and phenotypic diversification strategies. *Molecular Microbiology*, 71(6):1333–1340, March 2009.

[72] Avigdor Eldar and Michael B. Elowitz. Functional roles for noise in genetic circuits. *Nature*, 467:167–173, September 2010.

[73] Coco van Boxtel, Johan H. van Heerden, Niclas Nordholt, Phillipp Schmidt, and Frank J. Bruggeman. Taking chances and making mistakes: non-genetic phenotypic heterogeneity and its consequences for surviving in dynamic environments. *Journal of The Royal Society Interface*, 14(132), July 2017.

[74] Paul C Bressloff. Stochastic switching in biology: from genotype to phenotype. *Journal of Physics A: Mathematical and Theoretical*, 50(13):133001, 2017.

[75] Nils Eling, Michael D. Morgan, and John C. Marioni. Challenges in measuring and understanding biological noise. *Nature Reviews Genetics*, 20:536–548, September 2019.

[76] Ertugrul M. Ozbudak, Mukund Thattai, Han N. Lim, Boris I. Shraiman, and Alexander van Oudenaarden. Multistability in the lactose utilization network of Escherichia coli. *Nature*, 427:737–740, February 2004.

[77] Jerome T. Mettetal, Dale Muzzey, Juan M. Pedraza, Ertugrul M. Ozbudak, and Alexander van Oudenaarden. Predicting stochastic gene expression dynamics in single cells. *Proceedings of the National Academy of Sciences*, 103(19):7304–7309, May 2006.

[78] Paul J. Choi, Long Cai, Kirsten Frieda, and X. Sunney Xie. A stochastic single-molecule event triggers phenotype switching of a bacterial cell. *Science*, 322(5900):442–446, October 2008.

[79] Nathalie Q. Balaban, Jack Merrin, Remy Chait, Lukasz Kowalik, and Stanislas Leibler. Bacterial Persistence as a Phenotypic Switch. *Science*, 305(5690):1622–1625, September 2004.

[80] Martin Ackermann. A functional perspective on phenotypic heterogeneity in microorganisms. *Nature Reviews Microbiology*, 13:497–508, August 2015.

[81] Tom Philippi and Jon Seger. Hedging one's evolutionary bets, revisited. *Trends in ecology & evolution*, 4(2):41–44, 1989.

[82] Murat Acar, Jerome T. Mettetal, and Alexander van Oudenaarden. Stochastic switching as a survival strategy in fluctuating environments. *Nature Genetics*, 40:471–475, April 2008.

[83] Jan-Willem Veening, Eric J. Stewart, Thomas W. Berngruber, François Taddei, Oscar P. Kuipers, and Leendert W. Hamoen. Bet-hedging and epigenetic inheritance in bacterial cell development. *Proceedings of the National Academy of Sciences*, 105(11):4393–4398, March 2008.

[84] Jonathan R. Russell, Matthew T. Cabeen, Paul A. Wiggins, Johan Paulsson, and Richard Losick. Noise in a phosphorelay drives stochastic entry into sporulation in Bacillus subtilis. *EMBO Journal*, August 2017.

[85] Piyush B. Gupta, Christine M. Fillmore, Guozhi Jiang, Sagi D. Shapira, Kai Tao, Charlotte Kuperwasser, and Eric S. Lander. Stochastic State Transitions Give Rise to Phenotypic Equilibrium in Populations of Cancer Cells. *Cell*, 146(4):633–644, August 2011.

[86] Niraj Kumar, Gwendolyn M. Cramer, Seyed Alireza Zamani Dahaj, Bala Sundaram, Jonathan P. Celli, and Rahul V. Kulkarni. Stochastic modeling of phenotypic switching and chemoresistance in cancer cell populations. *Scientific Reports*, 9(10845):1–10, July 2019.

[87] Matthew Scott, Terence Hwa, and Brian Ingalls. Deterministic characterization of stochastic genetic circuits. *Proceedings of the National Academy of Sciences*, 104(18):7402–7407, May 2007.

[88] Ramon Grima. An effective rate equation approach to reaction kinetics in small volumes: Theory and application to biochemical reactions in non-equilibrium steady-state conditions. *The Journal of Chemical Physics*, 133(3):07B604, 2010.

[89] Rajesh Ramaswamy, Nélido González-Segredo, Ivo F Sbalzarini, and Ramon Grima. Discreteness-induced concentration inversion in mesoscopic chemical systems. *Nature Communications*, 3:779, 2012.

[90] Sayuri K. Hahl and Andreas Kremling. A Comparison of Deterministic and Stochastic Modeling Approaches for Biochemical Reaction Systems: On Fixed Points, Means, and Modes. *Frontiers in Genetics*, 7:185360, August 2016.

[91] Michael S. Samoilov and Adam P. Arkin. Deviant effects in molecular reaction pathways. *Nature Biotechnology*, 24:1235–1240, October 2006.

[92] Lucy Ham, Megan A. Coomer, Kaan Öcal, Ramon Grima, and Michael P. H. Stumpf. A stochastic vs deterministic perspective on the timing of cellular events. *Nature Communications*, 15(5286):1–10, June 2024.

[93] José M. G. Vilar, Hao Yuan Kueh, Naama Barkai, and Stanislas Leibler. Mechanisms of noise-resistance in genetic oscillators. *Proceedings of the National Academy of Sciences*, 99(9):5988–5992, April 2002.

[94] A. J. McKane, J. D. Nagy, T. J. Newman, and M. O. Stefanini. Amplified Biochemical Oscillations in Cellular Systems. *Journal of Statistical Physics*, 128(1):165–191, July 2007.

[95] Pål O. Westermark, David K. Welsh, Hitoshi Okamura, and Hanspeter Herzel. Quantification of Circadian Rhythms in Single Cells. *PLOS Computational Biology*, 5(11):e1000580, November 2009.

[96] Tanya L. Leise, Connie W. Wang, Paula J. Gitis, and David K. Welsh. Persistent Cell-Autonomous Circadian Oscillations in Fibroblasts Revealed by Six-Week Single-Cell Imaging of PER2::LUC Bioluminescence. *PLOS ONE*, 7(3):e33334, March 2012.

[97] Philipp Thomas, Arthur V. Straube, Jens Timmer, Christian Fleck, and Ramon Grima. Signatures of nonlinearity in single cell noise-induced oscillations. *Journal of Theoretical Biology*, 335:222–234, October 2013.

[98] D. L. K. Toner and R. Grima. Molecular noise induces concentration oscillations in chemical systems with stable node steady states. *Journal of Chemical Physics*, 138(5), February 2013.

[99] Chris N. Micklem and James C. W. Locke. Cut the noise or couple up: Coordinating circadian and synthetic clocks. *iScience*, 24(9), September 2021.

[100] Andrew D. Keller. Model genetic circuits encoding autoregulatory transcription factors. *Journal of Theoretical Biology*, 172(2):169–185, January 1995.

[101] Rajesh Karmakar and Indrani Bose. Positive feedback, stochasticity and genetic competence. *Physical Biology*, 4(1):29, March 2007.

[102] Maxim N. Artyomov, Jayajit Das, Mehran Kardar, and Arup K. Chakraborty. Purely stochastic binary decisions in cell signaling models without underlying deterministic bistabilities. *Proceedings of the National Academy of Sciences*, 104(48):18958–18963, November 2007.

[103] Timothy S. Gardner, Charles R. Cantor, and James J. Collins. Construction of a genetic toggle switch in Escherichia coli. *Nature*, 403(6767):339–342, January 2000.

[104] Hideki Kobayashi, Mads Kærn, Michihiro Araki, Kristy Chung, Timothy S. Gardner, Charles R. Cantor, and James J. Collins. Programmable cells: Interfacing natural and engineered gene networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(22):8414, June 2004.

[105] Michael Samoilov, Sergey Plyasunov, and Adam P. Arkin. Stochastic amplification and signaling in enzymatic futile cycles through noise-induced bistability with oscillations. *Proceedings of the National Academy of Sciences*, 102(7):2310–2315, February 2005.

[106] Hong Qian, Pei-Zhe Shi, and Jianhua Xing. Stochastic bifurcation, slow fluctuations, and bistability as an origin of biochemical complexity. *Physical Chemistry Chemical Physics*, 11(24):4861–4870, June 2009.

[107] Tsz-Leung To and Narendra Maheshri. Noise Can Induce Bimodality in Positive Transcriptional Feedback Loops Without Bistability. *Science*, 327(5969):1142–1145, February 2010.

[108] Anna Ochab-Marcinek and Marcin Tabaka. Bimodal gene expression in noncooperative regulatory systems. *Proceedings of the National Academy of Sciences*, 107(51):22096–22101, December 2010.

[109] R. Grima, D. R. Schmidt, and T. J. Newman. Steady-state fluctuations of a genetic feedback loop: An exact solution. *The Journal of Chemical Physics*, 137(3):035104, July 2012.

[110] Philipp Thomas, Nikola Popovi, and Ramon Grima. Phenotypic switching in gene regulatory networks. *Proceedings of the National Academy of Sciences*, 111(19):6994–6999, May 2014.

[111] John K. McSweeney and Lea Popovic. Stochastically-induced bistability in chemical reaction systems. *Annals of Applied Probability*, 24(3):1226–1268, June 2014.

[112] M. Ali Al-Radhawi, Domitilla Del Vecchio, and Eduardo D. Sontag. Multi-modality in gene regulatory networks with slow promoter kinetics. *PLOS Computational Biology*, 15(2):e1006784, February 2019.

[113] Jessica C. Mar. The rise of the distributions: why non-normality is important for understanding the transcriptome and beyond. *Biophysical Reviews*, 11(1):89, February 2019.

[114] Michał Komorowski, Maria J. Costa, David A. Rand, and Michael P. H. Stumpf. Sensitivity, robustness, and identifiability in stochastic chemical kinetics models. *Proceedings of the National Academy of Sciences*, 108(21):8645–8650, May 2011.

[115] Alexander P. Browning, David J. Warne, Kevin Burrage, Ruth E. Baker, and Matthew J. Simpson. Identifiability analysis for stochastic differential equation models in systems biology. *Journal of the Royal Society Interface*, 17(173), December 2020.

[116] Zhixing Cao and Ramon Grima. Accuracy of parameter estimation for auto-regulatory transcriptional feedback loops from noisy data. *Journal of The Royal Society Interface*, 16:20180967, 2019.

[117] Brian Munsky, Brooke Trinh, and Mustafa Khammash. Listening to the noise: random fluctuations reveal gene network parameters. *Molecular Systems Biology*, October 2009.

[118] Florian Hartig, Justin M. Calabrese, Björn Reineking, Thorsten Wiegand, and Andreas Huth. Statistical inference for stochastic simulation models – theory and application. *Ecology Letters*, 14(8):816–827, August 2011.

[119] Brian Munsky, Guoliang Li, Zachary R. Fox, Douglas P. Shepherd, and Gregor Neuert. Distribution shapes govern the discovery of predictive models for gene regulation. *Proceedings of the National Academy of Sciences*, 115(29):7533–7538, July 2018.

[120] Kaan Öcal, Michael U. Gutmann, Guido Sanguinetti, and Ramon Grima. Inference and uncertainty quantification of stochastic gene expression via synthetic models. *Journal of the Royal Society Interface*, 19(192):20220153, July 2022.

[121] Xiaoming Fu, Heta P. Patel, Stefano Coppola, Libin Xu, Zhixing Cao, Tineke L. Lenstra, and Ramon Grima. Quantifying how post-transcriptional noise and gene copy number variation bias transcriptional parameter inference from mrna distributions. *bioRxiv*, 2022.

[122] Ramon Grima and Pierre-Marie Esmenjaud. Quantifying and correcting bias in transcriptional parameter inference from single-cell data. *Biophysical Journal*, 123(1):4–30, January 2024.

[123] Donald A McQuarrie. Stochastic approach to chemical kinetics. *Journal of Applied Probability*, 4(3):413–478, 1967.

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

[125] David Schnoerr, Guido Sanguinetti, and Ramon Grima. Approximation and inference methods for stochastic biochemical kinetics - a tutorial review. *Journal of Physics A: Mathematical and Theoretical*, 50(9):093001, 2017.

[126] Daniel T. Gillespie. Deterministic Limit of Stochastic Chemical Kinetics. *The Journal of Physical Chemistry B*, 113(6):1640–1644, February 2009.

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

[128] Tobias Jahnke and Wilhelm Huisinga. Solving the chemical master equation for monomolecular reaction systems analytically. *Journal of Mathematical Biology*, 54(1):1–26, 2007.

[129] Stephen Smith and Vahid Shahrezaei. General transient solution of the one-step master equation in one dimension. *Physical Review E*, 91(6):062119, 2015.

[130] Crispin Gardiner. *Stochastic methods*, volume 4. Springer Berlin, 2009.

[131] David F Anderson, Gheorghe Craciun, and Thomas G Kurtz. Product-form stationary distributions for deficiency zero chemical reaction networks. *Bulletin of Mathematical Biology*, 72(8):1947–1970, 2010.

[132] P Arányi and J Tóth. A full stochastic description of the Michaelis-Menten reaction for small systems. *Acta biochimica et biophysica; Academiae Scientiarum Hungaricae*, 12(4):375–388, 1977.

[133] Ian J Laurenzi. An analytical solution of the stochastic master equation for reversible bimolecular reaction kinetics. *The Journal of Chemical Physics*, 113(8):3315–3322, 2000.

[134] R. Grima, D. R. Schmidt, and T. J. Newman. Steady-state fluctuations of a genetic feedback loop: An exact solution. *The Journal of Chemical Physics*, 137(3), July 2012.

[135] Yiling Wang, Zhenhua Yu, Ramon Grima, and Zhixing Cao. Exact solution of a three-stage model of stochastic gene expression including cell-cycle dynamics. *bioRxiv*, page 2023.08.29.555255, September 2023.

[136] Bingjie Wu, Holehouse James, Ramon Grima, and Chen Jia. Solving the time-dependent protein distributions for autoregulated bursty gene expression using spectral decomposition. Part I: Conventional models. *bioRxiv*, page 2023.11.21.568174, November 2023.

[137] Brian Munsky and Mustafa Khammash. The finite state projection algorithm for the solution of the Chemical Master Equation. *The Journal of Chemical Physics*, 124(4):044104, 2006.

[138] Brian Munsky and Mustafa Khammash. A multiple time interval finite state projection algorithm for the solution to the chemical master equation. *Journal of Computational Physics*, 226(1):818–835, September 2007.

[139] Verena Wolf, Rushil Goel, Maria Mateescu, and Thomas A. Henzinger. Solving the chemical master equation using sliding windows. *BMC Syst. Biol.*, 4(1):1–19, December 2010.

[140] R. B. Sidje and H. D. Vo. Solving the chemical master equation by a fast adaptive finite state projection based on the stochastic simulation algorithm. *Mathematical Biosciences*, 269:10–16, November 2015.

[141] Khanh N. Dinh and Roger B. Sidje. Understanding the finite state projection and related methods for solving the chemical master equation. *Physical Biology*, 13(3):035003, May 2016.

[142] Shev MacNamara, Kevin Burrage, and Roger B. Sidje. Multiscale Modeling of Chemical Kinetics via the Master Equation. *Multiscale Modeling & Simulation*, January 2008.

[143] Huy Vo and Brian Munsky. A Parallel Implementation of the Finite State Projection Algorithm for the Solution of the Chemical Master Equation. *bioRxiv*, page 2020.06.30.180273, July 2020.

[144] Vladimir Kazeev, Mustafa Khammash, Michael Nip, and Christoph Schwab. Direct solution of the Chemical Master Equation using quantized tensor trains. *PLOS Comp. Bio.*, 10(3):e1003359, March 2014.

[145] Vladimir Kazeev and Christoph Schwab. Tensor approximation of stationary distributions of chemical reaction networks. *SIAM J. Matrix Anal. & Appl.*, 36(3):1221–1247, January 2015.

[146] Sergey Dolgov and Boris Khoromskij. Simultaneous state-time approximation of the chemical master equation using tensor product formats. *Numerical Linear Algebra with Applications*, 22(2):197–219, June 2014.

[147] Huy D. Vo and Roger B. Sidje. An adaptive solution to the chemical master equation using tensors. *Journal of Chemical Physics*, 147(4), July 2017.

[148] Ankit Gupta, Jan Mikelson, and Mustafa Khammash. A finite state projection algorithm for the stationary solution of the chemical master equation. *The Journal of Chemical Physics*, 147(15):154101, 2017.

[149] Trang Dinh and Roger B. Sidje. An adaptive solution to the Chemical Master Equation using quantized tensor trains with sliding windows. *Physical Biology*, 17(6):065014, July 2020.

[150] Ion Gabriel Ion, Christian Wildner, Dimitrios Loukrezis, Heinz Koeppl, and Herbert De Gersem. Tensor-train approximation of the chemical master equation and its application for parameter inference. *Journal of Chemical Physics*, 155(3), July 2021.

[151] Daniel T Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22(4):403–434, 1976.

[152] Daniel T Gillespie. Exact stochastic simulation of coupled chemical reactions. *The journal of physical chemistry*, 81(25):2340–2361, 1977.

[153] Daniel T. Gillespie, Andreas Hellander, and Linda R. Petzold. Perspective: Stochastic algorithms for chemical kinetics. *The Journal of Chemical Physics*, 138(17), May 2013.

[154] Michael A. Gibson and Jehoshua Bruck. Efficient Exact Stochastic Simulation of Chemical Systems with Many Species and Many Channels. *The Journal of Physical Chemistry A*, 104(9):1876–1889, March 2000.

[155] Yang Cao, Hong Li, and Linda Petzold. Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. *The Journal of Chemical Physics*, 121(9):4059–4067, September 2004.

[156] Larry Lok and Roger Brent. Automatic generation of cellular reaction networks with Moleculizer 1.0. *Nature Biotechnology*, 23:131–136, January 2005.

[157] James M. McCollum, Gregory D. Peterson, Chris D. Cox, Michael L. Simpson, and Nagiza F. Samatova. The sorting direct method for stochastic simulation of biochemical systems with varying reaction execution behavior. *Comput. Biol. Chem.*, 30(1):39–49, February 2006.

[158] David F. Anderson. A modified next reaction method for simulating chemical systems with time dependent propensities and delays. *The Journal of Chemical Physics*, 127(21), December 2007.

[159] Alexander Slepoy, Aidan P. Thompson, and Steven J. Plimpton. A constant-time kinetic Monte Carlo algorithm for simulation of large biochemical reaction networks. *The Journal of Chemical Physics*, 128(20), May 2008.

[160] David J. Warne, Ruth E. Baker, and Matthew J. Simpson. Simulation and inference algorithms for stochastic biochemical reaction networks: from basic concepts to state-of-the-art. *Journal of the Royal Society Interface Interface*, 16(151):20180943, February 2019.

[161] J. Goutsias and G. Jenkinson. Markovian dynamics on complex reaction networks. *Physics Reports*, 529(2):199–264, August 2013.

[162] Thomas G. Kurtz. Strong approximation theorems for density dependent Markov chains. *Stochastic Processes and their Applications*, 6(3):223–240, February 1978.

[163] Daniel T Gillespie. The chemical Langevin equation. *The Journal of Chemical Physics*, 113(1):297–306, 2000.

[164] Bence Mélykúti, Kevin Burrage, and Konstantinos C. Zygalakis. Fast stochastic simulation of biochemical reaction systems by alternative formulations of the chemical Langevin equation. *Journal of Chemical Physics*, 132(16), April 2010.

[165] Ramon Grima, Philipp Thomas, and Arthur V Straube. How accurate are the nonlinear chemical Fokker-Planck and chemical Langevin equations? *The Journal of Chemical Physics*, 135(8):084103, 2011.

[166] David Schnoerr, Guido Sanguinetti, and Ramon Grima. Validity conditions for moment closure approximations in stochastic chemical kinetics. *The Journal of Chemical Physics*, 141(8):084103, August 2014. Publisher: American Institute of Physics.

[167] Andrew Duncan, Shuohao Liao, Tomáš Vejchodský, Radek Erban, and Ramon Grima. Noise-induced multistability in chemical systems: Discrete versus continuum modeling. *Physical Review E*, 91(4):042111, April 2015.

[168] Lucy Ham, Megan A. Coomer, and Michael P. H. Stumpf. The chemical Langevin equation for biochemical systems in dynamic environments. *Journal of Chemical Physics*, 157(9), September 2022.

[169] Philipp Thomas and Ramon Grima. Approximate probability distributions of the master equation. *Physical Review E*, 92(1):012120, 2015.

[170] Johan Elf and Måns Ehrenberg. Fast evaluation of fluctuations in biochemical networks with the linear noise approximation. *Genome research*, 13(11):2475–2484, 2003.

[171] E. W. J. Wallace, D. T. Gillespie, K. R. Sanft, and L. R. Petzold. Linear noise approximation is valid over limited times for any chemical system that is sufficiently large. *IET Systems Biology*, 6(4):102–115, August 2012.

[172] Luca Cardelli, Marta Kwiatkowska, and Luca Laurenti. Stochastic analysis of Chemical Reaction Networks using Linear Noise Approximation. *Biosystems*, 149:26–33, November 2016.

[173] Ramon Grima. Linear-noise approximation and the chemical master equation agree up to second-order moments for a class of chemical systems. *Physical Review E*, 92(4):042124, 2015.

[174] Leo A. Goodman. Population growth of the sexes. *Biometrics*, 9(2):212–225, 1953.

[175] P. Whittle. On the use of the normal approximation in the treatment of stochastic processes. *Journal of the Royal Statistical Society: Series B (Methodological)*, 19(2):268–281, 1957.

[176] Ingemar Nåsell. An extension of the moment closure method. *Theor. Popul. Biol.*, 64(2):233–239, September 2003.

[177] Joao Hespanha. Moment closure for biochemical networks. In *2008 3rd International Symposium on Communications, Control and Signal Processing*, pages 142–147, 2008.

[178] Ramon Grima. A study of the accuracy of moment-closure approximations for stochastic chemical kinetics. *The Journal of Chemical Physics*, 136(15):04B616, 2012.

[179] Angelique Ale, Paul Kirk, and Michael P. H. Stumpf. A general moment expansion method for stochastic kinetic models. *The Journal of Chemical Physics*, 138(17):174101, May 2013.

[180] David Schnoerr, Guido Sanguinetti, and Ramon Grima. Comparison of different moment-closure approximations for stochastic chemical kinetics. *The Journal of Chemical Physics*, 143(18):185101, November 2015.

[181] Eszter Lakatos, Angelique Ale, Paul D. W. Kirk, and Michael P. H. Stumpf. Multivariate moment closure techniques for stochastic kinetic models. *The Journal of Chemical Physics*, 143(9):094107, September 2015.

[182] Zhixing Cao and Ramon Grima. Linear mapping approximation of gene regulatory networks with stochastic dynamics. *Nature Communications*, 9(1):3305, 2018.

[183] JAM Janssen. The elimination of fast variables in complex chemical reactions. II. Mesoscopic level (reducible case). *Journal of Statistical Physics*, 57(1):171–185, 1989.

[184] Yang Cao, Daniel T Gillespie, and Linda R Petzold. The slow-scale stochastic simulation algorithm. *The Journal of Chemical Physics*, 122(1):014116, 2005.

[185] John Goutsias. Quasiequilibrium approximation of fast reaction kinetics in stochastic biochemical systems. *The Journal of Chemical Physics*, 122(18):184102, 2005.

[186] Philipp Thomas, Arthur V Straube, and Ramon Grima. The slow-scale linear noise approximation: an accurate, reduced stochastic description of biochemical networks under timescale separation conditions. *BMC systems biology*, 6(1):39, 2012.

[187] Pavol Bokes, John R. King, Andrew T. A. Wood, and Matthew Loose. Multiscale stochastic modelling of gene expression. *Journal of Mathematical Biology*, 65(3):493–520, September 2012.

[188] Jae K Kim, Krešimir Josić, and Matthew R Bennett. The validity of quasi-steady-state approximations in discrete stochastic simulations. *Biophysical journal*, 107(3):783–793, 2014.

[189] Stefano Bo and Antonio Celani. Multiple-scale stochastic processes: decimation, averaging and beyond. *Physics reports*, 670:1–59, 2017.

[190] Daniel T Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of Chemical Physics*, 115(4):1716–1733, 2001.

[191] Yang Cao, Daniel T. Gillespie, and Linda R. Petzold. Efficient step size selection for the tau-leaping simulation method. *Journal of Chemical Physics*, 124(4), January 2006.

[192] Yang Cao, Daniel T. Gillespie, and Linda R. Petzold. Adaptive explicit-implicit tau-leaping method with automatic tau selection. *Journal of Chemical Physics*, 126(22), June 2007.

[193] Andreas Hellander and Per Lötstedt. Hybrid method for the chemical master equation. *Journal of Computational Physics*, 227(1):100–122, November 2007.

[194] Stephan Menz, Juan C. Latorre, Christof Schütte, and Wilhelm Huisinga. Hybrid Stochastic–Deterministic Solution of the Chemical Master Equation. *Multiscale Modeling & Simulation*, October 2012.

[195] Andrew Duncan, Radek Erban, and Konstantinos Zygalakis. Hybrid framework for the simulation of stochastic chemical kinetics. *Journal of Computational Physics*, 326:398–419, December 2016.

[196] Stefanie Winkelmann and Christof Schütte. Hybrid models for chemical reaction networks: Multiscale theory and application to gene regulatory systems. *Journal of Chemical Physics*, 147(11), September 2017.

[197] Fei Liu, Monika Heiner, and David Gilbert. Hybrid modelling of biological systems: current progress and future prospects. *Briefings in Bioinformatics*, 23(3):bbac081, May 2022.

[198] Alexander Andreychenko, Luca Bortolussi, Ramon Grima, Philipp Thomas, and Verena Wolf. *Distribution Approximations for the Chemical Master Equation: Comparison of the Method of Moments and the System Size Expansion*, pages 39–66. Springer International Publishing, Cham, 2017.

[199] George E Briggs and John BS Haldane. A note on the kinetics of enzyme action. *Biochemical Journal*, 19(2):338, 1925.

[200] S Schnell and C Mendoza. Closed form solution for time-dependent enzyme kinetics. *Journal of Theoretical Biology*, 187(2):207–212, 1997.

[201] John J Tyson. Biochemical oscillations. In *Computational Cell Biology*, volume 1, pages 230–260. Springer, 2002.

[202] Christian Kuehn. Moment Closure—A Brief Review. In *Control of Self-Organizing Nonlinear Systems*, pages 253–271. Springer, Cham, Switzerland, January 2016.

[203] Christopher Rackauckas and Qing Nie. DifferentialEquations.jl A performant and feature-rich ecosystem for solving differential equations in Julia. *J. Open Res. Softw.*, 5(1):15, May 2017.

[204] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017.

[205] Elisabeth Roesch, Joe G. Greener, Adam L. MacLean, Huda Nassar, Christopher Rackauckas, Timothy E. Holy, and Michael P. H. Stumpf. Julia for biologists. *Nature Methods*, 20:655–664, May 2023.

[206] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[207] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Francis Bach. *Deep Learning*. MIT Press, Cambridge, Massachusetts, January 2017.

[208] P. Baldi, P. Sadowski, and D. Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature Communications*, 5(1):4308, 2014.

[209] Christof Angermueller, Tanel Pärnamaa, Leopold Parts, and Oliver Stegle. Deep learning for computational biology. *Mol. Syst. Biol.*, 12(7):878, 2016.

[210] Seonwoo Min, Byunghan Lee, and Sungroh Yoon. Deep learning in bioinformatics. *Brief. Bioinform.*, 18(5):851–869, September 2017.

[211] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4):045002, December 2019.

[212] Pankaj Mehta, Marin Bukov, Ching-Hao Wang, Alexandre G.R. Day, Clint Richardson, Charles K. Fisher, and David J. Schwab. A high-bias, low-variance introduction to Machine Learning for physicists. *Physics Reports*, 810:1–124, May 2019.

[213] Joe G. Greener, Shaun M. Kandathil, Lewis Moffat, and David T. Jones. A guide to machine learning for biologists. *Nature Reviews Molecular Cell Biology*, 23:40–55, January 2022.

[214] Ankit Gupta, Christoph Schwab, and Mustafa Khammash. DeepCME: A deep learning framework for computing solution statistics of the Chemical Master Equation. *PLOS Comp. Bio.*, 17(12):e1009623, 2021.

[215] Luca Bortolussi and Luca Palmieri. Deep abstractions of chemical reaction networks. In *Computational Methods in Systems Biology*, volume 11095, pages 21–38. Springer International Publishing, Cham, 2018.

[216] Denis Repin and Tatjana Petrov. Automated deep abstractions for stochastic chemical reaction networks. *Inf. Comput.*, 281:104788, December 2021.

[217] Christopher N. Davis, T. Deirdre Hollingsworth, Quentin Caudron, and Michael A. Irvine. The use of mixture density networks in the emulation of complex epidemiological individual-based models. *PLOS Computational Biology*, 16(3):e1006869, March 2020. Publisher: Public Library of Science.

[218] Francesca Cairoli, Ginevra Carbone, and Luca Bortolussi. Abstraction of Markov population dynamics via Generative Adversarial Nets. In *International Conference on Computational Methods in Systems Biology*, volume 1, pages 19–35. Springer, 2021.

[219] Gennady Gorin, Maria Carilli, Tara Chari, and Lior Pachter. Spectral neural approximations for models of transcriptional dynamics. *bioRxiv*, 2022.

[220] Ioana M. Gherman, Zahraa S. Abdallah, Wei Pang, Thomas E. Gorochowski, Claire S. Grierson, and Lucia Marucci. Bridging the gap between mechanistic biological models and machine learning surrogates. *PLOS Computational Biology*, 19(4):e1010988, April 2023.

[221] Daniel T. Gillespie, Linda R. Petzold, and Effrosyni Seitaridou. Validity conditions for stochastic chemical kinetics in diffusion-limited systems. *The Journal of Chemical Physics*, 140(5), February 2014.

[222] Stephen Smith and Ramon Grima. Spatial stochastic intracellular kinetics: A review of modelling approaches. *Bulletin of mathematical biology*, 81(8):2960–3009, 2019.

[223] Athel Cornish-Bowden. *Fundamentals of enzyme kinetics*. John Wiley & Sons, 2013.

[224] Anthony F Bartholomay. A stochastic approach to statistical kinetics with application to enzyme kinetics. *Biochemistry*, 1(2):223–230, 1962.

[225] David Schnoerr, Guido Sanguinetti, and Ramon Grima. The complex chemical Langevin equation. *The Journal of Chemical Physics*, 141:024103, 2014.

[226] Eric L Haseltine and James B Rawlings. Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. *The Journal of Chemical Physics*, 117(15):6959–6969, 2002.

[227] Marc R Roussel and Rui Zhu. Reducing a chemical master equation by invariant manifold methods. *The Journal of Chemical Physics*, 121(18):8716–8730, 2004.

[228] Philipp Thomas, Arthur V Straube, and Ramon Grima. Communication: limitations of the stochastic quasi-steady-state approximation in open biochemical reaction networks. *The Journal of chemical physics*, 135(18):181103, 2011.

[229] Jae K Kim, Krešimir Josić, and Matthew R Bennett. The relationship between stochastic and deterministic quasi-steady state approximations. *BMC systems biology*, 9(1):87, 2015.

[230] Hong Qian and Lisa M Bishop. The chemical master equation approach to non-equilibrium steady-state of open biochemical systems: Linear single-molecule enzyme kinetics and nonlinear biochemical reaction networks. *International journal of molecular sciences*, 11(9):3472–3500, 2010.

[231] Xingye Kan, Chang Hyeong Lee, and Hans G Othmer. A multi-time-scale analysis of chemical reaction networks: II. stochastic systems. *Journal of Mathematical Biology*, 73(5):1081–1129, 2016.

[232] Narmada Herath and Domitilla Del Vecchio. Reduced linear noise approximation for biochemical reaction networks with time-scale separation: The stochastic tQSSA+. *The Journal of Chemical Physics*, 148(9):094108, 2018.

[233] Erel Levine and Terence Hwa. Stochastic fluctuations in metabolic pathways. *Proceedings of the National Academy of Sciences*, 104(22):9224–9229, 2007.

[234] Marianne O Stefanini, Alan J McKane, and Timothy J Newman. Single enzyme pathways and substrate fluctuations. *Nonlinearity*, 18(4):1575, 2005.

[235] Ramon Grima. Noise-induced breakdown of the Michaelis-Menten equation in steady-state conditions. *Physical Review Letters*, 102(21):218103, 2009.

[236] Divya Singh and Srabanti Chaudhury. Single-molecule kinetics of an enzyme in the presence of multiple substrates. *ChemBioChem*, 19(8):842–850, 2018.

[237] Lee A Segel and Marshall Slemrod. The quasi-steady-state assumption: a case study in perturbation. *SIAM review*, 31(3):446–477, 1989.

[238] Lee A Segel. On the validity of the steady state assumption of enzyme kinetics. *Bulletin of Mathematical Biology*, 50(6):579–593, 1988.

[239] Chen Jia and Ramon Grima. Dynamical phase diagram of an auto-regulating gene in fast switching conditions. *The Journal of Chemical Physics*, 152(17):174110, 2020.

[240] Chen Jia. Model simplification and loss of irreversibility. *Physical Review E*, 93(5):052149, 2016.

[241] Nicholas J Higham. *Functions of matrices: theory and computation*, volume 104. SIAM, 2008.

[242] Emrah Kılıç and Pantelimon Stanica. The inverse of banded matrices. *Journal of Computational and Applied Mathematics*, 237(1):126–135, 2013.

[243] James W Brown and RV Churchill. *Complex variables and applications*. McGraw-Hill Higher Education, 2009.

[244] G Broggi, Luigi A Lugiato, and A Colombo. Transient bimodality in optically bistable systems. *Physical Review A*, 32(5):2803, 1985.

[245] Marcin Mierzejewski, Jerzy Dajka, Jerzy Łuczka, Peter Talkner, and Peter Hänggi. Dynamical bimodality in equilibrium monostable systems. *Physical Review E*, 74(4):041102, 2006.

[246] Chan F Lam and David G Priest. Enzyme Kinetics: Systematic Generation of Valid King-Altman Patterns. *Biophysical journal*, 12(3):248–256, 1972.

[247] Paul A Sims. An "Aufbau" Approach To Understanding How the King–Altman Method of Deriving Rate Equations for Enzyme-Catalyzed Reactions Works. *Journal of chemical education*, 86(3):385, 2009.

[248] Ji Yu, Jie Xiao, Xiaojia Ren, Kaiqin Lao, and X Sunney Xie. Probing gene expression in live cells, one protein molecule at a time. *Science*, 311(5767):1600–1603, 2006.

[249] David M Suter, Nacho Molina, David Gatfield, Kim Schneider, Ueli Schibler, and Felix Naef. Mammalian genes are transcribed with widely different bursting kinetics. *Science*, 332(6028):472–474, 2011.

[250] Mohammad Soltani, Cesar Augusto Vargas-Garcia, and Abhyudai Singh. Conditional Moment Closure Schemes for Studying Stochastic Dynamics of Genetic Circuits. *IEEE Transactions on Biomedical Circuits and Systems*, 9(4):518–526, August 2015.

[251] J. Hespanha. Moment closure for biochemical networks. In *2008 3rd International Symposium on Communications, Control and Signal Processing*, pages 142–147, 2008.

[252] C.S. Gillespie. Moment-closure approximations for mass-action models. *IET Systems Biology*, 3:52–58(6), January 2009.

[253] P. Azunre, C. Gómez-Uribe, and G. Verghese. Mass fluctuation kinetics: analysis and computation of equilibria and local dynamics. *IET Systems Biology*, 5:325–335(10), November 2011.

[254] Sisi Fan, Quentin Geissmann, Eszter Lakatos, Saulius Lukauskas, Angelique Ale, Ann C. Babtie, Paul D. W. Kirk, and Michael P. H. Stumpf. MEANS: python package for Moment Expansion Approximation, iNference and Simulation. *Bioinformatics*, 32(18):2863–2865, 05 2016.

[255] Atefeh Kazeroonian, Fabian Fröhlich, Andreas Raue, Fabian J. Theis, and Jan Hasenauer. CERENA: ChEmical REaction Network AnalyzerA Toolbox for the Simulation and Analysis of Stochastic Chemical Kinetics. *PLOS ONE*, 11(1):e0146732, January 2016.

[256] D.C.C Bover. Moment equation methods for nonlinear stochastic systems. *Journal of Mathematical Analysis and Applications*, 65(2):306–320, 1978.

[257] Khem Raj Ghusinga, Mohammad Soltani, Andrew Lamperski, Sairaj V. Dhople, and Abhyudai Singh. Approximate moment dynamics for polynomial and trigonometric stochastic systems. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 1864–1869, 2017.

[258] Peter Milner, Colin S. Gillespie, and Darren J. Wilkinson. Moment closure approximations for stochastic kinetic models with rational rate laws. *Math. Biosci.*, 231(2):99–104, June 2011.

[259] Chang Hyeong Lee. A Moment Closure Method for Stochastic Chemical Reaction Networks with General Kinetics. *MATCH Communications in Mathematical and in Computer Chemistry*, 70:785–800, 2013.

[260] N. Balakrishnan, Norman L. Johnson, and Samuel Kotz. A note on relationships between moments, central moments and cumulants from multivariate distributions. *Statist. Probab. Lett.*, 39(1):49–54, July 1998.

[261] L. Isserlis. On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables. *Biometrika*, 12(1/2):134–139, 1918.

[262] Matt J. Keeling. Multiplicative Moments and Measures of Persistence in Ecology. *J. Theor. Biol.*, 205(2):269–281, July 2000.

[263] Crow Edwin L. and Shimizu Kunio, editors. *Lognormal Distributions: Thoery and Applications*. Marcel Dekker, New York, 1st edition, 1988.

[264] Norman Lloyd. Johnson, Samuel. Kotz, and N. Balakrishnan. *Discrete multivariate distributions*. Wiley, New York, 1997.

[265] A. M. Mathal and P. G. Moschopoulos. A form of multivariate gamma distribution. *Annals of the Institute of Statistical Mathematics*, 44(1):97–106, March 1992.

[266] Edward Furman. On a multivariate gamma distribution. *Statistics & Probability Letters*, 78(15):2353–2360, October 2008.

[267] Abhyudai Singh and Joao Pedro Hespanha. Lognormal Moment Closures for Biochemical Reactions. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 2063–2068, 2006.

[268] Abhyudai Singh and João P. Hespanha. Approximate Moment Dynamics for Chemically Reacting Systems. *IEEE Transactions on Automatic Control*, 56(2):414–418, 2011.

[269] Kaan Öcal, Guido Sanguinetti, and Ramon Grima. Model reduction for the Chemical Master Equation: An information-theoretic approach. *The Journal of Chemical Physics*, 158(11):114113, 03 2023.

[270] Torkel E. Loman, Yingbo Ma, Vasily Ilin, Shashi Gowda, Niklas Korsbo, Nikhil Yewale, Chris Rackauckas, and Samuel A. Isaacson. Catalyst: Fast and flexible modeling of reaction networks. *PLos Computational Biology*, 19(10):e1011530, October 2023.

[271] Yingbo Ma, Shashi Gowda, Ranjan Anantharaman, Chris Laughman, Viral Shah, and Chris Rackauckas. ModelingToolkit: A Composable Graph Transformation System For Equation-Based Modeling. *arXiv*, 2021.

[272] Shashi Gowda, Yingbo Ma, Alessandro Cheli, Maja Gwóźzdź, Viral B. Shah, Alan Edelman, and Christopher Rackauckas. High-performance symbolic-numerics via multiple dispatch. *ACM Commun. Comput. Algebra*, 55(3):9296, jan 2022.

[273] Tobias Pietzsch, Lorenzo Duso, and Christoph Zechner. Compartor: a toolbox for the automatic generation of moment equations for dynamic compartment populations. *Bioinformatics*, 02 2021. btab058.

[274] Romain Veltz. BifurcationKit.jl, July 2020.

[275] Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, Ali Ramadhan, and Alan Edelman. Universal differential equations for scientific Machine Learning. *arXiv*, 2021.

[276] René Lefever, Grégoire Nicolis, and Pierre Borckmans. The brusselator: it does oscillate all the same. *Journal of the Chemical Society, Faraday Transactions 1*, 84:1013–1023, 1988.

[277] Christoph Zechner, Jakob Ruess, Peter Krenn, Serge Pelet, Matthias Peter, John Lygeros, and Heinz Koeppl. Moment-based inference predicts bimodality in transient gene expression. *Proceedings of the National Academy of Sciences*, 109(21):8340–8345, May 2012.

[278] Peter Milner, Colin S. Gillespie, and Darren J. Wilkinson. Moment closure based parameter inference of stochastic kinetic models. *Statistics and Computing*, 23(2):287–295, March 2013.

[279] Fabian Fröhlich, Philipp Thomas, Atefeh Kazeroonian, Fabian J. Theis, Ramon Grima, and Jan Hasenauer. Inference for stochastic chemical kinetics using moment equations and system size expansion. *PLOS Computational Biology*, 12(7):1–28, 07 2016.

[280] Jakob Ruess and John Lygeros. Moment-Based Methods for Parameter Inference and Experiment Design for Stochastic Biochemical Reaction Networks. *ACM Trans. Model. Comput. Simul.*, 25(2):1–25, February 2015.

[281] Alexander Lück and Verena Wolf. Generalized method of moments for estimating parameters of stochastic reaction networks. *BMC Systems Biology*, 10(1):1–12, December 2016.

[282] J. Hasenauer, V. Wolf, A. Kazeroonian, and F. J. Theis. Method of conditional moments (MCM) for the Chemical Master Equation. *Journal of Mathematical Biology*, 69(3):687–735, September 2014.

[283] Atefeh Kazeroonian, Fabian J. Theis, and Jan Hasenauer. Modeling of stochastic biological processes with non-polynomial propensities using non-central conditional moment equation. *IFAC Proceedings Volumes*, 47(3):1729–1735, January 2014.

[284] Lorenzo Duso and Christoph Zechner. Selected-node stochastic simulation algorithm. *Journal of Chemical Physics*, 148(16), April 2018.

[285] Hanna Josephine Wiederanders, Anne-Lena Moor, and Christoph Zechner. Automated Generation of Conditional Moment Equations for Stochastic Reaction Networks. In *Computational Methods in Systems Biology*, pages 286–293. Springer, Cham, Switzerland, August 2022.

[286] Baruch Barzel and Ofer Biham. Binomial Moment Equations for Stochastic Reaction Systems. *Physical Review Letters*, 106(15):150602, April 2011.

[287] Jiajun Zhang, Qing Nie, and Tianshou Zhou. A moment-convergence method for stochastic analysis of biochemical reaction networks. *The Journal of Chemical Physics*, 144(19), May 2016.

[288] Khem Raj Ghusinga, Cesar A. Vargas-Garcia, Andrew Lamperski, and Abhyudai Singh. Exact lower and upper bounds on stationary moments in stochastic biochemical systems. *Physical Biology*, 14(4):04LT01, June 2017. Publisher: IOP Publishing.

[289] Alexander Andreychenko, Linar Mikeev, and Verena Wolf. Model Reconstruction for Moment-Based Stochastic Chemical Kinetics. *ACM Transactions on Modeling and Computer Simulation*, 25(2):1–19, April 2015.

[290] Rafail V. Abramov. The multidimensional maximum entropy moment problem: a review of numerical methods. *Communications in Mathematical Sciences*, 8(2):377–392, 2010.

[291] Patrick Smadbeck and Yiannis N. Kaznessis. A closure scheme for chemical master equations. *Proceedings of the National Academy of Sciences*, 110(35):14261–14265, August 2013. Publisher: National Academy of Sciences Section: Biological Sciences.

[292] M. Vlysidis, P. H. Constantino, and Y. N. Kaznessis. *ZI-Closure Scheme: A Method to Solve and Study Stochastic Reaction Networks*, pages 159–174. Springer International Publishing, Cham, 2017.

[293] Vincent Wagner, Benjamin Castellaz, Marco Oesting, and Nicole Radde. Quasi-Entropy Closure: a fast and reliable approach to close the moment equations of the Chemical Master Equation. *Bioinformatics*, 38(18):4352–4359, September 2022.

[294] Davin Lunz, J. Frédéric Bonnans, and Jakob Ruess. Revisiting moment-closure methods with heterogeneous multiscale population models. *Mathematical Biosciences*, 350:108866, 2022.

[295] Tommaso Bianucci and Christoph Zechner. A local polynomial moment approximation for compartmentalised biochemical systems. *arXiv*, June 2023.

[296] Narendra Maheshri and Erin K. O'Shea. Living with noisy genes: how cells function reliably with inherent variability in gene expression. *Annual Review of Biophysics and Biomolecular Structure*, 36:413–434, 2007.

[297] Harley H McAdams and Adam Arkin. Its a noisy business! Genetic regulation at the nanomolar scale. *Trend in Genetics*, 15(2):65–69, 1999.

[298] Jonathan M. Raser and Erin K. O'Shea. Noise in gene expression: Origins, consequences, and control. *Science*, 309(5743):2010–2013, September 2005.

[299] Shangying Wang, Kai Fan, Nan Luo, Yangxiaolu Cao, Feilun Wu, Carolyn Zhang, Katherine A. Heller, and Lingchong You. Massive computational acceleration by using neural networks to emulate mechanism-based biological models. *Nature Communications*, 10(1):4354, September 2019.

[300] Qingchao Jiang, Xiaoming Fu, Shifu Yan, Runlai Li, Wenli Du, Zhixing Cao, Feng Qian, and Ramon Grima. Neural network aided approximation and parameter inference of non-Markovian models of gene expression. *Nature Communications*, 12(1):2618, May 2021.

[301] Christopher M. Bishop. Mixture density networks, 1994. available at `https://publications.aston.ac.uk/id/eprint/373/1/NCRG_94_004.pdf` (last accessed 26 April 2022).

[302] Kaan Öcal, Michael U. Gutmann, Guido Sanguinetti, and Ramon Grima. Inference and uncertainty quantification of stochastic gene expression via synthetic models. *Journal of The Royal Society Interface*, 19(192):20220153, 2022.

[303] L.U. Hjorth. Regularisation of mixture density networks. In *9th International Conference on Artificial Neural Networks*, volume 1999, pages 521–526, Edinburgh, UK, 1999. IEE.

[304] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep Sparse Rectifier Neural Networks. In *14th International Conference on Artificial Intelligence and Statistics*, pages 315–323, June 2011.

[305] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, January 1991.

[306] Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale Machine Learning. *SIAM Rev.*, 60(2):223–311, January 2018.

[307] Mike Innes. Flux: Elegant machine learning with Julia. *Journal of Open Source Software*, 3(25):602, May 2018.

[308] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035, 2019.

[309] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv*, 2014.

[310] Nir Friedman, Long Cai, and X. Sunney Xie. Linking stochastic dynamics to population distribution: an analytical framework of gene expression. *Physical Review Letters*, 97(16):168302, 2006.

[311] Chen Jia and Ramon Grima. Small protein number effects in stochastic models of autoregulated bursty gene expression. *The Journal of Chemical Physics*, 152(8):084115, 2020.

[312] Long Cai, Nir Friedman, and X. Sunney Xie. Stochastic protein expression in individual cells at the single molecule level. *Nature*, 440(7082):358–362, March 2006.

[313] Zakary S. Singer, John Yong, Julia Tischler, Jamie A. Hackett, Alphan Altinok, M. Azim Surani, Long Cai, and Michael B. Elowitz. Dynamic heterogeneity and DNA methylation in embryonic stem cells. *Mol. Cell*, 55(2):319–331, July 2014.

[314] Nicholas E. Phillips, Alice Hugues, Jake Yeung, Eric Durandau, Damien Nicolas, and Felix Naef. The circadian oscillator analysed at the single-transcript level. *Mol. Syst. Biol.*, 17(3):e10135, March 2021.

[315] I. M Sobol. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86–112, January 1967.

[316] Mike Innes, Alan Edelman, Keno Fischer, Chris Rackauckas, Elliot Saba, Viral B Shah, and Will Tebbutt. A differentiable programming system to bridge machine learning and scientific computing. *arXiv*, 2019.

[317] Heng Xu, Samuel O. Skinner, Anna Marie Sokac, and Ido Golding. Stochastic kinetics of nascent RNA. *Physica Review Letters*, 117(12):128101, September 2016.

[318] Markus Hafner, Markus Landthaler, Lukas Burger, Mohsen Khorshid, Jean Hausser, Philipp Berninger, Andrea Rothballer, Manuel Ascano Jr, Anna-Carina Jungkamp, Mathias Munschauer, Alexander Ulrich, Greg S. Wardle, Scott Dewell, Mihaela Zavolan, and Thomas Tuschl. Transcriptome-wide identification of RNA-binding protein and microRNA target sites by PAR-CLIP. *Cell*, 141(1):129–141, 2010.

[319] Brian Ingalls. Sensitivity analysis: from model parameters to system behaviour. *Essays Biochem.*, 45:177–194, 09 2008.

[320] Dan Cao and Roy Parker. Computational modeling of eukaryotic mRNA turnover. *RNA*, 7(9):11921212, 2001.

[321] David M. Suter, Nacho Molina, David Gatfield, Kim Schneider, Ueli Schibler, and Felix Naef. Mammalian genes are transcribed with widely different bursting kinetics. *Science*, 332(6028):472–474, 2011.

[322] Liang Chen, Chunjuan Zhu, and Feng Jiao. A generalized moment-based method for estimating parameters of stochastic gene transcription. *Mathematical Biosciences*, 345:108780, March 2022.

[323] Jan-Matthis Lueckmann, Giacomo Bassetto, Theofanis Karaletsos, and Jakob H Macke. Likelihood-free inference with emulator networks. In *1st Symposium on Advances in Approximate Bayesian Inference*, page 16, 2018.

[324] Rémi Bardenet, Arnaud Doucet, and Chris Holmes. On Markov chain Monte Carlo methods for tall data. *J. Mach. Learn. Res.*, 18(47):1–43, 2017.

[325] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, March 2010.

[326] N.S. Keskar, J. Nocedal, P.T.P. Tang, D. Mudigere, and M. Smelyanskiy. On large-batch training for deep learning: Generalization gap and sharp minima. *5th International Conference on Learning Representations*, 2017.

[327] Lutz Prechelt. Early stopping – but when? In *Neural Networks: Tricks of the Trade*, pages 53–67. Springer, Berlin, Heidelberg, 2nd edition, 2012.

[328] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13:25, 2012.

[329] Manuel Barrio, Kevin Burrage, André Leier, and Tianhai Tian. Oscillatory Regulation of Hes1: Discrete Stochastic Delay Modelling and Simulation. *PLOS Computational Biology*, 2(9):1–14, 09 2006.

[330] Andre Leier and Tatiana T. Marquez-Lago. Delay chemical master equation: direct and closed-form solutions. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2179):20150049, 2015.

[331] Xiao-Jiang Feng, Sara Hooshangi, David Chen, Genyuan Li, Ron Weiss, and Herschel Rabitz. Optimizing genetic circuits by global sensitivity analysis. *Biophysical Journal*, 87(4):2195–2202, 2004.

[332] Daniel E Zak, Gregory E Gonye, James S Schwaber, and Francis J Doyle. Importance of input perturbations and stochastic gene expression in the reverse engineering of genetic regulatory networks: insights from an identifiability analysis of an in silico network. *Genome research*, 13(11):2396–2405, 2003.

[333] Jörg Stelling, Ernst Dieter Gilles, and Francis J Doyle. Robustness properties of circadian clock architectures. *Proceedings of the National Academy of Sciences*, 101(36):13210–13215, 2004.

[334] Zhixing Cao, Tatiana Filatova, Diego A Oyarzún, and Ramon Grima. A stochastic model of gene expression with polymerase recruitment and pause release. *Biophysical Journal*, 119(5):1002–1014, 2020.

[335] Rudiyanto Gunawan, Yang Cao, Linda Petzold, and Francis J Doyle III. Sensitivity analysis of discrete stochastic systems. *Biophysical journal*, 88(4):2530–2540, 2005.

[336] Laurent Valentin Jospin, Hamid Laga, Farid Boussaid, Wray Buntine, and Mohammed Bennamoun. Hands-On Bayesian Neural Networks—A Tutorial for Deep Learning Users. *IEEE Computational Intelligence Magazine*, 17(2):29–48, April 2022.

[337] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5301–5310. PMLR, 09–15 Jun 2019.

[338] Chen Jia and Ramon Grima. Frequency domain analysis of fluctuations of mRNA and protein copy numbers within a cell lineage: theory and experimental validation. *Physical Review X*, 11(2):021032, May 2021.

[339] Juraj Szavits-Nossan and Ramon Grima. Steady-state distributions of nascent RNA for general initiation mechanisms. *Physical Review Research*, 5(1):013064, January 2023.

[340] Srividya Iyer-Biswas and Anton Zilman. *First-Passage Processes in Cellular Biology*, pages 261–306. John Wiley & Sons, Ltd, 2016.

[341] M. Frankowicz and G. Nicolis. Transient evolution towards a unique stable state: Stochastic analysis of explosive behavior in a chemical system. *Journal of Statistical Physics*, 33(3):595–609, December 1983.

[342] G. Nicolis and F. Baras. Intrinsic randomness and spontaneous symmetry-breaking in explosive systems. *Journal of Statistical Physics*, 48(5):1071–1090, September 1987.

[343] Karmeshu and Debasree Goswami. Transient Bimodality and Catastrophic Jumps in Innovation Diffusion. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 38(3):644–654, April 2008.

[344] Lorenzo Duso and Christoph Zechner. Stochastic reaction networks in dynamic compartment populations. *Proceedings of the National Academy of Sciences*, 117(37):22674–22683, September 2020.

[345] Lars Ferm, Per Lötstedt, and Andreas Hellander. A Hierarchy of Approximations of the Master Equation Scaled by a Size Parameter. *Journal of Scientific Computing*, 34(2):127–151, February 2008.

[346] Philipp Thomas, Hannes Matuschek, and Ramon Grima. Intrinsic noise analyzer: a software package for the exploration of stochastic biochemical kinetics using the system size expansion. *PloS one*, 7(6):e38518, 2012.

[347] Douglas E. Weidemann, James Holehouse, Abhyudai Singh, Ramon Grima, and Silke Hauf. The minimal intrinsic stochasticity of constitutively expressed eukaryotic genes is sub-Poissonian. *Science Advances*, 9(32), August 2023.

[348] Gábor Lente. A binomial stochastic kinetic approach to the Michaelis–Menten mechanism. *Chem. Phys. Lett.*, 568-569:167–169, May 2013.

[349] Gennady Gorin, John J. Vastola, and Lior Pachter. Studying stochastic systems biology of the cell with single-cell genomics data. *Cell Systems*, 14(10):822–843.e22, October 2023.

[350] Maria Carilli, Gennady Gorin, Yongin Choi, Tara Chari, and Lior Pachter. Biophysical modeling with variational autoencoders for bimodal, single-cell RNA sequencing data. *bioRxiv*, page 2023.01.13.523995, May 2023.

[351] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, December 2020.

[352] Quentin Badolle, Gabrielle Berrada, and Mustafa Khammash. Efficient Fisher Information Computation and Policy Search in Sampled Stochastic Chemical Reaction Networks through Deep Learning. *bioRxiv*, page 2023.04.13.535874, April 2023.

[353] Shangying Wang, Sara Capponi, and Simone Bianco. Inferring Conditional Probability Distributions of Noisy Gene Expression from Limited Observations by Deep Learning. *GEN Biotechnology*, 1(6):504–513, December 2022.

[354] Francesca Cairoli, Fabio Anselmi, Alberto D'Onofrio, and Luca Bortolussi. Generative abstraction of Markov population processes. *Theoretical Computer Science*, 977:114169, October 2023.

[355] Ying Tang, Jiayu Weng, and Pan Zhang. Neural-network solutions to stochastic reaction networks. *Nature Machine Intelligence*, 5:376–385, April 2023.

[356] Chuanbo Liu and Jin Wang. Language models as master equation solvers, 2023.

[357] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

[358] Alan A. Kaptanoglu, Jared L. Callaham, Aleksandr Aravkin, Christopher J. Hansen, and Steven L. Brunton. Promoting global stability in data-driven models of quadratic nonlinear dynamics. *Physical Review Fluids*, 6(9):094401, September 2021.

[359] J. Ruess, A. Milias-Argeitis, S. Summers, and J. Lygeros. Moment estimation for chemically reacting systems by extended Kalman filtering. *The Journal of Chemical Physics*, 135(16), October 2011.

[360] Tamara Kurdyaeva and Andreas Milias-Argeitis. Uncertainty propagation for deterministic models of biochemical networks using moment equations and the extended Kalman filter. *Journal of the Royal Society Interface*, 18(181):20210331, August 2021.

[361] Oliver K. Ernst, Tom Bartol, Terrence Sejnowski, and Eric Mjolsness. Deep Learning Moment Closure Approximations using Dynamic Boltzmann Distributions. *ArXiv e-prints*, May 2019.

[362] Changhae Andrew Kim, Nathan D. Ricke, and Troy Van Voorhis. Machine learning dynamic correlation in chemical kinetics. *The Journal of Chemical Physics*, 155(14):144107, 10 2021.

[363] Hugh R. Wilson and Jack D. Cowan. Excitatory and Inhibitory Interactions in Localized Populations of Model Neurons. *Biophysical Journal*, 12(1):1–24, January 1972.

[364] Alain Destexhe and Terrence J. Sejnowski. The WilsonCowan model, 36 years later. *Biological Cybernetics*, 101(1):1–2, July 2009.

[365] Jack D. Cowan. *Stochastic Neurodynamics*, pages 62–69. Morgan Kaufmann, San Mateo, 1991.

[366] Paul C. Bressloff. Metastable states and quasicycles in a stochastic Wilson-Cowan model of neuronal population dynamics. *Physical Review E*, 82(5):051903, November 2010.

[367] Marc Benayoun, Jack D. Cowan, Wim van Drongelen, and Edward Wallace. Avalanches in a Stochastic Model of Spiking Neurons. *PLoS Computational Biology*, 6(7):e1000846, July 2010.

[368] Edward Wallace, Marc Benayoun, Wim van Drongelen, and Jack D. Cowan. Emergent Oscillations in Networks of Stochastic Spiking Neurons. *PLoS ONE*, 6(5):e14804, May 2011.

[369] Toru Ohira and Jack D. Cowan. *Stochastic Neurodynamics and the System Size Expansion*, pages 290–294. Springer US, Boston, MA, 1997.

[370] Paul C. Bressloff. Stochastic Neural Field Theory and the System-Size Expansion. *SIAM Journal on Applied Mathematics*, 70(5):1488–1521, January 2010.

[371] Grégory Dumont, Georg Northoff, and André Longtin. Linear noise approximation for oscillations in a stochastic inhibitory network with delay. *Physical Review E*, 90(1):012702, July 2014.

[372] Grégory Dumont, Georg Northoff, and André Longtin. A stochastic model of input effectiveness during irregular gamma rhythms. *Journal of Computational Neuroscience*, 40(1):85–101, February 2016.

[373] Priscilla E. Greenwood and Lawrence M. Ward. *Stochastic Neuron Models*. Springer, 2016.

[374] Arthur S. Powanwe and André Longtin. Determinants of Brain Rhythm Burst Statistics. *Scientific Reports*, 9(1):1–23, December 2019. Number: 1 Publisher: Nature Publishing Group.

[375] Toru Ohira and Jack D. Cowan. Master-equation approach to stochastic neurodynamics. *Physical Review E*, 48(3):2259–2266, September 1993.

[376] Michael A. Buice and Jack D. Cowan. Field-theoretic approach to fluctuation effects in neural networks. *Physical Review E*, 75(5):051919, May 2007.

[377] Michael A. Buice, Jack D. Cowan, and Carson C. Chow. Systematic Fluctuation Expansion for Neural Network Activity Equations. *Neural Computation*, 22(2):377–426, February 2010.

[378] Jonathan D. Touboul and G. Bard Ermentrout. Finite-size and correlation-induced effects in mean-field dynamics. *Journal of Computational Neuroscience*, 31(3):453–484, November 2011.

[379] Michael E. Rule, David Schnoerr, Matthias H. Hennig, and Guido Sanguinetti. Neural field models for latent state inference: Application to large-scale neuronal recordings. *PLOS Computational Biology*, 15(11):e1007442, November 2019. Publisher: Public Library of Science.