COMPUTATIONAL METHODS FOR MODELING MULTISTEP REACTIONS AND PARAMETER INFERENCE IN TRANSCRIPTIONAL PROCESSES

Thesis

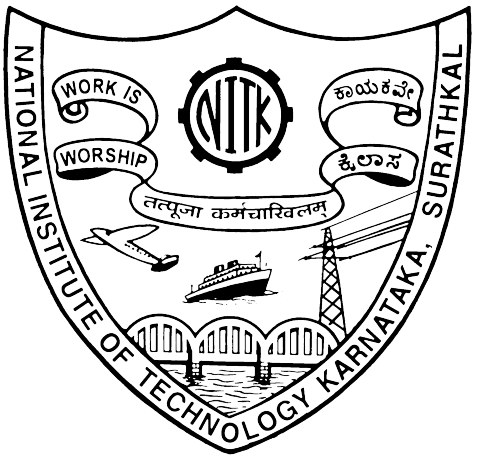
Submitted in partial fulfilment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

KEERTHI SRINIVAS SHETTY

(135024CS13F03)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA

SURATHKAL, MANGALORE-575025

JANUARY, 2020

DECLARATION

I hereby ***declare*** that the Research Thesis entitled **COMPUTATIONAL METHODS FOR MODELING MULTISTEP REACTIONS AND PARAMETER INFERENCE IN TRANSCRIPTIONAL PROCESSES** which is being submitted to the ***National Institute of Technology Karnataka, Surathkal*** in partial fulfillment of the requirements for the award of the Degree of ***Doctor of Philosophy*** is a ***bona fide report of the research work carried out by me***. The material contained in this thesis has not been submitted to any University or Institution for the award of any degree.

KEERTHI SRINIVAS SHETTY
Register No.: 135024CS13F03
Department of Computer Science and Engineering

Place: NITK, Surathkal
Date:

CERTIFICATE

This is to *certify* that the Research Thesis entitled **COMPUTATIONAL METHODS FOR MODELING MULTISTEP REACTIONS AND PARAMETER INFERENCE IN TRANSCRIPTIONAL PROCESSES**, submitted by **KEERTHI SRINIVAS SHETTY**(Register Number: 135024CS13F03) as the record of the research work carried out by her, is *accepted as the Research Thesis submission* in partial fulfillment of the requirements for the award of degree of ***Doctor of Philosophy***.

Dr. Annappa B
Research Supervisor
Professor
Department of Computer Science and Engineering
NITK Surathkal - 575025

Chairman - DRPC
(Signature with Date and Seal)

To my beloved

Amma & Papa

ACKNOWLEDGEMENT

Undertaking this PhD has been a truly life-changing experience for me and it would not have been possible to do without the support and guidance that I received from many people.

Firstly, I would like to express my sincere gratitude to my advisor *Prof.Annappa* for the continuous support of my Ph.D study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

Besides my advisor, I wholeheartedly convey my sincere appreciation to my mentor *Dr.Sriram Rajamani*, Distinguished Scientist and Managing Director, Microsoft Research India. His sheer joy and enthusiasm toward research motivated me throughout my PhD. Under his guidance, not only I got rigorous training in research paper writing, design and execution, but also enjoyed the freedom to pursue my dreams and I am always grateful to him for that. He has been an exceptional mentor and a very compassionate human being, which was apparent in his unconditional support during difficult times of my life. No word can justify my sincere gratitude and respect for him. Thank you so much sir.

I would like to thank the rest of my thesis committee: *Dr.Shashidhar G Koolagudi* and *Dr.Pushparaj Shetty D*, for their insightful comments and encouragement, but also for the hard question which incented me to widen my research from various perspectives. I am thankful to *Dr.Jeny Rajan* for his support and providing me access to Image processing lab. I am also thankful to *Dr.Basavaraj Talawar* for his support.

I am grateful to *Dr.Alwyn Roshan Pais*, Head of the Department, Deptartment of Computer Science and Engineering, NITK Surathkal for all his support and encouragement. I sincerely thank all teaching, technical and administrative staff of the Department of Computer Science and Engineering, NITK, for their help during my studies here. I heartily thank all the help extended by my friends during my studies at NITK. I feel

immensely proud to acknowledge the facilities and kindhearted support by *Prof. K. Uma Maheshwar Rao*, respected Director of NITK Surathkal.

Special mention to *Dr.Natarajan Shankar*, SRI International for enlightening me the first glance of research during my master's and providing me continuous support, motivation with his immense knowledge.

I express deep and sincere gratitute to my family : where the most basic source of my life energy resides. I have an amazing family, unique in many ways, and the stereotype of a perfect family in many others. I am grateful to my mother *Kala Shetty* and father *Srinivas Shetty* for their unconditional love and support all these years; I am grateful to my lovely brother *Preetham Shetty* for his unconditional support during my tough times. I am grateful to my uncle *Ratnakar Shetty* for his unconditional love and support in all my endeavour. I am grateful to my forever interested, encouraging and always enthusiastic grandmother *Rukmini(Rukku)* : she was always keen to know what I was doing and how I was proceeding, although it is likely that she has never grasped what it was all about! A special heartfelt appreciation to my loving husband *Chetan Srinidhi* for all the unconditional support, care and love which you have provided throughout my research career.

Finally, I thank my Lord, for making me feel his presence in my everyday life. When I look back upon this difficult period of my life, I can see only one set of footprints; now realizing that he carried me through this part of my journey. Love you Lord for the countless blessing you have bestowed on me.

Thanks for all your encouragement!

ABSTRACT

A major task in Systems Biology is to conduct accurate mechanistic simulations of multistep reactions. The simulation of a biological process from experimental data requires detailed knowledge of its model structure and kinetic parameters. Despite advances in experimental techniques, estimating unknown parameter values from observed data remains a bottleneck for obtaining accurate simulation results. Therefore, the goal is to focus on development of computationally efficient parameter inference methods for characterizing transcriptional bursting process, for inferring unknown kinetic parameters, given single-cell time-series data.

Many biochemical events involve multistep reactions. One of the most important biological processes in gene expression, which involve multistep reactions, is the transcriptional process. Models for multistep reactions necessarily need multiple states, and it is a challenge to compute model parameters that best agree with experimental data. To address this issue, first, a novel model reduction strategy is devised, representing several number of promoter OFF states by a single state, accompanied by specifying a time delay for burst frequency. This model approximates complex promoter switching behavior with Erlang-distributed ON/OFF times. To explore combined effects of parameter inference and simulation, using this model reduction, two inference methods are developed namely, Delay-Bursty MCEM and Clumped-MCEM. These methods are applied to time-series data of endogenous mouse glutaminase promoter to validate model assumptions and infer the values of kinetic parameters. Simulation results are summarized below:

1. Models with multiple OFF states produce behaviour that is most consistent with experimental data and the bursting kinetics are promoter specific.

2. Delay-Bursty MCEM and Clumped-MCEM inference are more efficient for time-series data. The comparison with the state-of-the-art Bursty

$MCEM^2$ method shows that Delay-Bursty MCEM and Clumped-MCEM produce similar numerical accuracy. However, these methods are better in terms of efficiency. Delay-Bursty MCEM reduces computational cost by 37.44% as compared to Bursty $MCEM^2$. Clumped-MCEM reduces computational cost by 57.58% when compared with Bursty $MCEM^2$ and 32.19% when compared with Delay-Bursty MCEM.

Keywords : Model reduction; Parameter Inference; Mass action kinetics; Multistep promoter model; Single-cell time-series data.

Contents

List of Tables

List of Figures

List of Algorithms

ABBREVIATIONS

SSA	Stochastic Simulation Algorithm
MCEM	Monte Carlo Expectation Maximization
ABC	Approximate Bayesian Computation
SGD	Stochastic Gradient Descent
DSSA	Delay Stochastic Simulation Algorithm
MCEM	Modified Cai's Exact SSA Method
ODE	Ordinary Differential Equation
CME	Chemical Master Equation
PDE	Partial Differential Equation
DM	Direct Method
FSP	Finite State Projection
MCMC	Markov Chain Monte Carlo
EM	Expectation Maximization
RJMCMC	Reversible Jump Markov Chain Monte Carlo
LNA	Linear Noise Approximation
SDE	Stochastic Differential Equation
MH	Metropolis-Hastings
AIC	Akaike Information Criterion
MLEs	Maximum Likelihood Estimates
CE	Cross Entropy

NOTATIONS

k_{on}	promoter switching rate from OFF to ON
k_{off}	promoter switching rate from ON to OFF
k_m	mRNA production rate (in model 4.1)
γ_m	mRNA degradation rate
B_m	burst size
k_m	burst frequency (in model 4.2)
τ	time delay
S_i	N molecular species
$\mathrm{X}_i(t)$	number of molecular species S_i at time t
θ_j	kinetic rate constant
$\mathrm{h}_j(\mathrm{x}(t))$	a function that quantifies the number of possible ways reaction can occur
y	observed data
K	total number of simulated trajectories
$k^{'}$	indexes $K^{'}$ simulated trajectories that are consistent with the observed data
$K^{'}$	simulated trajectories that are consistent with the observed data
$\hat{\theta}_j^{(0)}$	initial guess for parameter θ_j
$\hat{\theta}_j^{(1)}$	first update for parameter θ_j
i	indexes the start of the simulation
${r_k}'$	the total number of reactions firing
$r_{jk'}$	the number of times the j^{th} reaction fires
a_{jk}^i	the value of the propensity function for the j^{th} reaction, immediately after the i^{th} event
$\tau_{ik'}$	the time interval between the events

Chapter 1

Introduction

This chapter provides a context for this thesis work in the interdisciplinary field of research named Systems Biology. The context is used to provide motivations for this thesis and briefly describes the contribution to the field. The structure of this thesis is also outlined here.

1.1 Motivations

1.1.1 Modeling chemical kinetics

To predict biological behaviours, Systems Biology seeks to combine experiments with computation, aiming to understand how biological processes produce specific behaviours at the system level, ultimately, developing new biological processes for useful purposes. Systems Biology takes into account the structure and dynamic interactions within the biological processes and aims to use this understanding for important purposes e.g., effective prevention and/or treatment of diseases.

Computational modeling and simulation plays a two-fold development role in Systems Biology. First, biological process are abstracted to form a model. The model encodes the temporal evolution of its state in a formal form. Second, it allows to visualize and to predict the causal effect of the biological system in time, through a computer simulation.

Essentially, the model is an effort to explicitly encode the knowledge of

the biological process in a precise form. The features of the model must include sufficient information for analyzing the system dynamics. For example, in molecular modelling, the model must be able to manage all the detailed information (velocity and/or position) of all molecular species. In contrast, a whole-cell model, must include only a description of all the key cellular processes. Therefore, to some extent, the biological model is an abstraction of the real system; however, it is useful to formalize the understanding of the biological process. In addition, modelling also provides an effective way to highlight gaps in the knowledge of biological processes.

The temporal behaviour of a given biological model is then realized by conducting simulation (*in silico* experiments). These simulation results are compared with real experimental data obtained from wet lab experiments. The inconsistency shows a lack of knowledge in the model of the biological process under study. Models which are validated can be used to discover indirect and hidden implications in the biological process, which sometimes are hard to perform in a wet lab. For example, *in silico* experiments, one can isolate some vital genes and observe in detail their individual and group behaviour. This is impossible in a wet lab condition since the cell may not survive or may not exist. The results produced *in silico* experiments are used for forming hypothesis, and suggesting new experiments, making its predictive feature extremely useful for quantitative analysis of biological processes.

To sum up, biological modeling and simulation in the post-genomic era are becoming increasingly important. The knowledge of biological process is integrated into a model, and testable predictions are made through simulation. Therefore,*in silico* experiments are highly preferred in terms of speed, ease and cost; however, it is also important to emphasize that *in silico* experiments are not an alternative to real biological experiments (such as wet lab experiments). Instead, *in silico* experiments are better used as complementary to wet lab experiments to advance biological research. The methodology of computational biology and experimental biology is depicted in 1.1. The method of integrating stochastic models and single-cell experiments to infer valuable information about gene expression model is

depicted in 1.2.

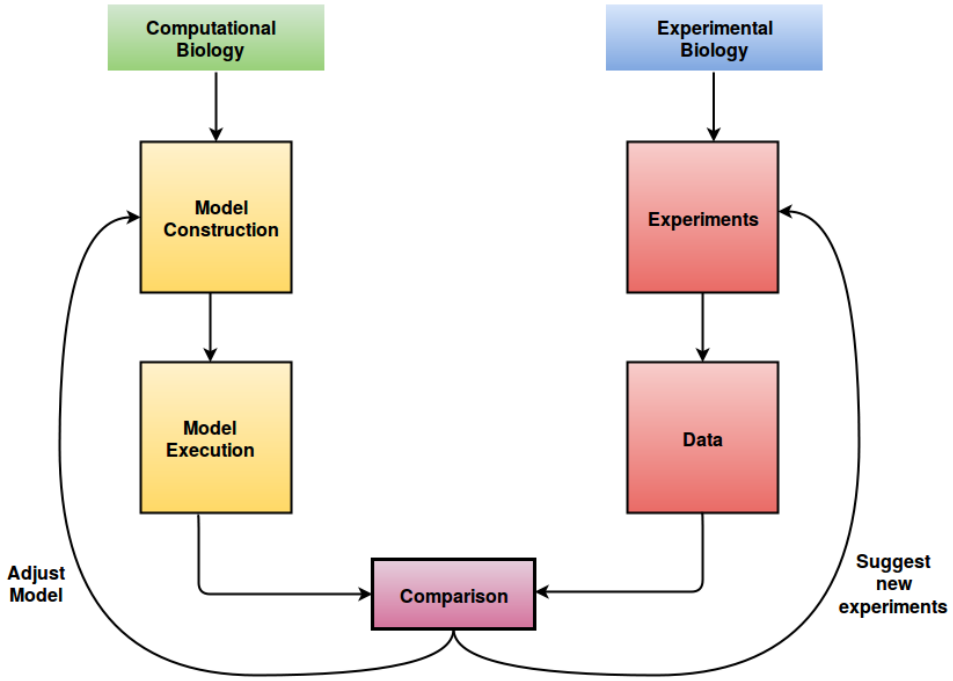


Figure 1.1: The methodology of computational biology and experimental biology.

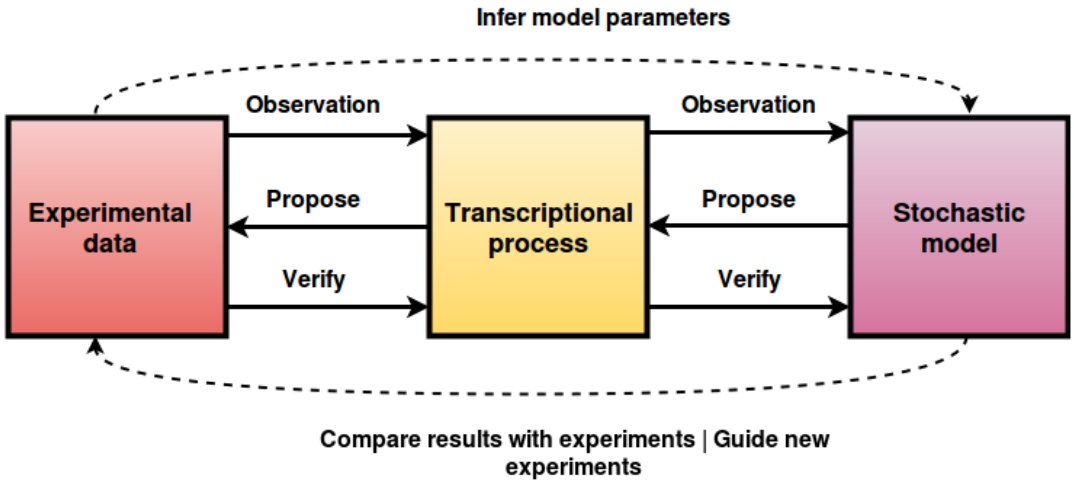


Figure 1.2: Integrating stochastic models and single-cell experiments to infer valuable information about gene expression model.

1.1.2 Noise in gene expression

Gene expression is the process, in which the information in a DNA sequence is converted into mRNAs and proteins, playing an essential role in the execution of all cellular functions. The widely studied aspect of gene expression is the stochasticity or noise associated with the process. This biological noise can be categorized into : Intrinsic noise and Extrinsic noise. Intrinsic noise (Barrio et al., 2010) arises in the system due to small number of key molecules, and also due to the uncertainty of knowing when a reaction occurs and which reaction it might be. Extrinsic noise (Barrio et al., 2010) is entirely different as the state changes are due to fluctuations in external conditions, such as temperature. Stochastic Simulation Algorithm (SSA) (Gillespie, 1977) is able to capture intrinsic noise of biochemical reactions because it takes into account discrete nature of species where the reactions between species are considered as stochastic events.

1.1.3 The use of delays

The desire for more realistic and, consequently, more accurate models is driven by the use of delays(Burrage et al., 2017). It is essential to introduce delays in order to conform models with observations and experimental data. In this case, complex processes are lumped while underlying mechanisms and inherent intermediate steps are not explicitly accounted for. Yet, the time that such processes require is included in the form of a constant delay or delay distribution. In this case, the delay distribution is used as a model-reduction technique, making the model smaller, and the analysis potentially feasible.

1.2 The need and challenges for parameter inference method

The construction of a suitable model depends on several factors, such as molecule numbers, distributions, the type of reactions and their time scales,

along with the noticeable effects of discreteness and intrinsic noise. To generate new hypothesis, successfully implemented models must be consistent with the data, reflect essential system properties, and also help answer specific questions about the system. Hence, the choice of a suitable parameter inference method depends on the models.

An ideal inference method must (1) leverage the intrinsic noise of the system, to better identify underlying mechanisms (Munsky et al., 2009), (2) accommodate unobserved/incomplete data, (3) infer the unknown kinetic parameters, and (4) provide computationally efficient performance for different multistep models. Currently, existing methods satisfy only a subset of these requirements. Suter et al. (2011) has performed Hidden Markov Model parameter inference for two and three state promoter models. These models assume noise-free promoter activity and RNA levels between discretely observed time points, but they do not provide an efficient means to characterize models with larger numbers of states . Daigle et al. (2012) has developed Monte Carlo Expectation Maximization (MCEM) with Modified Cross Entropy Method ($MCEM^2$), to infer kinetic parameters using stochastic simulation. The Bursty $MCEM^2$ (Daigle et al., 2015) modifies the $MCEM^2$ to accommodate the multistep model of transcriptional bursting. Toni et al. (2009) has developed an Approximate Bayesian Computation (ABC) based method, for inferring parameters and model structure, using stochastic simulations. Unfortunately, when using this method to discriminate between promoter models with increasing numbers of states, the addition of each state increases the number of unknown kinetic parameters (e.g. switching rates). More complex models become non-identifiable, in the presence of limited amounts of experimental data. Note that this drawback applies to any parameter inference method that explicitly represents transitions between individual promoter states. Finally, stochastic simulation of multistep promoter models suffers from a linear increase in computational cost; with the addition of each promoter state making the study of more complex models difficult. In the light of these observations, the goal is to focus on development of computationally efficient parameter inference

method for characterizing transcriptional bursting model by inferring the unknown kinetic parameters, given single-cell time-series data.

1.3 Problem statement

Development of a multistep promoter model to analyse transcriptional bursting process represented by biochemical reactions in terms of intrinsic noise and exact sampling of switching times between individual elements present in the system.

Objectives

- Development of model reduction strategy for the multistep transcriptional bursting process.
- Development of computationally efficient parameter inference method for characterizing transcriptional bursting model, given experimentally observed time-series data.
  - Analysis of intrinsic noise in the model caused by stochastic nature of biochemical reactions and molecules.
  - Calculation of exact sampling of switching times between individual elements.
  - To infer unknown parameters from the proposed model.

1.4 Contributions

This thesis aims to address two major issues in Systems Biology:

1. Model reduction for multistep reactions.
2. Development of parameter inference method for discrete stochastic systems.

Following are the specific contributions made:

- The study focuses on formulating multistep promoter models which accurately characterize transcriptional bursting. To this end, a novel model reduction strategy is devised, representing several number of promoter OFF states by a single state, accompanied by specifying a delay for burst frequency. This model approximates complex promoter switching behavior with Erlang-distributed ON/OFF times.

- The simulation part of the parameter inference method is performed by modifying two existing simulation algorithms, namely, Delay Stochastic Simulation Algorithm (DSSA) and Modified Cai's Exact SSA Method (MCEM). Both these algorithms are based on the idea of delays, providing accurate representation of proposed multistep model.

- Two parameter inference methods are developed by using this model reduction. Both strategies enable simulation and parameter inference.

  - **Delay-Bursty MCEM:** This approach combines Monte Carlo extension of Expectation Maximization (MCEM) and Delay Stochastic Simulation Algorithm (DSSA) to infer unknown parameters of discrete stochastic systems, given incomplete data.
  - **Clumped-MCEM:** This approach combines Monte Carlo extension of Expectation Maximization (MCEM) and Modified Cai's Exact SSA Method (MCEM) to infer unknown parameters of discrete stochastic systems, given incomplete data.

  These methods are applied to time-series data of endogenous mouse glutaminase promoter to validate the model assumptions and infer the kinetic parameters.

- Delay-Bursty MCEM and Clumped-MCEM reduce the computationally intensive task of modeling every single detail of multistep reactions. A delayed reaction is used to mimic the effects of these processes on the overall system dynamics.

- The empirical results support two main claims of this research:(1) Models

with multiple OFF states produce behaviour which is most consistent with experimental data and (2) Delay-Bursty MCEM, and Clumped-MCEM inference is more efficient for time-series data. The comparison of these methods with the state-of-the-art Bursty $MCEM^2$ method reveals that the same accuracy can be produced in less time.

1.5 Structure of the thesis

The thesis is outlined as follows.

- In Chapter 2, some well-known distributions from probability theory are recalled. The stochastic models and an algorithm for simulating their time-evolution is introduced in a detailed form, which gives basic notions and related information to understand results outlined in this thesis.

- In Chapter 3, various parameter inference approaches, based on intrinsic noise in gene regulatory models are briefly reviewed. It also discusses the model selection briefly.

- In Chapter 4, the widely used gene expression model, random telegraph model and its multistep model formulation is introduced. The multistep promoter model formulation is presented for two different inference approaches that have been developed in this work, namely; Delay-Bursty MCEM and Clumped-MCEM.

- In Chapter 5, two parameter inference methods developed are presented, namely, Delay-Bursty MCEM and Clumped-MCEM. Application of these algorithms to time-series data of endogenous mouse glutaminase promoter, validates model assumptions and infer the values of kinetic parameters. These methods show that Delay-Bursty MCEM and Clumped-MCEM produce the same numerical accuracy as Bursty $MCEM^2$ in less time.

- Finally, conclusions and possibilities for further research are presented in Chapter 6.

Chapter 2

Background Theory

Systems Biology is an interdisciplinary field of research involving mathematics, biology and computer science at different level of detail; a non trivial task, giving a comprehensive background of the required notions. In this chapter, the focus is on the related background, necessary to understand the results, outlined during the research. Moreover, the biological knowledge is recalled only when presenting models in Chapter 4.

This chapter introduces some well known distributions from probability theory. The Stochastic models and an algorithm for simulating their time-evolution is introduced in a detailed form.

2.1 Notions of probability theory

The models and representations that are considered in this thesis, provide a framework for thinking about the state of a biological processes; the reactions that can take place and the change in state that occurs as a result of particular chemical reactions. The state of biological processes evolves continuously through time, with discrete changes in state, occuring as the result of reaction events. These reaction events are stochastic and are governed by concepts from probability theory. Therefore, it is necessary to understand few concepts from probability theory in order to precisely comprehend these processes.

In order to represent biological processes; probability measure is introduced

with $\mathbb{P}$, discrete random variables with $\mathbb{N}$ as sample space and continuous random variables with $\mathbb{R}$ as sample space.

2.1.1 Exponential distribution

The most important continuous distribution in the theory of discrete-event stochastic simulation is exponential distribution (Evans et al., 2000)(Wilkinson, 2006). If X has exponential distribution with parameter $\lambda > 0$ it is written as,

$$X \sim Exp(\lambda)$$

and X has a probability density function

$$f(\mathrm{x}) = \begin{cases} \lambda e^{-\lambda \mathrm{x}}, \mathrm{x} \geq 0 \\ 0, otherwise \end{cases}$$

The cumulative distribution function is given by

$$F(\mathrm{x}) = \begin{cases} 0, \mathrm{x} < 0 \\ 1 - e^{-\lambda \mathrm{x}}, \mathrm{x} \geq 0 \end{cases}$$

A very important property characterizes this distribution: the memoryless property. If $X \sim Exp(\lambda)$ then for any positive $s, t \in \mathbb{R}$

$$\mathbb{P}(X > s + t | X > t) = \mathbb{P}(X > s) \tag{2.1}$$

which holds since

$$\mathbb{P}(X > s + t | X > t) = e^{-\lambda s}$$

Finally, let $\{X_i \sim Exp(\lambda_i) | i = 1, \ldots, n\}$ where all the variables are independent, thus defines the new variable $Y = min\{X_1, \ldots, X_n\}$. For such a variable,

$$\mathbb{P}(Y > \mathrm{x}) = \mathbb{P}(X_1 > \mathrm{x}; \ldots; X_n > \mathrm{x}) = \prod_{i=1}^{n} \mathbb{P}(X_i > \mathrm{x}) = e^{-\mathrm{x} \sum_{i=1}^{n} \lambda_i}$$

which means,

$$Y \sim Exp(\lambda_1 + \ldots + \lambda_n)$$

Sampling from the exponential distribution

The stochastic simulation is heavily based on how to generate exponentially distributed numbers. The sampling of a value, for a continuous random variable, can be obtained by an Inverse Monte-Carlo Algorithm based on the following considerations: given a continuous random variable X, with cumulative distribution F and given $p \sim U[0,1]$; holds that

$$\mathrm{x} = F^{-1}(p)$$

The computation of F^{-1}, though difficult, can be evaluated by the inverse of F, for the exponential distribution. Assuming $X \sim Exp(\lambda)$, and noting the definition of the exponential distribution,

$$\mathbb{P}(X \leq \mathrm{x}) = \int_0^x \lambda e^{-\lambda u} du$$

is the probability of X being smaller than x. Such a value is a probability, so is a number in $[0,1]$; also, assuming that a number can be picked $r \sim U[0,1]$. It can be written as,

$$\int_0^{\mathrm{x}} \lambda e^{-\lambda u} du = r$$

which integrates as

$$e^{-\lambda \mathrm{x}} = 1 - r$$

and, as known from the property of the uniform distribution; if $r \sim U[0,1]$ then $(1-r) \sim U[0,1]$. Now, computing the value for x is fairly easy, as applying the logarithm results in

$$\mathrm{x} = \lambda^{-1} ln r^{-1}$$

Once a value for r is picked, this equation permits to generate a sample for X.

2.1.2 Erlang distribution

The exponential distribution is a special case of a more general continuous distribution, the Erlang distribution (Evans et al., 2000). A random variable X following Erlang distribution is denoted as

$$X \sim \Gamma(n, \lambda)$$

where $n \in \mathbb{N}, n > 0$ is called the shape, and $\lambda > 0$ is the rate. When $n \in \mathbb{R}$ the distribution is called the Gamma distribution. Erlang distribution has probability density function defined as

$$f(\mathrm{x}) = \begin{cases} \frac{\lambda^n \mathrm{x}^{n-1} e^{-\lambda \mathrm{x}}}{(n-1)!}, \mathrm{x} \geq 0 \\ 0, \mathrm{x} < 0 \end{cases}$$

and cumulative distribution function defined as

$$F(\mathrm{x}) = \begin{cases} 0, \mathrm{x} < 0 \\ 1 - e^{-\lambda \mathrm{x}} \sum_{i=0}^{n-1} \frac{(\lambda \mathrm{x})^i}{i!}, \mathrm{x} \geq 0 \end{cases}$$

Three important properties can be stated for this distribution:

- Firstly, when the shape is 1, it reduces to the exponential distribution

  $$X \sim \Gamma(1, \lambda) \Rightarrow X \sim Exp(\lambda))$$

  and can be easily verified by the analytical form of the density function.

- Secondly, the summation of independent exponentially distributed random variables follows an Erlang distribution, namely

  $$X_1 \sim Exp(\lambda) \wedge X_2 \sim Exp(\lambda) \Rightarrow (X_1 + X_2) \sim \Gamma(2, \lambda)$$

  and also

  $$X_1 \sim \Gamma(n_1, \lambda) \wedge X_2 \sim \Gamma(n_2, \lambda) \Rightarrow (X_1 + X_2) \sim \Gamma(n_1 + n_2, \lambda),$$

  if X_1 and X_2 are independent.

- Finally, it is easy to notice that this distribution has infinite support in the same sense as the exponential one.

2.2 Stochastic models

This section introduces the importance of stochastic modelling, both for simulation and inference, using simple example which appears in Wilkinson (2006).

The example considered here is known as the linear birth-death process. Initially it is perhaps helpful to view this as a model for the number of bacteria in a bacterial colony. It is assumed that each bacterium in the colony gives rise to new individuals at the rate λ and each bacterium dies at the rate μ. Let the number of bacteria in the colony at time t be denoted $X(t)$. Assume that the number of bacteria in the colony at time zero is known to be x_0. Viewed in a continuous deterministic manner, this description of the system leads directly to the Ordinary Differential Equation (ODE)

$$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = \lambda X(t) - \mu X(t) = (\lambda - \mu)X(t)$$

The analytical solution for this ODE is

$$X(t) = \mathrm{x}_0 exp((\lambda - \mu)t)$$

The qualitative behavior of the bacterial colony for the deterministic ones is summarized in Table 2.1.

Table 2.1: Qualitative behavior of the bacterial colony in deterministic model.

Condition	$lim_{t\to\infty}X(t)$	Population size
$\lambda > \mu$	$+\infty$	size increases exponentially
$\lambda = \mu$	$X(t_0)$	constant size
$\lambda < \mu$	0	size decreases exponentially

In particular, the solution clearly depends only on $\lambda - \mu$ and not on the particular values that λ and μ take. In some sense, therefore, $\lambda - \mu$ is a sufficient description of the system dynamics. But it points out to a flip-side: namely, study of the experimental data on bacteria numbers can only provide information about $\lambda - \mu$, and not on the particular values of λ and μ

separately. Of course, this is not a problem if the continuous deterministic model is really appropriate, as then $\lambda - \mu$ is the only thing one needs to know and the precise values of λ and μ are not important for predicting system behaviour. Note, however, that the lack of identifiability of λ and μ has implications for model inference, as well as inference for rate constants. From the experimental data, it is clear that in this model a pure birth or death process, or a process involving both births and deaths are not known, as it is not possible to know if λ or μ is zero. To perform inferences and predictions about such biological processes stochastic models is needed. The important feature of the stochastic model is that it depends explicitly on both λ and μ, and not just on $\lambda - \mu$. This has important implications for the use of stochastic models for inference from experimental data. Using stochastic models representation for biological process, simulation and model inference can be conducted accurately considering stochastic dynamics having no deterministic counterparts (Caravagna and Hillston J, 2010; Kouyous et al., 2006).

2.3 Chemical Master Equation

The Chemical Master Equation (CME) is a widely used formalism describing stochastic reaction systems in well-mixed scenarios. The CME is a system of Ordinary Differential Equations (ODEs) describing all the state transitions by biochemical reactions. This section introduces the definition of the CME as in Gillespie (1976); Gillespie (1977); Marchetti et al. (2017). Suppose the biochemical reaction system starts with an initial state $X(t_0) = \mathrm{x}_0$ at time t_0. The purpose of the stochastic chemical kinetics is to infer the probability $\mathbb{P}(\mathrm{x}, t|\mathrm{x}_0, t_0)$. The probability function $\mathbb{P}(\mathrm{x}, t|\mathrm{x}_0, t_0)$ is

$$\mathbb{P}(\mathrm{x}, t|\mathrm{x}_0, t_0) = \text{ probability that the system state is } X(t) = \mathrm{x} \text{ at time } t, \text{ given the initial state } X(t_0) = \mathrm{x}_0 \text{ at time } t_0.$$

The probability $\mathbb{P}(\mathrm{x}, t|\mathrm{x}_0, t_0)$ is called the grand probability function as it gives the probabilities of all reachable states of the system at time t, given the initial state $X(t_0) = \mathrm{x}_0$ at time t_0. Knowing $\mathbb{P}(\mathrm{x}, t|\mathrm{x}_0, t_0)$, all the statistical properties (e.g., mean, variance) can be calculated for every species at any time $t > t_0$.

To derive the time evolution for the grand probability, consider an infinitesimal time interval $[t, t + dt)$ so that there is at most one reaction firing in this interval. Suppose at time $t + dt$ the system state is $X(t + dt) = \mathrm{x}$. There are two cases in order to reach the state x in the next infinitesimal time $t + dt$, given the current time t.

1. be at state $X(t) = \mathrm{x} - v_j$ at time t and reaction R_j fires in the next time $t + dt$ which leads to the next state $X(t + dt) = \mathrm{x}$.

2. already be at state $X(t) = \mathrm{x}$ at time t and no reaction fires in the next infinitesimal time interval $[t, t + dt)$.

The grand probability $\mathbb{P}(\mathrm{x}, t + dt|\mathrm{x}_0, t_0)$ is thus written as

$$\begin{aligned}\mathbb{P}(\mathrm{x}, t + dt|\mathrm{x}_0, t_0) = \sum_{j=1}^{M} \mathbb{P}\left\{R_j \text{fires in}[t, t + dt)\right\} \mathbb{P}\left\{\mathrm{x} - v_j, t|\mathrm{x}_0, t_0\right\} \\ + \mathbb{P}\left\{\text{no reaction fires in}[t, t + dt)\right\} \mathbb{P}\left\{\mathrm{x}, t|\mathrm{x}_0, t_0\right\}\end{aligned} \tag{2.2}$$

where $\mathbb{P}\left\{\text{no reaction fires in}[t, t + dt)\right\}$ denotes the probability that no reaction fires in the infinitesimal time interval $[t, t + dt)$. Note that when the state vector $\mathrm{x} - v_j$ gives negative populations, the probability $\mathbb{P}\left\{\mathrm{x} - v_j, t|\mathrm{x}_0, t_0\right\}$ in Equation 2.2 is zero because the populations of species must be positive.

The probability that reaction R_j fires in the next infinitesimal time interval $[t, t + dt)$ is given as:

$$\mathbb{P}\left\{R_j \text{ fires in } [t, t + dt)\right\} = a_j(\mathrm{x})dt + o(dt) \tag{2.3}$$

where the $o(dt)$ is used to express that it asymptotically approaches zero faster than dt. In other words, the probability that there is more than one firing of R_j in an infinitesimal time interval $[t, t + dt)$ is in the order of $o(dt)$ and thus it is negligible.

The probability that no reaction fires in the infinitesimal time interval $[t, t+dt)$ can be computed as:

$$\begin{aligned}\mathbb{P}\left\{\text{no reaction fires in}[t, t+dt)\right\} &= \prod_{j=1}^{M}\left(1-\mathbb{P}\left\{R_j \text{ fires in } [t, t+dt)\right\}\right) \\ &= \prod_{j=1}^{M}\left(1-a_j(\mathrm{x})dt+o(dt)\right) \\ &= 1-\sum_{j=1}^{M} a_j(\mathrm{x})dt+o(dt)\end{aligned} \tag{2.4}$$

Substituting Equation 2.3 and 2.4 into Equation 2.2 gives

$$\begin{aligned}\mathbb{P}\left\{\mathrm{x}, t+dt|\mathrm{x}_0, t_0\right\} = &\sum_{j=1}^{M}\mathbb{P}\left\{\mathrm{x}-v_j, t|\mathrm{x}_0, t_0\right\}\left(a_j(\mathrm{x}-v_j)dt+o(dt)\right) \\ &+\mathbb{P}\left\{\mathrm{x}, t|\mathrm{x}_0, t_0\right\}\left(1-\sum_{j=1}^{M} a_j(\mathrm{x})dt+o(dt)\right)\end{aligned} \tag{2.5}$$

Subtract $\mathbb{P}\left\{\mathrm{x}, t|\mathrm{x}_0, t_0\right\}$ from both sides of Equation 2.5 divide through by dt and finally consider the limit $dt \to 0$ with a remark that $lim_{dt\to 0} o(dt)/dt = 0$; this results in

$$\begin{aligned}\frac{d\mathbb{P}\left\{\mathrm{x}, t|\mathrm{x}_0, t_0\right\}}{dt} = &\sum_{j=1}^{M}\left(a_j(\mathrm{x}-v_j)\mathbb{P}\left\{\mathrm{x}-v_j, t|\mathrm{x}_0, t_0\right\}\right) \\ &-\mathbb{P}\left\{\mathrm{x}, t|\mathrm{x}_0, t_0\right\}\sum_{j=1}^{M} a_j(\mathrm{x})\end{aligned} \tag{2.6}$$

Equation 2.6 is called the Chemical Master Equation (CME). It is in fact a collection of differential equations in which each differential equation represents the probability of each possible state of the system at the time t. Thus, CME provides a complete description of the time evolution of the grand probability $\mathbb{P}\left\{\mathrm{x}, t|\mathrm{x}_0, t_0\right\}$.

The solution of CME gives the probabilities of all possible states at any time; however, directly solving CME poses a lot of computational challenges. An analytical and/or direct numerical approach to solve CME in general is non-trivial and difficult to find, except for rather simple cases. These difficulties have motivated in defining alternative techniques to find its solution, leading to the definition of stochastic models. Stochastic simulation

algorithm can precisely be defined from the CME, so that the probability density function of the defined stochastic process is the exact solution of the CME.

2.4 Representation of stochastic chemical kinetics

To define stochastic simulation of biological processes, vector-based representation is introduced (Gillespie, 1976).

Consider a well-mixed biochemical system consisting of N molecular species $S_1, \ldots, S_N$. These species interacts through M chemical reactions $R_1, \ldots, R_M$.

The state of the system, at time t, is represented by an N-dimensional integer value vector $X(t) \in \mathbb{N}^N$ such that

$$X(t) = (X_1(t), \ldots, X_N(t))^T$$

where $X_i(t)$ denotes the population of species S_i at time t. In general, it is denoted as $X(t) = \mathrm{x}$.

A reaction R_j is given by its associated state-change vector $v_j \in \mathbb{N}^N$

$$v_j = (v_{1,j}, \ldots, v_{N,j})^T$$

where $v_{i,j}$ denotes the change in the molecular population of S_i caused by one reaction R_j. When informally seeing the structure of a chemical reaction, one can say that some molecules are created and others are destroyed by the reaction. More precisely, if in a reaction ω molecules of a species appear as reactants and ω' appear as products, then if $\omega > \omega'$; $|\omega' - \omega|$ molecules are consumed, if $\omega < \omega'$; $(\omega' - \omega)$ molecules are created and finally, if $\omega = \omega'$, the reaction does not affect the species. According to this consideration, the state-change vector is defined as $v_{i,j} = (\omega' - \omega) = \Delta\omega$ so that

$$v_{i,j} = \begin{cases} -\Delta\omega, \text{ if } R_j \text{ consumes } \Delta\omega \text{ molecules of species } X_i(t) \\ 0, \text{ if } R_j \text{ does not affect species } X_i(t) \\ \Delta\omega, \text{ if } R_j \text{ creates } \Delta\omega \text{ molecules of species } X_i(t) \end{cases}$$

The stoichiometry matrix $D \in \mathbb{N}^{N \times M}$ can be defined from the state-change vector of the reactions as

$$D = [v_1, v_2, \ldots, v_M]$$

A consequence of this algebraic representation shows the semantics of firing a chemical reaction which turns out to be represented as simple vector summation. Given $X(t) = \mathrm{x}$ and the firing of reaction R_j modifies the state based on the equation

$$\mathrm{x}' = \mathrm{x} + v_j \tag{2.7}$$

where x' denotes the new state vector; where the reactants are removed and the products are inserted, which can be easily verified. The Equation 2.7 is demonstrated using a simple example: consider molecules of species A, B and C, and two reactions

$$R_1 : A + B \xrightarrow{k_1} C$$

$$R_2 : C \xrightarrow{k_2} C + A$$

The state vector for R_1 and R_2 is, $X(t_0) = \mathrm{x}_0$. The state-change vectors v_1 and v_2 can be given as

$$\mathrm{x}_0 = \begin{pmatrix} n_A \\ n_B \\ n_C \end{pmatrix} \qquad v_1 = \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix} \qquad v_2 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

where n_A, n_B and n_c represent the number of molecules of species A, B and C. The changes induced by the firing of R_1 is given in v_1. The first component of the v_1, which refers to species A, is -1, since one molecule A is consumed. In contrast, the changes induced by the firing of R_2 is given in v_2, the third component of vector v_2, is 0; since one molecule C is consumed/produced and at the same time, it appears as a reactant and a product. For such a system the stoichiometry matrix is defined as $D \in \mathbb{N}^{3 \times 2}$

$$D = [v_1 \quad v_2] = \begin{bmatrix} -1 & 1 \\ -1 & 0 \\ 1 & 0 \end{bmatrix}$$

The sequential firing of reaction R_1 and R_2 changes the state vector $X(t_0)$ as

$$X(t_1) = \mathrm{x}_0 + v_1 = \begin{pmatrix} n_A \\ n_B \\ n_C \end{pmatrix} + \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} n_A - 1 \\ n_B - 1 \\ n_C + 1 \end{pmatrix} = \mathrm{x}_1$$

$$X(t_2) = \mathrm{x}_1 + v_2 = \begin{pmatrix} n_A - 1 \\ n_B - 1 \\ n_C + 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} n_A \\ n_B - 1 \\ n_C + 1 \end{pmatrix}$$

where t_1 and t_2 are the times at which reactions R_1 and R_2 fire, respectively.

2.5 The notion of propensity function

Given $X(t) = \mathrm{x}$, each reaction is associated with a propensity function $a_j(\mathrm{x})$ to each R_j as mentioned in Gillespie (1976); Gillespie (1977). $a_j(\mathrm{x})dt$ is the probability of reaction R_j to fire in state x in the next infinitesimal time $[t, t + dt)$. The type of reaction determines the propensity function. Table 2.2 summarizes the analytical form of the propensity functions for chemical reactions, as originally defined (Gillespie, 1977). Hence $[A]$ denotes the number of molecules A in the system state and $X_i(t)$ denotes species A assigned to location i in $X(t)$. It is fairly easy to notice that well-stirred assumption gives rise to the combinatorial form of such functions. Finally, it is important to note that, for the reactions in Table 2.2, if $[A] = 0$, then the appropriate propensity function evaluates to 0, practically defining the non-applicability of the reaction in the current state because of the absence of the reactants. Of course, this holds only for reactions requiring non-empty reactants; in this case only for the first order and the second order.

For example, consider reactions $R_1 : A + B \xrightarrow{k_1} C$ and $R_2 : C \xrightarrow{k_2} C + A$. The propensity functions for R_1 and R_2 is given as:

$$a_1(\mathrm{x}) = k_1 X_1(t) X_2(t) \qquad\qquad a_2(\mathrm{x}) = k_2 X_3(t)$$

for state vector $X(t) = \mathrm{x}$ since, species A, B and C are mapped to locations 1, 2 and 3 in the state vector, respectively.

Table 2.2: Analytical form of the propensity functions.

Type	Reaction	Propensity
zero order	$\phi \xrightarrow{k} B$	k
first order	$A \xrightarrow{k} B$	$k[A]$
second order	$2A \xrightarrow{k} B$	$k[A]\frac{([A]-1)}{2}$

2.6 The Stochastic Simulation Algorithm

The Stochastic Simulation Algorithm (SSA) (Gillespie, 1976; Gillespie, 1977; Gillespie and Petzold, 2006) is an alternative approach to solve CME by producing possible realizations of the Equation 2.6.

Algorithm 1 Stochastic Simulation Algorithm

1: Initialize the time $t = t_0$ and the system's state $\mathrm{x} = \mathrm{x}_0$, final time T
2: With the system in state x at time t, evaluate all the $a_j(\mathrm{x})$ and sum $a_0(\mathrm{x}) \leftarrow \sum_{j=1}^{M} a_j(\mathrm{x})$
3: let $r_1, r_2 \sim U[0,1]$
4: $\tau \leftarrow a_0(\mathrm{x})^{-1} ln(r_1^{-1})$
5: let j such that $\sum_{i=1}^{j-1} a_i(\mathrm{x}) < r_2 \cdot a_0(\mathrm{x}) \leq \sum_{i=1}^{j} a_i(\mathrm{x})$
6: $\mathrm{x} \leftarrow \mathrm{x} + v_j$
7: $t \leftarrow t + \tau$
8: Go to step 2 or stop.

The SSA procedure is outlined in Algorithm 1. The input of SSA is an initial simulation time t_0; a maximum simulation time T and an initial state x_0 such that $X(t) = \mathrm{x}_0$. At each step of the algorithm, two decisions are taken: when to fire the next reaction and what reaction it can be. Given the system in state x at time t, the putative time τ for the next reaction to fire is chosen by sampling an exponentially distributed random variable such that

$$\tau \sim Exp(a_0(\mathrm{x}))$$

where $a_0(\mathrm{x}) = \sum_{j=1}^{M} a_j(\mathrm{x})$. The sampling of τ is obtained by the Inverse Monte-Carlo Algorithm by using uniformly distributed numbers r_1 and r_2, generated in step (3). Once τ is sampled, another random variable with values in $1, \ldots, M$ denoting the type of reaction to fire at time $t + \tau$, is sampled in step (5) according to the following inequalities

$$\sum_{i=1}^{j-1} a_i(\mathrm{x}) < r_2 \cdot a_0(\mathrm{x}) \leq \sum_{i=1}^{j} a_i(\mathrm{x})$$

which model a probabilistic choice dependent on the evaluations of the propensity functions for the M reactions. Again, this means that every reaction is chosen with weighted probability $\frac{a_j(\mathrm{x})}{a_0(\mathrm{x})}$. In fact another formulation for such a choice is given by $j = min\{n | r_2 \cdot a_0(\mathrm{x}) \leq \sum_{i=1}^{n} a_i(\mathrm{x})\}$. When both the variables have been sampled, the system state is updated performing the firing of R_j and setting time to $t + \tau$, as given in steps (6) and (7). Notice that, even if it may seem an intuitive interpretation that the values for τ represent the durations of the reactions (i.e. a reaction starts firing at time t and completes at time $t + \tau$), the interpretation turns out to be confusing when introducing the notions of delay in stochastic simulation algorithms. In fact, it appears from the discussion on the mathematical foundations of the SSA, that the values of τ represent time instants in which the system state is left unchanged. Indeed, the correct interpretation keeping the system at time t, is that the system is left unchanged in $[t, t + \tau)$ which then performs an instantaneous change by firing a reaction at time $t + \tau$.

Mathematical foundations of the SSA

The mathematical foundations of a simple algorithm SSA can be precisely investigated. Here, two points have to be specifically discussed.

- Why the putative time for the next reaction to fire is an exponential random variable?
- Why the reaction to fire is chosen with weighted probability?

Given $X(t) = \mathrm{x}$, it denotes the probability of the next reaction to fire at time $t + \tau$ as $p(\tau, j | \mathrm{x}, t)$, and the probability of the reaction to fire is R_j,

given that it is going to fire at $t + \tau$, as $p(j|\tau; \mathrm{x}, t)$. Here, $p(\tau, j|\mathrm{x}, t)$ is a probability density function of a continuous random variable assuming values in $[\infty, 0)$, and $p(j|\tau; \mathrm{x}, t)$ is a probability mass function of a discrete random variable assuming values in $[0, M]$. The algorithm is correct if and only if the continuous random variable turns out to be exponentially distributed, and the discrete random variable too has weighted probability dependent on the propensity functions.

An example computation

Consider a system described by an initial state x_0 and two reactions R_1 and R_2. Assume the initial state x_0 to be such that the reactions can fire an arbitrary amount of times.

Some steps of the computation $SSA(t_0, \mathrm{x}_0, T)$ are shown, where $T > t_0$. To shorten the notation, $X(t') = \mathrm{x}'$ is used to denote the assignments of the variables $t \leftarrow t'$ and $\mathrm{x} \leftarrow \mathrm{x}$. Initially, the propensity functions are evaluated so that $a_0(\mathrm{x}_0) = a_1(\mathrm{x}_0) + a_2(\mathrm{x}_0)$ and the putative time for the next reaction to fire is generated as $\tau_1 \sim Exp(a_0(\mathrm{x}_0))$; each of the reaction is chosen to fire with probability either $\frac{a_1(\mathrm{x}_0)}{a_0(\mathrm{x}_0)}$ or $\frac{a_2(\mathrm{x}_0)}{a_0(\mathrm{x}_0)}$. If R_1 is chosen, $X(t_0 + \tau_1) = \mathrm{x}_0 + v_1$, otherwise $X(t_0 + \tau_1) = \mathrm{x}_0 + v_2$.

Assume to fire reaction R_1 and $t_0 + \tau_1 < T$. In the next step of the algorithm, the propensity functions are evaluated so that $a_0(\mathrm{x}_0+v_1) = a_1(\mathrm{x}_0+v_1)+a_2(\mathrm{x}_0+v_1)$ and the putative time for the next reaction to fire is generated as $\tau_2 \sim Exp(a_0(\mathrm{x}_0+v_1))$; again each reaction is chosen to fire with probability either $\frac{a_1(\mathrm{x}_0+v_1)}{a_0(\mathrm{x}_0+v_1)}$ or $\frac{a_2(\mathrm{x}_0+v_1)}{a_0(\mathrm{x}_0+v_1)}$. If R_1 is again chosen, $X(t_0 + \tau_1 + \tau_2) = \mathrm{x}_0 + 2v_1$, otherwise $X(t_0 + \tau_1 + \tau_2) = \mathrm{x}_0 + v_1 + v_2$. A graphical representation of these sequences of steps for the SSA is given in Figure 2.1, where R_1 fires first, and R_2 second.

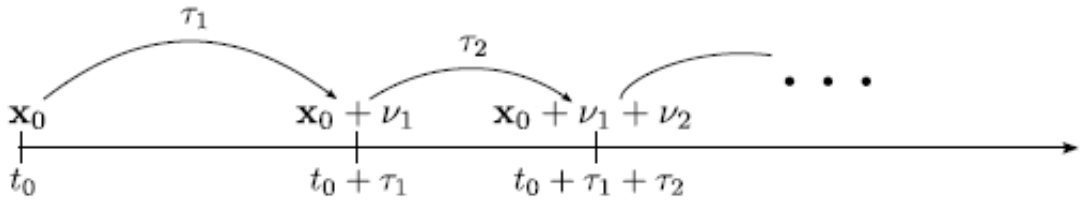


Figure 2.1: A graphical representation of SSA computation.

2.7 Summary

This chapter presents basic concepts which is required to understand results outlined in Chapter 4 and Chapter 5. Section 2.1 introduces some well known distributions from probability theory. Section 2.2 presents stochastic models. Section 2.3 presents Chemical Master Equation. Section 2.4 contains representation of stochastic chemical kinetics. Section 2.5 presents the notion of propensity function. Section 2.6 gives basic working of Stochastic Simulation Algorithm.

Chapter 3

Literature Survey

Gene regulatory models are estimated by numerical or analytical approaches. Providing nonintuitive insights into gene regulatory models [1] can be highly effective for both, numerical or analytical approaches. This chapter briefly discusses the various parameter inference approaches based on intrinsic noise in gene regulatory models. Also, model selection is reviewed briefly and discussed.

3.1 Introduction

It is often necessary to rely on inference approach, based on observable information, to understand the inner mechanism of the cellular system; the direct observations or measurements of the values of parameters are rarely possible. In the case of gene regulatory model, the modern experimental technologies may allow researchers to obtain the following information:

1. Single-cell level mRNAs or proteins expressions measured at steady state.

2. Snapshots of single-cell level mRNAs or proteins expressions collected at various time stages using the samples from the same populations.

3. The temporal tracking of mRNA or protein molecules of individual cells.

[1] **Note:** gene regulatory models and gene expression models are used interchagbly, wherever applicable.

From the viewpoint of Chemical Master Equation (CME) model, such observations yield direct information on the equilibrium distribution; the temporal distribution at different times or realizations of the trajectories of CME. Here, the existing inference approaches that can utilize information mentioned above to draw the meaningful conclusion of the underlying model is discussed.

3.1.1 Analytical approach

The extent to which a given model can be solved analytically, often plays a significant role in determining the inference approaches. If one can obtain the analytical solution of the given CME, likelihood-based approaches will be an obvious choice for inferring the unknown parameters. For instance, considering the two-state model, where the gene can switch between active and inactive states, the steady-state distribution of mRNA copy number can be solved analytically (Raj et al., 2006). The key parameters, such as the rates of activation and deactivation of mRNA can then be inferred directly, using the maximum likelihood approach (Raj et al., 2006; Tan and Oudenaarden, 2010). Also, based on the observation; in the inference of complex dynamical model, it is often hard to know in advance whether the model is adequate for explaining the data or whether the unknown parameters are identifiable (Tan and Oudenaarden, 2010). Thus, the validation of the inference results requires additional measures. For instance, by comparing the experimental data and the simulated samples from the model specified by the estimated parameters, the adequacy of the model can be assessed (Raj et al., 2006). The identifiability of the parameters can be investigated, by a thorough searching of the parameters space, to see if the model can also be fitted with other values (Zenklusen et al., 2008).

So et al. (2011) have inferred parameters from the two-state model, by fitting the analytical formulas of fano factor and the square coefficient of variation to the observed values. The similar method is used by Gandhi et al. (2011), regarding the coordination of genes during cell divisions. The

dependence between genes introduced by cell divisions has been explored by numerically fitting the analytical expression of the covariance between gene expressions. The moment-based approach can also be used to distinguish different model assumptions. Singh et al. (2012) have used the analytical formula of fano factor as the basis for inferring the source of noise in protein level. This has been done to find out whether the noise is due to the Poisson fluctuation in RNA numbers or by the stochastic transitions between different states of the gene.

If exact moment equality cannot be derived, the moment closure methods are usually employed to establish approximated expressions of key moments for inference purpose. Milner et al. (2013) have studied the inference of model parameters based on time series observations of CME system and modeled the observed data as Gaussian distributed random variables, whose means and variance are determined by moment closure scheme. Kugler (2012) has also considered a similar approach, but has focused on fitting the parameter by minimizing the distance between the observed moments and the moments predicted by the model. For the biological systems with rational propensity functions, Pedraza and Oudenaarden A (2005) have proposed moment closure schemes for three genes system. The interactions are modeled by Hill functions, applying linear expansion around the steady state. Achimescu and Lipan (2006), Raffard et al. (2008) have explored the inference in a single gene system with mRNA and protein species, where the propensity function is considered as rational functions.

To quantify the uncertainty of the inferred parameters through moment-based approach (Zechner et al., 2012), state that, due to the large number of cells measured simultaneously in cytometry experiments, it is reasonable to expect that the sample size is large. Hence, the empirical moments follow normal distribution, whose means and variances can be expressed as functions of moments. As a result, given suitable moment closure scheme; to determine the dependency of particular moments on unknown parameters, it is possible to quantify the uncertainty of estimations, using the frequentist property of maximum likelihood estimator

or by employing the posterior distribution, in case of the assignment of prior distribution to the unknown parameters. Similar methods have been presented and can be refered to (Ruess and Lygeros, 2013, 2015; Schilling et al., 2016).

The chosen closure scheme, serving only as an approximation of the system under investigation, is another primary source of uncertainty. The reasonable level of approximation is attained by applying either different moment closure schemes on different systems, or different value of parameters for the same system. However, the error introduced by moment closure scheme is often hard to evaluate in practice. Schilling et al. (2016) have proposed an adaptive algorithm to handle this issue. In this approach, given the current parameter values, the samples are generated using stochastic simulation algorithm. The fitness, of the employed moment closure schemes, is evaluated by the discrepancy between the simulated samples and observations. The adaptive algorithm selects the most appropriate moment closure schemes and also adopt different schemes in different parts of the parameter space.

Shortcoming of analytical approach

The inference methods discussed so far are based on analytical formulas. While such approaches are often easy to implement and require minimal computation resources, there are several potential drawbacks. As mentioned earlier, the exact analytical formulas cannot be established for most CME models and are forced to adopt approximation formulations. However, the discrepancy, between the approximation formulations and the true model, is often hard to evaluate; it is also hard to quantify the bias and uncertainty introduced by the approximation schemes during the inference procedure. In addition, certain summary statistics (such as the moments) of the observed data is used in the approximation formulations and thus may not be able to fully utilize the information.

3.1.2 Numerical approach

In this section, the inference approaches that utilize the numerical method to bridge the gap between the data and model parameters are investigated.

Inference of discrete stochastic model

The numerical solution of CME system, with the desired precision, can be calculated using Finite State Projection (FSP) approach. Such a numerical solution can then allow inferring the unknown parameters, by searching the parameter space and locating the values that minimize the distance between numerical solution and the empirical distribution. The CME model on the *lac operon of Escherichia coli.* has been fitted by numerically searching the values of parameters, minimizing the L1 distance between the FSP solution and observed distribution (Munsky et al., 2009). In calculating this distance metric, measurements are obtained under various conditions and have been assigned different weights for specifying the relative importance. The detailed optimization procedure includes a random initial guess; the value is then updated, using gradient-based and simulated annealing searches. Similar methods are presented in (Neuert et al., 2013; Senecal et al., 2014; Shepherd et al., 2013).

The numerical solution of CME is recalculated throughout the optimization algorithm in the aforementioned FSP based inference approach. Even though the FSP method can reduce the computational burden of finding the numerical solution of CME significantly, it can still be computationally very demanding, particularly when the dimension of parameter space is large. In this regard, it can be worthy to consider likelihood-free inference approach to avoid the difficulty of finding the solution of CME. In particular, the Bayesian method is known as Approximate Bayesian Computation (ABC) (Beaumont et al., 2002; Pritchard et al., 1999; Tavare et al., 1997) and can be used for such a purpose. In a standard ABC rejection algorithm, a particle θ^* is firstly sampled from the prior distribution of the unknown parameter, and is used to generate a simulated data set X_{θ^*}. The proximity between X_{θ^*} and the observed data

set X can then be evaluated, based on a chosen distance metric. The decision on whether to reject or accept the particle θ^* is then taken on the basis of whether the distance is greater or smaller than a predefined threshold ϵ. This procedure necessarily allows to obtain independent sample of θ from density $p(\theta|d(X, \widehat{X}) < \epsilon)$, which can be regarded as a reasonable approximation to the posterior distribution $p(\theta|X)$ for small ϵ. Thus, the posterior samples of parameters, without evaluating the likelihood function, can be obtained, as long as samples from the given model with specified parameters can be simulated. ABC can be a suitable choice for inferring parameters in CME.

The acceptance rate of particles determines the efficiency of an ABC sampling algorithm. By adopting Markov Chain Monte Carlo (MCMC) method (Marjoram et al., 2003) and sequential sampling technique (Liepe et al., 2014; Toni et al., 2009), the acceptance rate of particles can be improved. The ABC SMC algorithm utilizes a gradient of thresholds $(\epsilon_1, \epsilon_2, \ldots, \epsilon_T)$ in strictly decreasing order with ϵ_T as the desired threshold. The sampled particles are propagated through a sequence of intermediate distributions, corresponding to the intermediate thresholds, until the final set of samples represents the posterior target distribution.

The control of the false rejection error is another critical issue in applying ABC algorithm. The error rejects proposed particle, even though the distribution of experimental data is consistent with the distribution of simulated data. This error is caused by using the finite size of samples, to approximate the true distribution, as defined by the particle. This error can be reduced by increasing the size of simulated data set at the price of increasing computational cost. The ABC algorithm named INSIGHT shows that, if the distance metric used for analyzing flow cytometry data (Lillacci and Khammash, 2013), is Kolmogorov distance, the false rejection error then depends on the specified threshold ϵ; the size of experimental and simulated data. Moreover, the size of experimental data, in a typical flow cytometry experiment, is often large, whereas, the required size of the sample for attaining a reasonable false rejection error can be surprisingly small. As long as the size of observed data is large, the ABC algorithm can be implemented

in a very efficient way. In addition, the use of Kolmogorov distance allows the bounds of a mismatch index to be estimated, and is defined as the distance between the distribution of experimental data and the distribution of best fitted model. This index grants valuable insight on the fundamental discrepancy between experimental data and the stochastic model and can be used to determine whether alternative models may be investigated or not.

ABC method also opens the possibility of using Bayes factor or posterior probability to compare competing models (Liepe et al., 2014; Toni et al., 2009). For instance, the ABC SMC algorithm (Toni et al., 2012), is used to study the MEK/ERK phosphorylation dynamics using time course data obtained from vivo cells. The estimated posterior probabilities are then used to rank candidate models that represent different hypothesis on the underlying systems. Moreover, by comparing the simulated samples and the observed data, it also makes a direct diagnosis of the discrepancy between the model and the data possible (Ratmann et al., 2009).

The inference problem can be handled in a different manner if the expressions of mRNA or protein can be monitored within individual cells, continuously (Golding et al., 2005; Yu et al., 2006). In particular, as described in Gillespie algorithm (Gillespie, 1977), given the complete information on a particular trajectory of CME over time $[t_0, t_n]$, (including the initial copy number(s) $x(t_0)$, as well as the firing times of each reaction up to time t_n), the likelihood function is expressed as the product of exponential and multinomial densities. The corresponding inference problem can then be easily solved. For instance, given the full trajectory, in a stoichiometric system, where the propensity functions are linear functions of the unknown parameters, the maximum likelihood estimator can be solved analytically (Daigle et al., 2012).

Nevertheless, the complete information is hard to obtain. In practice, the system state can be observed at a few discrete time points. The observed data is represented as $\mathrm{x}(t_0), \mathrm{x}(t_1), \ldots, \mathrm{x}(t_n)$, the likelihood function is then the products of transition likelihood $p(\mathrm{x}(t_i)|\mathrm{x}(t_{i-1}), \theta)$ whose expression is usually not analytical. For example, considering a single molecular species that evolves

according to a simple birth and death process, if the copy numbers are 10 and 20 at time 0 and t respectively, then any full trajectory that satisfies the following condition is consistent with the observation:

1. The total number of births minuses the total number of deaths during $(0, t]$ equals 10.

2. The birth events and the death events can occur in any order and at any time as long as the total copy number never drops below 0.

Consequently, transition probability from 0 and t is the sum of probabilities of all the consistent full trajectories, which can be hard to compute if the system is complex.

Many authors have thus explored approximated approaches to estimate the transition probability, so that the unknown parameters can be inferred with conventional methods. Reinker et al. (2006), under the assumption that the number of firings are limited or the propensity functions remain constant during the period $(t_{i-1}, t_i]$, show that the transition probability can be approximated with relatively simple analytical formulas. This approach is roughly equivalent to approximating the exact transition probability, as the sum of probabilities of the most probable paths from t_{i-1} to t_i. The transition likelihood from t_{i-1} to t_i is estimated using non-parametric kernel density function based on the simulated realizations of the system at t_i given initial condition $(t_{i-1}, \mathrm{x}(t_{i-1}))$ (Tian et al., 2007).

The maximum likelihood estimator of parameters can also be found using Expectation Maximization (EM) algorithm. In the E-step, the expectation of the likelihood function of the full trajectory, conditional on the observations and the current value of parameters, is evaluated. Then in the M-step, the value of parameters can be updated by maximizing the conditional expectation. Due to the intractability of EM algorithm, the Monte Carlo extension of EM algorithm (MCEM) is often used. In MCEM, the conditional expectation is estimated, based on the sampled, full trajectories. The major difficulty in applying MCEM in CME system lies in the fact that, the simulated trajectories must be consistent with the observed data. This

can be hard to achieve, if unmodified Stochastic Simulation Algorithm is used. Horvath and Manini (2008) suggest that the full path must be simulated piece-wise for each interval $(t_{i-1}, t_i]$. Daigle et al. (2012) argue that, to implement MCEM efficiently, the initial choice of parameters must be the values that are likely to generate consistent trajectories. An iterative algorithm based on the Cross Entropy (CE) method (Rubinstein, 1997) is used to find such initial values. In each iteration, trajectories are simulated using previous parameter values but, only the trajectories that are closed to the observed path are used for updating the parameters.

Wang et al. (2010) have presented an approach to maximize likelihood function using Stochastic Gradient Descent (SGD). The gradient of likelihood function can be determined based on the expectation of the duration, in which the system stays on different states and the number of transitions between states are conditional on the observed path. A Reversible Jump Markov Chain Monte Carlo (RJMCMC) algorithm, is implemented for simulating paths that are consistent with the observations, where new paths are proposed by adding/deleting certain set of reactions from the initially proposed path. This method can also be applied to the dataset, where only part of the species is observed.

In addition to the frequentist approaches, Bayesian methods that utilize MCMC sampling algorithms can also be used to solve such problems. Boys et al. (2008) use MCMC algorithm to sample the full trajectories, conditional on the observations. The efficiency of MCMC sampling is improved by using reversible jumping and blocking update methods. In general, the Bayesian approach can be directly applied to the system with unobserved species, since Bayesian approach can readily impute such missing information in the same way as imputing the full trajectories.

Inference of continuous stochastic model

The discreteness of CME is often the major obstacle in obtaining its solution. It can be approximated by other continuous stochastic processes, including the Linear Noise Approximation (LNA) and the Stochastic Differential

Equation (SDE). In this section, existing inference approaches for the continuous stochastic models is discussed using time-series data.

LNA approximates the CME as the sum of deterministic term and stochastic fluctuation. As is mentioned in Komorowski et al. (2009), the stochastic fluctuation can be modeled by SDE. The drifting and diffusion parameters of SDE depend on the deterministic part of LNA. As a result, the solution which is proposed by LNA is always multivariate Gaussian distribution, whose mean vector and covariance matrices can be determined by the propensity functions. Thus, the posterior distribution can be sampled, directly using standard Metropolis-Hastings (MH) algorithm; given the suitable choice of prior values over the unknown parameters. This framework can readily include the presence of unobserved species and also the measurement errors (assuming to be an additive Gaussian noise). This method has been employed to estimate the GFP protein degradation rate from the cycloheximide experiment. Fearnhead et al. (2014) also consider the inference problem using LNA and show that such approach can be statistically and computationally more efficient than approaches based on deterministic differential equation or SDE.

In a full SDE approximation, unlike in LNA, the transition probability between two successive observations is often analytically intractable. It is possible to estimate such transition probability by discretizing the trajectory of SDE system; a technique commonly known as Euler-Maruyama approximation. Under this approximation, the sample path between two successive observations is discretized into multiple segments. The increment in each segment is then modeled as independent Gaussian random variables, whose means and values are determined by SDE. This approximation forms the basis of the Bayesian inference framework proposed for the general stoichiometric model (Golightly and Wilkinson, 2005). A MCMC scheme is then applied to obtain posterior samples of unknown kinetic parameters. The sampling procedure is alternative, between the sampling of parameters conditional on the augmented data, the sampling of missing data given observations and the current set of parameters, due to the need of imputing

values to discretize the SDE, as well as handling the unobserved species. Further enhancement of this scheme is possible with advanced sampling methods. The dependence between the parameters and missing data can be overcome through the use of sequential MCMC methods to sample the model parameters (Golightly and Wilkinson, 2006). The accuracy of Euler-Maruyama approximation increases as the number of imputed value increases. However, increasing the number of imputed values also increases the computational cost which can break down an ordinary Bayesian imputation algorithm. To overcome this issue, Golightly and Wilkinson (2008) have proposed a global MCMC strategy with an improved Gibbs sampler. In this approach, a Brownian motion process is used, to impute values between successive observations, so as to prevent the increase in computational cost as the number of segments increase. However, the speed of such computation scheme is still hindered by the complexity of the model, and for this particle, Markov Chain Monte Carlo method can be implemented to lessen the computation burden (Golightly and Wilkinson, 2011).

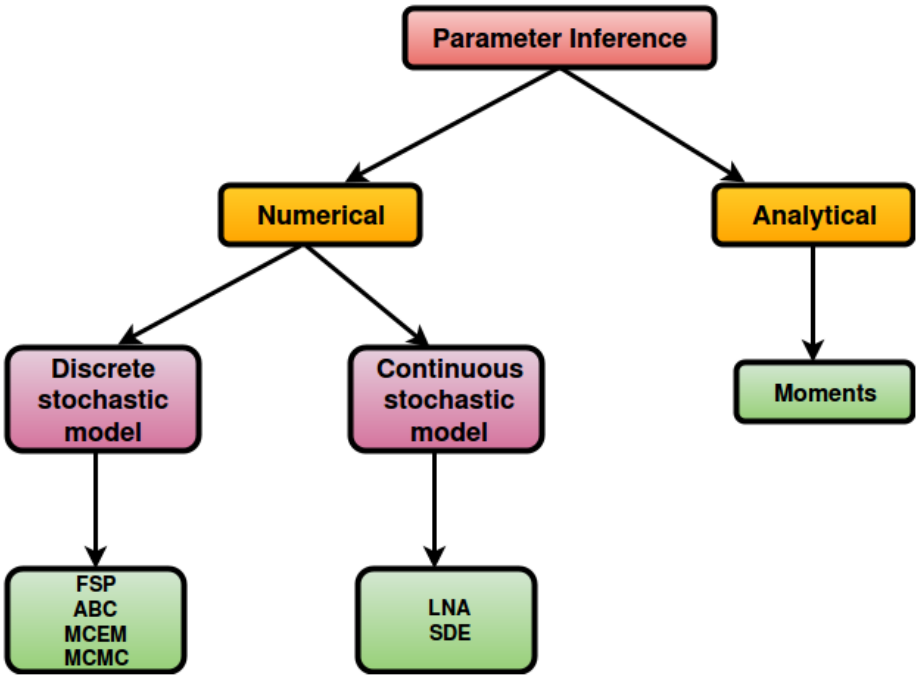


Figure 3.1: Categorization of different approaches for parameter inference.

3.2 Model selection

A model can be proposed with different level of details for explaining the mechanics of gene regulatory system. Nevertheless, it is often difficult to observe the inner mechanism of the regulatory system directly and hence have to rely on the available information to choose between different models. For instance, there is a need to find out if it is possible to choose between the two-state model and multistate model, by analyzing the single-cell level distribution of the copy number of protein molecules. There is also a need to know how to make sure that the selected model is sophisticated enough to explain what has been observed or whether the observed information is sufficient to infer the detail of the model being proposed. In this section, the relevant literature that deals with the problem of model selection in the context of studying CME system is discussed.

The fitness of the model is evaluated by proposing suitable metrics. This

metrics can be used to measure the distance between model and data. In earlier works (Babtie et al., 2014; Kugler, 2012; Liepe et al., 2014) have used this metrics to measure the distance, between the observed and predicted values or time derivatives of state variables or between the predictive and observed moments. If the predictive distribution can be obtained, χ^2 test can be used to determine whether the prediction of model is consistent with the data (Zenklusen et al., 2008). Also, Euclidean distance or Hellinger distance can be used as metric of discrepancy or the difference (Munsky et al., 2009; Silk et al., 2014; Sunnaker et al., 2013). For comparing models with different level of details, Akaike Information Criterion (AIC) and Bayes factor can be used to penalize additional complexity of model parameters or structure (Babtie et al., 2014; Liepe et al., 2014; Silk et al., 2014; Sunnaker et al., 2013; Toni et al., 2009). The fitness and complexity of the model can also be balanced by measuring the uncertainty introduced by the model. For instance, Neuert et al. (2013), the log-likelihoods are used as measurement of fitness, while the uncertainty is evaluated by cross-validation. Specifically, the uncertainty is defined as the average log-likelihoods of complete data set calculated using parameters that are obtained through fitting the model with sampled partial data set. The best model is chosen based on the balance between fitness and uncertainty.

As many authors have pointed out, due to the complexity of dynamical system, even if good fit has been achieved, using a particular model or particular set of parameters, there may be other alternative models or sets of parameters that can fit the data equally well. Consequently, it is often useful to search the space of candidate models or the parameter space thoroughly, before making a final conclusion. Villaverde et al. (2015) have explored the predictive accuracy of fitted model using a consensus approach for a fixed model. This approach searches the parameter space of the given model and collects sets of all parameter values that fit the data well. Then the accuracy of prediction can be analyzed, based on whether these collected sets of parameter values can reach a consensus. By grouping the parameters into modules of meta-parameters, the burden of searching the parameter space

can be reduced. Topological Sensitive Analysis (Babtie et al., 2014) is used to explore the uncertainty present in the structure of the model. This approach proposes alternative structure by modifying the relationship between nodes in the given model. Restrictions are imposed to limit the search space. Gaussian process regression is used to evaluate the fitness of the proposed structure. Topological Filtering method (Sunnaker et al., 2013) is used to explore alternative models by constructing a tree of models. The base model is the root of this tree which consists of many detailed interactions. The creation of new nodes includes removing interactions and the associated parameters, step by step. Analysis on the fitness of the model is carried along the way. The process of the new nodes creation is stopped only if further simplification makes the model unfit for the data. This approach may create multiple branches and candidate models at the end of each branch. The model is then collected for further study. Finally, the exploration of alternative models may guide the researchers to design new experiments or to discriminate different models.

3.3 Summary

This chapter briefly summarizes the inference methods to infer the unknown kinetic parameters from the model using single-cell gene expression data. Also, model selection is reviewed briefly and discussed.

Chapter 4

Model Formulation for Multistep Reaction Processes

This chapter introduces the widely used gene expression model, random telegraph model and its multistep model formulation. The multistep promoter model formulation is presented for two different inference approaches that have been developed in this work, namely; Delay-Bursty MCEM and Clumped-MCEM respectively.

4.1 Introduction

A fundamental issue in Systems Biology is to formulate simple models to describe biological processes with multistep reactions. This is very important because recent theoretical and experimental studies have shown that a wide variety of biochemical events exhibits multistep reactions. Among them, the most important biological processes in gene expression, that involve multistep reaction, are transcriptional and translational processes, that produce mRNAs and proteins respectively. Transcription plays a major role in all cellular functions. The irregularity of transcription process results in diseases such as cancer, diabetes and neurological disorders (Lee and Young, 2013). Despite its importance, the mechanistic details of gene expression are not very well understood. In particular, lack of molecular-level explanation for bursts in gene expression is observed in prokaryotes and eukaryotes (Cai

et al., 2006; Raj et al., 2006). The accurate characterization of the mechanisms underlying expression bursts is very important, as the properties of these bursts have been implicated in disease related processes such as bacterial phenotype switching (Choi et al., 2008) and HIV activation (Singh et al., 2010). Recently, several works have provided proof for the synthesis of mRNAs (Chubb et al., 2006; Dar et al., 2012; Golding et al., 2005; Halpern et al., 2015; Ochiai et al., 2014; Raj et al., 2006; Senecal et al., 2014; So et al., 2011; Suter et al., 2011; Taniguchi et al., 2010; Zong et al., 2010) and proteins (Cai et al., 2006; Yu et al., 2006) in bursts. Although the origins of the transcriptional burst remain poorly understood (Chubb and Liverpool, 2010), it has been shown that stochastic switching between promoter active and inactive states leads to bursts (Blake et al., 2003; Boeger et al., 2008; Harper et al., 2011; Larson, 2011; Mao et al., 2010; Mariani et al., 2010; Miller Jensen et al., 2011; Raser and OShea, 2004; Suter et al., 2011). The random telegraph model (Dobrzyski and Bruggeman, 2009; Peccoud and Ycart, 1995; Shahrezaei and Swain, 2008) is most commonly used to analyze transcriptional bursting. This model has been used as the key model for several works to infer parameters from experimental data. In general, the assumption of random telegraph model is not valid because it involves multiple kinetic steps in promoter activation (Jia and Kulkarni, 2011; Pedraza and Paulsson, 2008; Xu et al., 2013). All these experimental facts combine with the above analysis and are motivation to introduce a more accurate model for the multistep reaction processes. Therefore, the aim of this work is to design a model which accurately characterizes transcriptional bursting and is consistent with the observed data.

4.2 Biological motivation for multistep model formulation

Several experimental studies on promoters shed light on multistep OFF mechanism that are exhibited. The effective number of steps for most

promoters, due to simultaneous regulation by multiple transcription factors, as well as chromatin modifications, is considered to be larger than two (Zhang et al., 2012). The distribution of time, the human prolactin gene promoter spents in an inactive state is inferred to be strongly non-exponential and thus indicative of multiple, sequential OFF states, as is studied by Harper et al. (2011). In particular, when promoters are modeled as an irreversible cycle, endogenous promoters show five sequential inactive steps, while minimal synthetic promoters exhibit only one (Zoller et al., 2015). P_{RM} in phage lambda, where complex mechanism of regulation gives rise to 128 regulatory states (Sanchez et al., 2013) and the Endo16 gene in sea urchin, where cis-regulatory domain contains > 30 binding sites for 15 different proteins that perform combinatorial regulation (Yuh et al., 1998), are classical examples of multistep promoters. Recently, to explore kinetic control - the combinatorial control of gene expression, through regulation of different steps in the transcription cycle, is presented (Scholes et al., 2017).

4.3 Random telegraph model

The gene expression description, widely popular for both, its simplicity and generality, is the random telegraph model. It has been first proposed by Ko (Ko M S H, 1991) and later has been expanded by Peccoud and Ycart (1995). The random telegraph model is represented by using biochemical reactions as follows:

$$\begin{aligned} DNA_{off} &\underset{k_{off}}{\overset{k_{on}}{\rightleftharpoons}} DNA_{on} \\ DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\ mRNA &\xrightarrow{\gamma_m} \phi\ . \end{aligned} \tag{4.1}$$

In model 4.1, the promoter switches, from OFF to ON and ON to OFF state, are with rate k_{on} and k_{off} respectively. The mRNA production happens from the ON state with rate k_m. mRNAs live for an exponentially-distributed time interval with mean lifetime $1/\gamma_m$, where γ_m is the rate of mRNA degradation.

4.4 Transcriptional bursting model

The representation of transcriptional bursting model based on random telegraph model is given in Model 4.2.

$$\begin{aligned} DNA_{on} &\xrightarrow{k_m} DNA_{on} + B_m \times mRNA \\ mRNA &\xrightarrow{\gamma_m} \phi \\ B_m &\sim Geometric(c_3)\ . \end{aligned} \tag{4.2}$$

Basically, transcriptional bursting is represented by two parameters; where k_m and B_m denote the burst frequency and burst size respectively. In this model formulation of 4.2 (Gillespie, 2007), mRNA bursts arrives at exponentially-distributed time intervals with rate k_m. Each burst produces a geometrically-distributed number of transcripts B_m with the mean value $(1 - c_3)/c_3$ (Evans et al., 2000).

4.5 Multistep formulation of random telegraph model

The realistic representation of multiple, sequential OFF states in 4.1 is shown in Model 4.3. Pictorial representation of Model 4.3 is depicted in Fig.4.1 and its abridged version is depicted in Fig.4.2.

$$\begin{aligned} DNA_{off_1} &\xrightarrow{k_1} DNA_{off_2} \\ &\vdots \\ DNA_{off_{N-1}} &\xrightarrow{k_{N-1}} DNA_{off_N} \\ DNA_{off_N} &\xrightarrow{k_N} DNA_{on} \\ DNA_{on} &\xrightarrow{k_{off}} DNA_{off_1}\ . \end{aligned} \tag{4.3}$$

Model 4.3 (Fig.4.1) differs from the model 4.1 in the distribution of time spent in OFF states. In contrast to 4.1, it is now non-exponential. It follows hypoexponential distribution (sum of exponential random variables) (Evans et al., 2000) which approaches an Erlang distribution (Evans et al., 2000) when switching rates are identical.

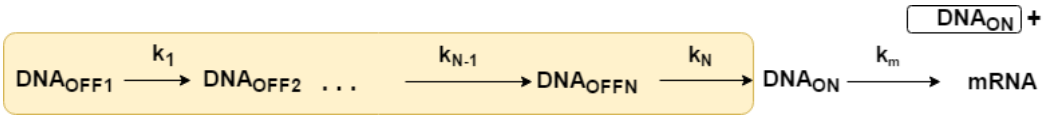


Figure 4.1: Original model of promoter activation

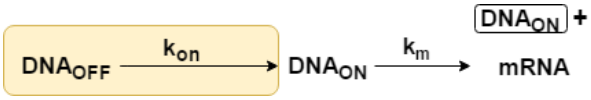


Figure 4.2: Abridged model

4.6 Model reduction stratergy for multistep transcriptional bursting model

The model reduction for multistep promoter models is achieved through the use of time delays. The key idea here is to lump multistep reaction of processes, by equivalent delayed reactions that transform reactants into products, after a prescribed time delay. A novel model reduction stratergy, that represents several OFF states, by a single state, accompanied by specifying time delays for burst frequency, is devised. This stratergy enables, both efficient simulation and parameter inference which is demonstrated in Chapter 5.

4.6.1 Multistep formulation of transcriptional bursting model

The representation of the transcriptional bursting model 4.2, in terms of 4.3, requires the generation of inter-burst arrival times. This is achieved by introducing prescribed delays for inter-burst arrival times. The corresponding bursting model is formulated as

$$\begin{aligned} DNA_{off_1} \longrightarrow \ldots \longrightarrow DNA_{off_N} &\xrightarrow{k_{on},\tau} DNA_{on} \\ DNA_{on} &\xrightarrow{k_{off}} DNA_{off_1} \\ DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA\,. \end{aligned} \tag{4.4}$$

Model 4.4 is modeled as a delayed reaction by generating delay time(τ) as an Erlang distribution. There are two points to consider:

- First, the delay time of the reaction, which is modeled as an Erlang distribution, is the time from the initiation to completion. The firing time τ of the delayed reaction is generated as $Erlang(N, k)$ distribution, in which the shape parameter N corresponds to the distribution of the sum of N and are independent exponentially distributed numbers, with the same rate parameter k.

- Second, this work focuses on delayed and nondelayed nonconsuming reactions.

When τ is set to 0, it reduces to random telegraph model where the promoter switching times from OFF to ON and ON to OFF states are exponentially distributed.

The correspondence between the parameters of the Erlang distribution and the number of promoter states is most accurate, when the switching rates are equal. When switching rates are not equal, the closed form for the sum is not known. In this case, it has been empirically observed that the slowest promoter transitions become rate limiting and thus mask the presence of faster transitions (Daigle et al., 2015).

4.6.2 Comparison with Barrio et al. 2013 paper

- The focus of the thesis is to develop computationally efficient parameter inference methods for characterizing transcriptional bursting process, for inferring unknown kinetic parameters, given single-cell time-series data. On the other hand, the work of Barrio et al. 2013 focuses on representing chains of chemical reactions by reduced models. The abridgement is achieved by generation of model-specific delay distribution functions, consecutively fed to a delay stochastic simulation algorithm. Barrio et al. also shows analytical description of delay distributions for the system which consists of first-order

reactions, with or without additional backward bypass reactions. Further, they also discussed why one must adopt numerical approach for monomolecular reactions.

- To model multistep promoter OFF states with bursting (as shown in Model 4.4), first, delay constant (τ in the Model 4.4) associated with each reaction is specified. As in Barrio et al., we assume model specific delay distribution. Our work focuses only on delayed and nondelayed nonconsuming reactions whereas Barrio et. al model reduction approaches are more universally applicable as it does not rely on time scale separation conditions. We consider reactions that follow mass action kinetics. The use of delays in our work is described below.

  - The more realistic representation of reversible reaction of Model 4.1 is given in Model 4.3. The representation of the transcriptional bursting model 4.2, in terms of 4.3, requires the generation of inter-burst arrival times. This is achieved by introducing prescribed delays for inter-burst arrival times. The corresponding bursting model is explained in Subsection 4.6.1

- The Barrio et al. work does not require any time-scale separation conditions to be accurate. Thus, Barrio et al. approach largely increases the range of reducible biochemical models. It depends on model specific delay distribution, consecutively fed to a delay stochastic simulation algorithm. The approach proposed in Barrio et al. is summarized below:

  - The approach in Barrio et al. (2013) was based on the idea of random walks and first-arrival times (Van Kampen, 2007). It considers some kind of reaction blocks that could be lumped. These restrictions were related not only to the type of reactions contained in the blocks but also to how blocks could be connected among themselves. Namely, a linear chain of reactions composing

a block

$$S_1 \xrightarrow{c_1} S_2 \underset{c_{2_r}}{\overset{c_2}{\rightleftharpoons}} \dots S_{n-2} \underset{c_n-2_r}{\overset{c_{n-2}}{\rightleftharpoons}} S_{n-1} \xrightarrow{c_{n-1}} S_n \ . \tag{1}$$

Equation (1) was shown to be exactly reducible to a single delayed reaction

$$S_1 \xrightarrow{c_1,\tau} S_{n'} \ . \tag{2}$$

with appropriate delay distribution. Here, S_1 and S_n, required the irreversibility of the first and last reaction of the linear chain.

- Barrio et al. approach also provides an exact reduction in scenarios solely composed of unimolecular and/or backward bypass reactions, as the delay distributions can be derived analytically. For all other monomolecular reactions (constitutive creation, degradation, or forward bypass reactions), Barrio et al. approach accuracy can be tailored at will, as the delay distributions can be derived numerically, either in terms of first-passage time (SSA) runs or matrix exponentials for sampled time points. In these cases, the accuracy depends on the number of SSA simulations obtained for the first-passage distribution, or the number of time points at which the matrix exponential is calculated, respectively.
- Barrio et al. work also presents to reduce models with backward and forward bypass reactions, degradation of involved molecular species, and constitutive creation of intermediate species through Arnoldi estimates (Trefethen and Bau, 1997).

- The model reduction techniques presented in Barrio et al. 2013 cannot be used for the models we study in our thesis because we consider reactions that follow mass action kinetics - i.e. where $a_j(\mathrm{x}(t)) = \theta_j h_j(\mathrm{x}(t))$. Where θ_j, a kinetic rate constant and $h_j(\mathrm{x}(t))$, a function that quantifies the number of possible ways reaction R_j, can occur, given system state x. For instance, for unimolecular,

homo-bimolecular and hetero-bimolecular reactions $h_j(\mathrm{x}(t))$ takes the form x_1, $\frac{\mathrm{x}_1(\mathrm{x}_1-1)}{2}$, $\mathrm{x}_1\mathrm{x}_2$ respectively. In our work, for the delayed nonconsuming reactions, $h_j(\mathrm{x}(t))$ is set to 1. Hence, accommodating other types of reactions using Barrio et al's delay distribution for the problem setting in our thesis is not possible.

4.6.3 Delay estimation for unimolecular and bimolecular reactions

Degradation is an essential process in all biological processes. The special unimolecular reaction $A \rightarrow \phi$ represents the degradation of species A. The reaction $\phi \rightarrow A$ is called a synthesis reaction. The A molecules are introduced into the biological system from outside, e.g., species reservoir. Synthesis reactions are often used to model the effects of outside environment on the system dynamics.

An A molecule can associate with a B molecule to produce a complex C through an association reaction $A + B \rightarrow C$. Such a reaction is called a bimolecular reaction. The special bimolecular reaction $2A \rightarrow B$ is called a dimerization, where two molecules of the same species A are consumed to produce a B molecule.

This work focuses on reactions that follow mass action kinetics - i.e. where $a_j(\mathrm{x}(t)) = \theta_j h_j(\mathrm{x}(t))$. Where θ_j, a kinetic rate constant and $h_j(\mathrm{x}(t))$, a function that quantifies the number of possible ways reaction R_j, can occur, given system state x. For instance, for unimolecular, homo-bimolecular and hetero-bimolecular reactions $h_j(\mathrm{x}(t))$ takes the form x_1, $\frac{\mathrm{x}_1(\mathrm{x}_1-1)}{2}$, $\mathrm{x}_1\mathrm{x}_2$ respectively. In this work, for the delayed nonconsuming reactions, $h_j(\mathrm{x}(t))$ is set to 1. Hence, accommodating unimolecular and bimolecular reactions using delay distribution in this setting is not possible.

4.7 Experimental data

The Delay-Bursty MCEM and Clumped-MCEM is applied to actual time-lapse microscopy data from a reporter gene driven by a mammalian

promoter. A single trajectory of luminescence data collected once every five minutes, from the glutaminase promoter is extracted (Suter et al., 2011)(Fig. 1C in Suter et al.). The data smoothing and calibration is performed to convert light intensity values to numbers of proteins. This data consists of 539 measurements, sampled approximately, once every five minutes for 43.5h arbitrary units (Daigle et al., 2015). Fig. 4.3 and Fig. 4.4 displays the glutaminase trajectory before and after preprocessing.

Note: The Bmal1a and Prl2c2 (datasets provided in (Suter et al., 2011)) includes other components for bursting patterns for example, circadian component. The goal of present work is to characterize transcriptional bursting process which involves multistep reactions. Hence, the proposed methods has been tested only with time-series data of endogenous mouse glutaminase promoter from (Daigle et al., 2015). Further, the performance of model systems other than glutaminase promoter (Daigle et al., 2015) could not be evaluated due to lack of publicly available data.

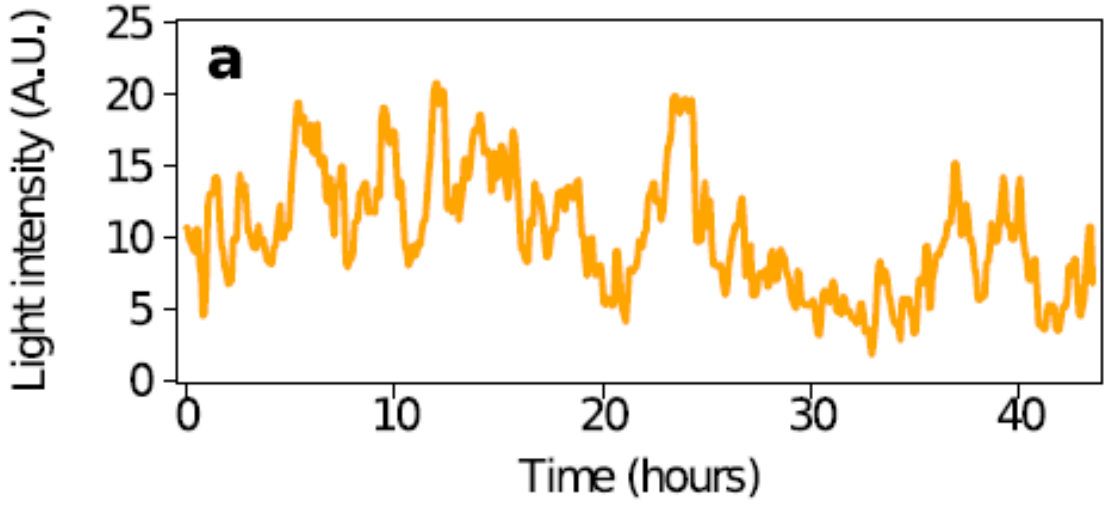


Figure 4.3: Glutaminase promoter time-lapse microscopy data from (Suter et al., 2011).

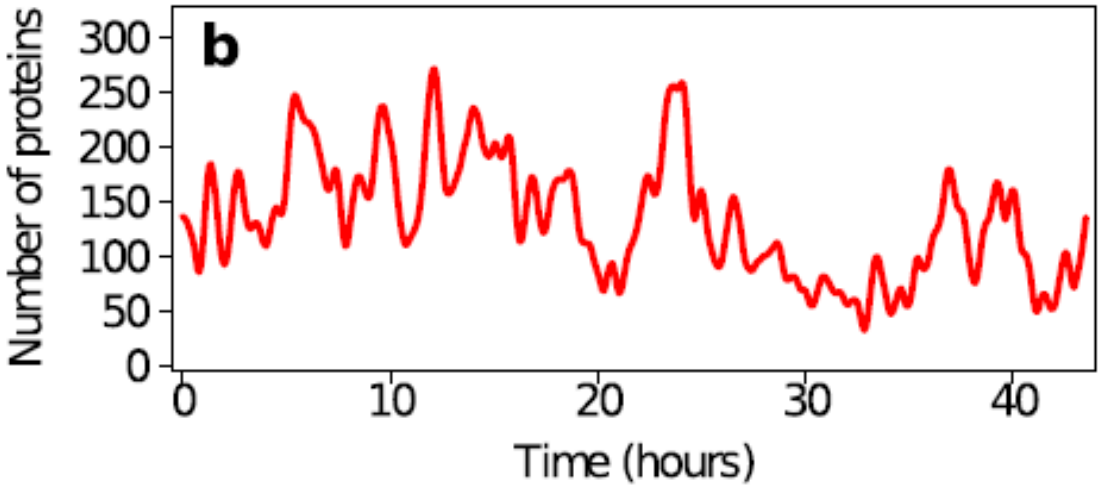


Figure 4.4: Glutaminase promoter time-lapse microscopy data from (Daigle et al., 2015).

4.8 The interpretation of experimental data for multistep models

$$
\begin{aligned}
DNA_{off_1} \longrightarrow \ldots \longrightarrow DNA_{off_N} &\xrightarrow{k_{on},\tau} DNA_{on} \\
DNA_{on} &\xrightarrow{k_{off}} DNA_{off_1} \\
DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\
mRNA &\xrightarrow{.924} \phi \\
mRNA &\xrightarrow{12.6} mRNA + Protein \\
Protein &\xrightarrow{1.98} \phi\ .
\end{aligned}
\tag{4.5}
$$

$$
\begin{aligned}
DNA_{on_1} \longrightarrow \ldots \longrightarrow DNA_{on_N} &\xrightarrow{k_{on},\tau} DNA_{off} \\
DNA_{off} &\xrightarrow{k_{on}} DNA_{on} \\
DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\
mRNA &\xrightarrow{.924} \phi \\
mRNA &\xrightarrow{12.6} mRNA + Protein \\
Protein &\xrightarrow{1.98} \phi\ .
\end{aligned}
\tag{4.6}
$$

In this study, all models share fixed, identical rates of mRNA degradation, protein translation and protein degradation.

- mRNA degradation is derived from the 45 min glutaminase reporter mRNA half life experimentally determined by (Suter et al., 2011).
- Protein degradation is derived from the 21 minute luciferase protein half life experimentally determined by (Suter et al., 2011).
- Protein translation is reported in (Molina et al., 2013).

In addition, the unobserved initial promoter state for multistep OFF model and multistep ON model is set to DNA_{off} (4.5) and DNA_{on} (4.6) respectively. The protein number is set as 137 for glutaminase dataset. For the unobserved initial number of mRNA molecules, values from 0, 10, 20, 30 are tried. However, number of $mRNAs = 20$ molecules allowed the simulation of trajectories with the largest observed data likelihood. So this number is used in all simulations. The initial value of the reaction clock is set to 0 for all model simulations.

4.9 Parameter inference using glutaminase promoter time-series data

$$\begin{aligned} DNA_{off_1} &\overset{k_{on},\tau}{\underset{k_{off}}{\rightleftharpoons}} DNA_{on} \\ DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\ mRNA &\xrightarrow{.924} \phi \\ mRNA &\xrightarrow{12.6} mRNA + Protein \\ Protein &\xrightarrow{1.98} \phi\ . \end{aligned} \tag{4.7}$$

$$\begin{aligned} DNA_{off_1} \longrightarrow DNA_{off_2} &\xrightarrow{k_{on},\tau} DNA_{on} \\ DNA_{on} &\xrightarrow{k_{off}} DNA_{off_1} \\ DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\ mRNA &\xrightarrow{.924} \phi \\ mRNA &\xrightarrow{12.6} mRNA + Protein \\ Protein &\xrightarrow{1.98} \phi\ . \end{aligned} \tag{4.8}$$

$$
\begin{aligned}
DNA_{off_1} \longrightarrow \ldots \longrightarrow DNA_{off_3} &\xrightarrow{k_{on},\tau} DNA_{on} \\
DNA_{on} &\xrightarrow{k_{off}} DNA_{off_1} \\
DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\
mRNA &\xrightarrow{.924} \phi \\
mRNA &\xrightarrow{12.6} mRNA + Protein \\
Protein &\xrightarrow{1.98} \phi \, .
\end{aligned} \tag{4.9}
$$

$$
\begin{aligned}
DNA_{off_1} \longrightarrow \ldots \longrightarrow DNA_{off_4} &\xrightarrow{k_{on},\tau} DNA_{on} \\
DNA_{on} &\xrightarrow{k_{off}} DNA_{off_1} \\
DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\
mRNA &\xrightarrow{.924} \phi \\
mRNA &\xrightarrow{12.6} mRNA + Protein \\
Protein &\xrightarrow{1.98} \phi \, .
\end{aligned} \tag{4.10}
$$

$$
\begin{aligned}
DNA_{off_1} \longrightarrow \ldots \longrightarrow DNA_{off_5} &\xrightarrow{k_{on},\tau} DNA_{on} \\
DNA_{on} &\xrightarrow{k_{off}} DNA_{off_1} \\
DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\
mRNA &\xrightarrow{.924} \phi \\
mRNA &\xrightarrow{12.6} mRNA + Protein \\
Protein &\xrightarrow{1.98} \phi \, .
\end{aligned} \tag{4.11}
$$

$$
\begin{aligned}
DNA_{off_1} \longrightarrow \ldots \longrightarrow DNA_{off_{10}} &\xrightarrow{k_{on},\tau} DNA_{on} \\
DNA_{on} &\xrightarrow{k_{off}} DNA_{off_1} \\
DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\
mRNA &\xrightarrow{.924} \phi \\
mRNA &\xrightarrow{12.6} mRNA + Protein \\
Protein &\xrightarrow{1.98} \phi \, .
\end{aligned} \tag{4.12}
$$

$$
\begin{aligned}
DNA_{off_1} \longrightarrow \ldots \longrightarrow DNA_{off_{15}} &\xrightarrow{k_{on},\tau} DNA_{on} \\
DNA_{on} &\xrightarrow{k_{off}} DNA_{off_1} \\
DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\
mRNA &\xrightarrow{.924} \phi \\
mRNA &\xrightarrow{12.6} mRNA + Protein \\
Protein &\xrightarrow{1.98} \phi \ .
\end{aligned}
\tag{4.13}
$$

The unknown kinetic parameters of the models 4.7, 4.8, 4.9, 4.10, 4.11, 4.12, 4.13 are $\tau, k_{on}, k_{off}, k_m$. These models includes the mRNA degradation, protein translation and degradation reactions with .924 (Suter et al., 2011), 12.6 (Molina et al., 2013), 1.98 (Suter et al., 2011) respectively. Models 4.7, 4.8, 4.9, 4.10, 4.11, 4.12, 4.13 represents 1, 2, 3, 4, 5, 10, 15 promoter OFF states, respectively. These models include, bursting with the correct parameterization. It assumes random bursts production. The unknown parameters of the model are initialized to 1. But c_3 is initialized to 0.5. The unobserved initial promoter state and number of mRNAs are initialized to DNA_{off} and 20, respectively. The time delay value ranging from $0.5 - 5$ is selected (when present i.e. denoted as τ). Table 4.1 defines reactions for Model 4.7, 4.8, 4.9, 4.10, 4.11, 4.12, 4.13. The pictorial representation of the models 4.7 to 4.13 is depicted in Fig.4.5.

Table 4.1: Reactions defining Model 4.7-4.13.

Reaction	Rate Constant	Interpretation
$\text{DNA}_{off_1} \longrightarrow \ldots \longrightarrow DNA_{off_N} \xrightarrow{k_{on},\tau} DNA_{on}$	k_{on}	multistep promoter activation
$\text{DNA}_{on} \xrightarrow{k_{off}} DNA_{off_1}$	k_{off}	promoter inactivation
$\text{DNA}_{on} \xrightarrow{k_m} DNA_{on} + mRNA$	k_m	transcription
$\text{mRNA} \xrightarrow{.924} \phi$	.924(Suter et al., 2011)	mRNA degradation
$\text{mRNA} \xrightarrow{12.6} mRNA + Protein$	12.6 (Molina et al., 2013)	translation
$\text{Protein} \xrightarrow{1.98} \phi$	1.98 (Suter et al., 2011)	protein degradation

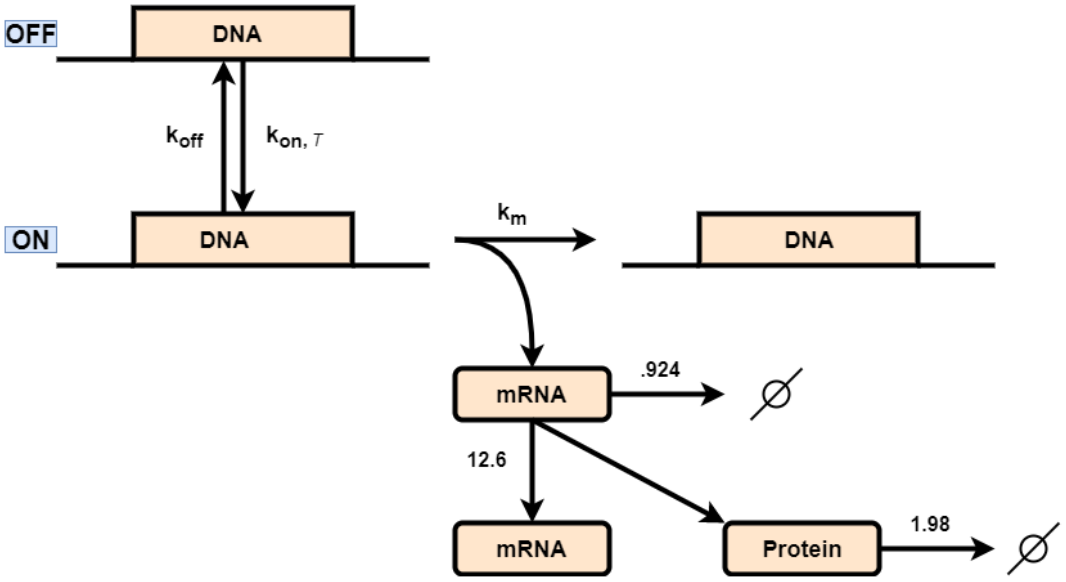


Figure 4.5: Diagrammatic representation of model description using time-series data.

Stoichiometric representation

The stoichiometric representation of the models 4.7 to 4.13 is given as follows.

$$R = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{matrix} ON \\ OFF \\ mRNA \\ protein \\ t \end{matrix} \qquad P = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{matrix} ON \\ OFF \\ mRNA \\ protein \\ t \end{matrix}$$

The matrices R and P represents the reactant and product stoichiometric coefficients for each species (row) in each reaction (column).

- the 1^{st} column of R is $\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}$ and the 1^{st} column of P is $\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \end{bmatrix}$. This shows that the 1^{st} reaction occurs as follows: $DNA_{on} \longrightarrow DNA_{off}$.

- the 2^{nd} column of R is $\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \end{bmatrix}$ and the 2^{nd} column of P is

$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}$. This shows that the 2^{nd} reaction occurs as follows: $DNA_{off} \longrightarrow DNA_{on}$.

- the 3^{rd} column of R is $\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}$ and the 3^{rd} column of P is $\begin{bmatrix} 1 & 0 & 1 & 0 & 0 \end{bmatrix}$. This shows that the 3^{rd} reaction occurs as follows: $DNA_{on} \longrightarrow DNA_{on} + mRNA$.

- the 4^{th} column of R is $\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix}$ and the 4^{th} column of P is $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix}$. This shows that the 4^{th} reaction occurs as follows: $mRNA \longrightarrow \phi$.

- the 5^{th} column of R is $\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix}$ and the 5^{th} column of P is $\begin{bmatrix} 0 & 0 & 1 & 1 & 0 \end{bmatrix}$. This shows that the 5^{th} reaction occurs as follows:$mRNA \longrightarrow mRNA + Protein$.

- the 6^{th} column of R is $\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix}$ and the 6^{th} column of P is $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix}$. This shows that the 6^{th} reaction occurs as follows: $Protein \longrightarrow \phi$.

- The 6^{th} rows of R and P are reserved for the reaction clock (t), and they should only contain zeros.

4.10 Summary

Section 4.2 presents biological motivation for multistep models. Section 4.3 describes widely used random telegraph model. Section 4.4 describes transcriptional bursting model. Section 4.5 and 4.6 details multistep promoter formulation of widely used random telegraph model and bursting model, respectively. Section 4.7 presents experimental data used in this work. Section 4.8 gives the interpretation of experimental data for multistep models. Section 4.9 gives model description for inference using glutaminase data.

Chapter 5

Delay-Bursty MCEM and Clumped-MCEM: Inference for Multistep Reaction Processes

5.1 Introduction

Many biochemical events involve multistep reactions. In order to lump multistep reactions, delays are employed (Barrio et al., 2013; Leier et al., 2014). This is, to avoid a computationally intensive task of modeling every single detail of multistep reactions, a delayed reaction is used to mimic the effects of these processes on the overall system dynamics.

A major task in computational systems biology is to conduct accurate mechanistic simulations of multistep reactions. The simulation of a biological process from experimental data requires detailed knowledge of its model structure and kinetic parameters. Despite advances in experimental techniques, the estimating unknown parameter values from observed data remains a bottleneck for obtaining accurate simulation results. Many methods exist for parameter estimation in deterministic biochemical systems; methods for discrete stochastic systems are less well developed (Wang et al., 2010). In recent years, it has become increasingly clear that, stochasticity plays a crucial role in many biological processes. For instance, intrinsic noise has been reported to have an impact on, cellular gene expression and

regulation (Cai et al., 2008); cellular differentiation (Suel et al., 2007); (ion) channel gating in neurons (White et al., 2000); pattern formation (Rudge and Burrage, 2008) and evolution (Eldar et al., 2009). As a consequence, modeling and simulation frameworks that are able to represent stochastic systems accurately have become increasingly popular.

The dynamics of a stochastic system are described by a probability distribution which cannot usually be obtained analytically (approximate methods such as finite state projection have been used with some success (Munsky and Khammash, 2010)). Instead, sampling methods like the Stochastic Simulation Algorithm (SSA) (Gillespie, 1977) are used to generate ensembles of trajectories from the unknown distribution. The SSA cannot be directly used for delayed models. Therefore, this chapter focuses on the modification of existing methods for delayed models (Barrio et al., 2006) and (Cai, 2007), which are then used for the two inference approaches that have been developed.

5.2 Maximum likelihood estimation

A natural approach for parameter estimation, given the stochastic nature of biochemical models, is to choose values that maximize the probability of the observed data with respect to the unknown parameters (Maximum Likelihood Estimates or MLEs). In the case of fully observed/complete data, where the number of molecules of each system species is known at all time points, MLEs can be calculated analytically. However, since realistic biochemical models are discretely and partially observed, computational MLE methods are needed to accurately characterize multistep promoter models and simulate their behavior.

Fig.5.1 displays workflow for modeling, parameter inference and model selection.

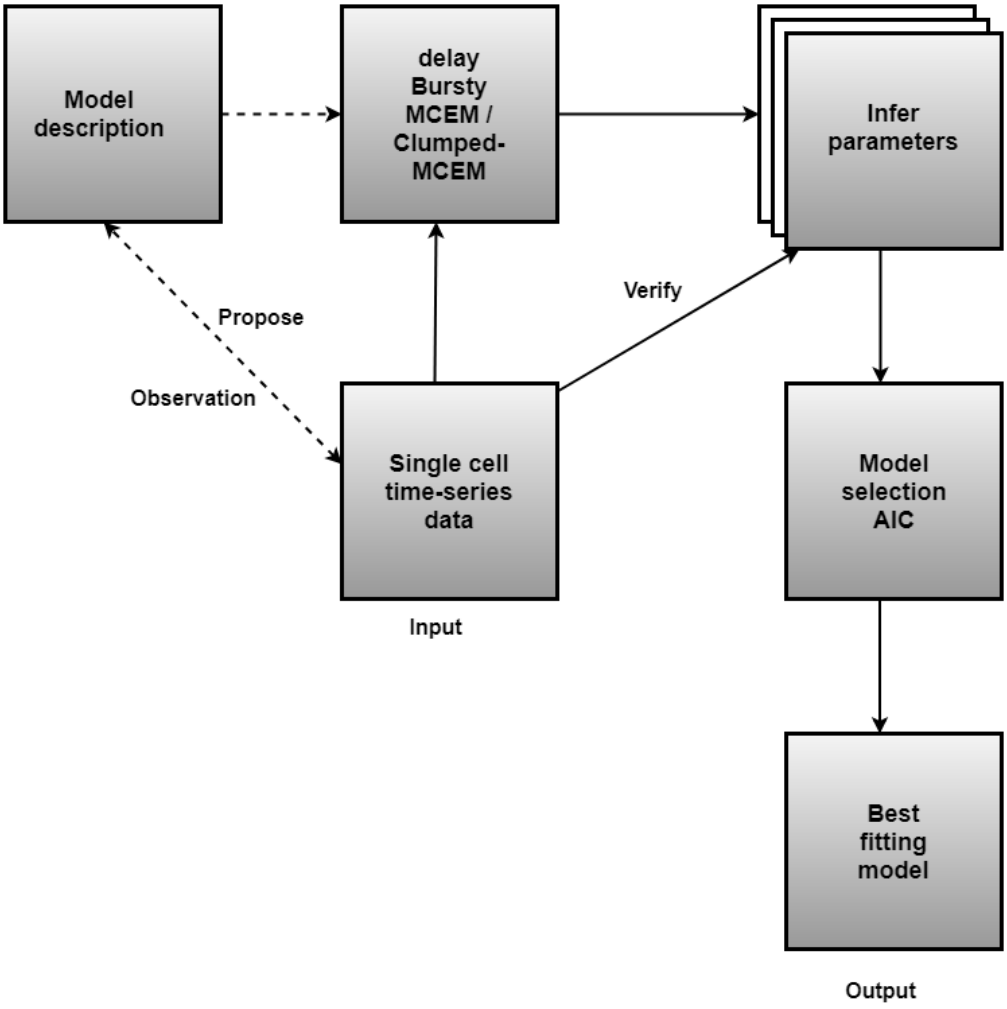


Figure 5.1: Workflow for modeling, parameter inference and model selection.

5.3 The expectation maximization algorithm : an overview

Stochatic models are commonly used to model biological data. Much of their popularity is attributed to the existence of efficient and robust procedures for learning parameters from observations. However, very often, the only data available for developing a stochatic model is incomplete. The expectation maximization algorithm enables parameter estimation in stochastic models with incomplete data. The working of expectation maximization algorithm is

demonstrated below by considering example of coin-flipping experiment.

A coin-flipping experiment

A simple coin-flipping experiment (Do and Batzoglou, 2008) in which a pair of coins A and B of unknown biases, θ_A and θ_B, respectively, are given (on any given flip, coin A may land on heads with probability θ_A and tails with probability $1 - \theta_A$, similarly, coin B may land on heads with probability θ_B and tails with probability $1 - \theta_B$). The goal is to estimate $\theta = (\theta_A, \theta_B)$ by repeating the flipping procedure five times; randomly choosing one of the two coins (with equal probability), and perform ten independent coin tosses with the selected coin.. Thus, the entire procedure involves a total of 50 coin tosses.

This experiment considers two vectors; $\mathrm{x} = (\mathrm{x}_1 \ldots \mathrm{x}_5)$ and $\mathrm{z} = (\mathrm{z}_1 \ldots \mathrm{z}_5)$, where $\mathrm{x}_i \in \{0 \ldots 10\}$ is the number of heads observed during the i^{th} set of tosses, and $\mathrm{z}_i \in \{A, B\}$ is the identity of the coin used during the i^{th} set of tosses. In this setting, parameter estimation is known as the complete data case in which the values of all relevant random variables (the result of each coin flip and the type of coin used for each flip) are known. Here, a simple way to estimate θ_A and θ_B is to calculate the observed flips of heads for each coin:

$$\widehat{\theta}_A = \frac{number\_of\_heads\_using\_coin\_A}{total\_number\_of\_flips\_using\_coin\_A} \tag{5.1}$$

$$\widehat{\theta}_B = \frac{number\_of\_heads\_using\_coin\_B}{total\_number\_of\_flips\_using\_coin\_B} \tag{5.2}$$

This intuitive guess is, in fact, known in the statistical literature as maximum likelihood estimation (the maximum likelihood method assesses the quality of a statistical model, based on the probability it assigns to the observed data). If $logP(\mathrm{x}, \mathrm{z}; \theta)$ is the logarithm of the joint probability (or log likelihood) of obtaining any particular vector of observed head counts x and coin types z, then the formulae in (5.1) solve for the parameters $\theta = (\theta_A, \theta_B)$ that maximize $logP(\mathrm{x}, \mathrm{z}; \theta)$.

Now consider a more challenging variant of the parameter estimation problem, where the recorded head counts x are given, but, the identities z of

the coins used for each set of tosses are not known; z is referred to as hidden variables or latent factors. Parameter estimation in this new setting is known as the incomplete data case. This time, calculating heads for each coin toss is no longer possible, because the coin used for each set of tosses is not known. However, if there can be some way of completing the data (in this case, guessing correctly which coin has been used in each of the five sets), then the parameter estimation for this problem with incomplete data can be reduced to maximum likelihood estimation with complete data.

One iterative scheme for obtaining completion of the incomplete data can work as follows: starting from some initial parameters, $\widehat{\theta}^{(t)} = (\widehat{\theta}_A^{(t)}, \widehat{\theta}_B^{(t)})$; determine for each of the five sets, whether coin A or coin B has more likely generated the observed flips (using the current parameter estimates). Then, assume these completions (that is, guessed coin assignments) to be correct, and apply the regular maximum likelihood estimation procedure to get $\widehat{\theta}^{(t+1)}$. Finally, repeat these two steps until convergence. As the estimated model improves, so does the quality of the resulting completions.

The Expectation Maximization algorithm is a refinement on this basic idea. Rather than picking the single most likely completion of the missing coin assignments on each iteration, the Expectation Maximization algorithm computes probabilities for each possible completion of the missing data, using the current parameters $\widehat{\theta}^{(t)}$. These probabilities are used to create a weighted training set consisting of all possible completions of the data. Finally, a modified version of maximum likelihood estimation, that deals with weighted training examples, provides new parameter estimates, $\widehat{\theta}^{(t+1)}$. By using weighted training examples rather than choosing the single best completion, the Expectation Maximization algorithm accounts for the confidence of the model in each completion of the data.

In summary, the Expectation Maximization algorithm alternates between the steps of guessing a probability distribution over completions of missing data, given the current model (the E-step) and then re-estimating the model parameters using these completions (the M-step). The name *E-step* comes from the fact that one does not usually need to explicitly form the probability

distribution over completions, rather the need is to only compute *expected* sufficient statistics over these completions. Similarly, the name *M-step* comes from the fact that model re-estimation can be thought of as *maximization* of the expected log-likelihood of the data.

5.4 Simulation

This section modifies existing simulation algorithms to simplify multistep reactions. The modified algorithms are presented in following subsections namely, Delay Stochastic Simulation Algorithm (DSSA) and Modified Cai's Exact SSA Method (MCEM).

5.4.1 Delay Stochastic Simulation Algorithm for multistep reactions

The Delay Stochastic Simulation Algorithm (DSSA) is the extension of SSA for simulating models with delays. The DSSA differs from the SSA by making a clear distinction between the reaction type and reaction delay. First, it divides reactions with delays into two groups (1) Consuming delayed reactions; (2) Nonconsuming delayed reactions. When a consuming reaction occurs, the numbers of reactant molecules are updated at the time of initiation, while numbers of product molecules are updated at the end of the time delay. When a nonconsuming reaction occurs, the numbers of reactants and products are updated only at completion. The choice of reaction type for modeling of biochemical reactions depends on the biological context. For instance, a single gene is transcribed simultaneously (by several RNA polymerases) and the DNA itself is not consumed by the first transcription. Thus, one can assume that, the transcription process is a nonconsuming reaction. The reaction delay is the time from the initiation to completion (i.e. processing of the reactants to the appearance of the products). In this work, DSSA version of nondelayed and delayed nonconsuming reactions is considered. As is mentioned, nondelayed and delayed nonconsuming reactions have only one update point for updating

both numbers of reactants and products molecules. The former when the nondelayed reaction happens, the latter when the delay ends.

To model multistep promoter OFF states with bursting, as shown in Model 4.4, first, the delay constant (τ in the model 4.4) associated with each reaction is specified. If the nondelayed reaction is chosen, then the state is updated as in SSA (Algorithm 1). But, if delayed reaction is selected, it is not updated until prescribed time delay.

As mentioned in Model 4.4, the distribution of time spent in multistep promoter approaches an Erlang distribution. Using this distribution, it is formulated in such a way that for a burst arrival with delay τ_j, the current burst arrival should depend on the historical state at time $t - \tau_j$. It can be interpreted as the probability that a burst occurred in $[t - \tau_j, t - \tau_j + dt)$ that is to be updated in $[t, t + dt)$. The method for the implementation of this algorithm is given in Algorithm 2.

Algorithm 2 Delay Stochastic Simulation algorithm

Input: a model of M reactions in which each reaction R_j, $j = 1 \dots M$ and propensity a_j, the initial state x_0 at time 0.
Output: trajectories of the model.

1. Set time $t = 0$ with state $X(t) = \mathrm{x}_0$
2. Calculate propensity functions $a_j(\mathrm{x}), j = 1 \dots M, a_0(\mathrm{x}) = \sum_{j=1}^{M} a_j(\mathrm{x})$
3. Generate random numbers $r_1, r_2 \sim U(0, 1)$
   compute $\tau = \frac{1}{a_0} ln \left(\frac{1}{r_1} \right)$
4. If there are nonconsuming delayed reaction to finish in the time interval $[t, t + \tau)$, update time $t \leftarrow \tau$, where τ is the time when the first delayed reaction finishes. Update reactants and products. Repeat step 2 and 3. If there is no delayed reaction to finish in $[t, t + \tau)$, proceed to step 5.
5. Generate j from a random number r_2
   $\sum_{j'=1}^{j-1} a_{j'}(t) < r_2 a_0(t) \leq \sum_{j'=1}^{j} a_{j'}(t)$
   If R_j is nondelayed reaction, update reactants and products.

6. set $t \leftarrow t + \tau$, go to step 2 or stop.

5.4.2 Modified Cai's Exact SSA Method

Cai's method (Cai, 2007) involves two computationally intensive steps in computing the firing time τ. To begin with, it has to consider each element of the delayed event queue *Tstruct*, to evaluate the cdf F of the firing time τ. Following this, the relative completion times of delayed events, in the delayed event queue *Tstruct*, need to be updated by the simulation.

This work proposes the Modified Cai's Exact SSA Method (MCEM) to simplify multistep reactions. The idea for processing delays for Model 4.4 is as follows. Let R be a nonconsuming delayed reaction with the delay τ_d, assuming that R is initiated at time $t + \tau$. First, an event is created with time $t + \tau + \tau_d$ and stored for later processing. Then the simulation continues processing until time $t+\tau+\tau_d$. At this point, the delay reaction R is retrieved to update the state. This work is based on the fact that none of the delayed reactions are scheduled to complete before the specified time that follows an exponential distribution. Considering that the delayed event occurs after time $t + \tau$, the firing time τ does not change (Thanh et al., 2017). Thus, it is safe to select the next reaction R_j, with probability $\frac{a_j}{a_0}$, to initiate at time $t + \tau$. The Modified Cai's Exact SSA Method (MCEM) considers absolute completion times of delayed reactions instead of relative time. The method for the implementation of this algorithm is given in Algorithm 3.

Algorithm 3 Modified Cai's Exact SSA Method (MCEM)

Input: a model of M reactions in which each reaction R_j, $j = 1 \dots M$ and propensity a_j, the initial state x_0 at time 0.
Output: trajectories of the model.

1. Set time $t = 0$ with state $X(t) = \mathrm{x}_0$
2. Calculate propensity functions $a_j(\mathrm{x}), j = 1 \dots M, a_0(\mathrm{x}) = \sum_{j=1}^{M} a_j(\mathrm{x})$
3. Generate random numbers $r_1, r_2 \sim U(0, 1)$

compute $\tau = \frac{1}{a_0} ln\left(\frac{1}{r_1}\right)$
set $t_{next} = t + \tau$

4. IF $\tau_d > t_{next}$ let τ_d be the completion time of delayed reaction R_j
   select reaction firing R_j with probability $\frac{a_j}{a_0}$ by finding the smallest reaction index j' such that $\sum_{j=1}^{j'} a_j \geq r_2 a_0$
   Nonconsuming delayed reaction: update reactants and products after delay τ_d
   ELSE Nondelayed reaction: update state by reactants and products of R_j

5. set $t = t_{next}$

6. Go to step 2 or stop.

5.5 Discrete-state stochastic reaction kinetics

Consider, discrete, stochastic chemical kinetic models that assume a well-mixed reactor volume, consisting of N molecular species, denoted by S_i for $i = 1 \ldots N$. The system state is represented by the N-dimensional random process $X(t) = (X_1(t) \ldots X_N(t))$ at time t, where $X_i(t)$ denotes the number of molecular species S_i at time t. The firing of M reactions $R_1 \ldots R_M$, evolve through the discrete-valued molecular population numbers. $a_j(\mathrm{x}(t))dt$ $(j = 1 \ldots M)$ gives the probability that, reaction R_j fires in the next infinitesimal time interval $[t, t + dt)$, given $X(t) = \mathrm{x}$ with sum $a_0(\mathrm{x}(t))$. This work focuses on reactions that follow mass action kinetics - i.e. where $a_j(\mathrm{x}(t)) = \theta_j h_j(\mathrm{x}(t))$. Where θ_j, a kinetic rate constant and $h_j(\mathrm{x}(t))$, a function that quantifies the number of possible ways reaction R_j, can occur, given system state x. In this work, for the delayed nonconsuming reactions, $h_j(\mathrm{x}(t))$ is set to 1.

5.6 Simulation for reactions with delays

The implementation of the Algorithm 2 and 3 provides a numerical procedure for generating system trajectories of the molecular populations from their underlying distribution. It works by selecting the time to the next reaction (τ) and the index of the next reaction (j') as exponential with mean $1/a_0(\mathrm{x})$ and categorical with probabilities $a_j(\mathrm{x})/a_0(\mathrm{x})$ ($j = 1, \ldots, M$) random variables, respectively. Given, x_0 and final time T, application of the Algorithm 2 and 3 yields a trajectory $z \equiv (\tau_1, j_1', \ldots, \tau_r, j_r')$, Where r is the number of times the j^{th} reaction fires. The likelihood of the complete system trajectory (x_0, z), as the function of kinetic parameters θ, is given by,

$$f_\theta(\mathrm{x}_0, z) = \big(\prod_{i=1}^{r} \theta_{j_i'} h_{j_i'}(\mathrm{x}_{i-1})\big) \times exp\big(-\sum_{i=1}^{r+1}[\tau_i \sum_{j=1}^{M} \theta_j h_j(\mathrm{x}_{i-1})]\big)\ . \tag{5.3}$$

5.7 Parameter inference using maximum likelihood approach

Single-cell time-series data is incomplete as it provides the number of molecules for a species at d discrete time instances. The observed data is represented as $y \equiv (\mathrm{x}_0, \mathrm{x}_1', \ldots, \mathrm{x}_d')$, where x_i' denotes the numbers of molecules of a subset of the N species, at some time point t_i. The Expectation Maximization (EM) (Dempster et al., 1977), given an incomplete data, is an algorithm to calculate maximum likelihood. Given $\hat{\theta}^{(0)}$, this algorithm is based on iterative computation (Robert and Casella, 2004):

$$\hat{\theta}^{(n+1)} = arg_\theta max \left(\mathbb{E}\left[log f_\theta(\mathrm{x}_0, z)|y, \hat{\theta}^{(n)}\right]\right) \tag{5.4}$$

$$\hat{\theta}^{(n+1)} = arg_\theta max \left(\sum_{z \in Z(y)} \left[g(z|y, \hat{\theta}^{(n)}) \times log f_\theta(\mathrm{x}_0, z)\right]\right) \tag{5.5}$$

where $\mathbb{E}\left[\cdot|y, \hat{\theta}^{(n)}\right]$ is the expectation operator w.r.to the conditional distribution of z given y and $\theta^{(n)}$. $Z(y)$ is the set of all valid trajectories that are consistent with y. $g\left[z|y, \hat{\theta}^{(n)}\right]$ denotes the unknown conditional density of z.

An explicit evaluation of the summation is intractable in Equation 5.5, instead, the following methods are used to generate reaction trajectories:
Method 1: This method combines Monte Carlo approach of Expectation Maximization (MCEM) and Algorithm 2 and is termed as Delay-Bursty MCEM.
Method 2: This method combines Monte Carlo approach of Expectation Maximization (MCEM) and Algorithm 3 and is termed as Clumped-MCEM.
Method 1 and **Method 2** generates reaction trajectories to approximate $\hat{\theta}^{(n+1)}$:

$$\hat{\theta}^{(n+1)} \approx arg_{\theta}max\Big(\sum_{k=1}^{K}\left[I\left(z_k^n \in Z(y)\right) \times log f_{\theta}\left(\mathrm{x}_0, z_k^n\right)\right]\Big) \tag{5.6}$$

$$\hat{\theta}^{(n+1)} = arg_{\theta}max\Big(\sum_{k'=1}^{K'} log(f_{\theta}(\mathrm{x}_0, z_{k'}^n))\Big) \tag{5.7}$$

where z_k^n is the k^{th} Delay Stochastic Simulation Algorithm (DSSA) (Algorithm 2) or Modified Cai's Exact SSA Method (MCEM) (Algorithm 3) trajectory, simulated using the parameter vector $\hat{\theta}^n$. $I(z_k^n \in Z(y))$ is an indicator function taking a value of 1 if z_k^n is consistent with y (otherwise 0). K is the total number of simulated trajectories. $k^{'}$ indexes only the $K^{'}$ simulated trajectories that are consistent with the observed data (Equation 5.7). K is set to the value that leads to the number of consistent trajectories $K^{'}$. Simplifying Equation 5.7 as in (Wilkinson, 2006), the maximum likelihood estimates for each reaction is given by

$$\hat{\theta_j}^{(n+1)} = \frac{\sum_{k'=1}^{K'} r_{jk'}^n}{\sum_{k'=1}^{K'}\Big(\sum_{i=1}^{r_{k'}^n+1} a_{jk}^{in} \times \tau_{ik'}^n\Big)} \; . \tag{5.8}$$

Equation 5.8 can be rewritten as,

$$\hat{\theta_j}^{(1)} = \hat{\theta_j}^{(0)} \times \frac{\sum_{k'=1}^{K'} r_{jk'}^n}{\sum_{k'=1}^{K'}\Big(\sum_{i=1}^{r_{k'}^n+1} a_{jk}^{in} \times \tau_{ik'}^n\Big)} \tag{5.9}$$

where $\hat{\theta_j}^{(0)}$ and $\hat{\theta_j}^{(1)}$ indicates the initial guess and first update respectively, for parameter θ_j. i indexes the start of the simulation and $r_k{'}$ is the total number of reactions firing, arriving at the final time $r_k{'}+1$. $r_{jk'}$ is the number of times the j^{th} reaction fires. a_{jk}^i is the value of the propensity function for the

j^{th} reaction, immediately after the i^{th} event; $\tau_{ik'}$ is the time interval between the events.

Method 1: The Delay-Bursty MCEM uses ascent-based MCEM (Caffo et al., 2005) to select number of consistent trajectories $K^{'}$ and iterations n. The trajectories of this method are generated using Algorithm 2. The MCEM version of the maximum likelihood estimates for each reaction is given in Equation 5.9.

Method 2: The Clumped-MCEM involves two phases, as in Bursty $MCEM^2$. When an initial guess is given for the unknown parameters $(\hat{\theta_j}^{(0)})$, the Cross Entropy (CE) method begins by simulating K trajectories using Algorithm 3. The $K' = \rho \times K$ trajectories, that are closest to a given observed data, are selected based on the computation of distance (Daigle et al., 2015) from each trajectory to the observed data. It leads to compute better parameter estimates for $\hat{\theta_j}^{(1)}$. This process is repeated until final time. Upon reaching the final time, the CE phase computes the update shown in Equation 5.9. The CE phase parameter estimates are used as input parameters to MCEM phase. This phase simulates trajectories using Algorithm 3. Upon reaching the final time, the MCEM phase computes the update shown in Equation 5.9 (with K' replaced by K''). Table 5.1 summarizes the phases of Delay-Bursty MCEM and Clumped-MCEM.

Table 5.1: The phases of Delay-Bursty MCEM and Clumped-MCEM simulation.

Methods	CE phase	MCEM phase
Delay-Bursty MCEM	-	Delay Stochastic Simulation Algorithm (DSSA)
Clumped-MCEM	Modified Cai's Exact SSA Method (MCEM)	Modified Cai's Exact SSA Method (MCEM)

5.8 Results

In this section, empirical results are presented to support two main claims:

1. Models with multiple OFF states produce behaviour which is most consistent with experimental data.

2. Delay-Bursty MCEM and Clumped-MCEM inference is more efficient for time-series data.

5.8.1 Accuracy of the model

In this work, Akaike Information Criterion (AIC) (Akaike, 1974) is used to compare the complexity of different models. It gives lower value for models which best fit observed experimental data. It is given by

$$AIC = 2m - 2log(\hat{L}) \ . \tag{5.10}$$

where m denotes the number of unknown parameters from the model.

The goal is to decide on parameter values(τ and Number Of States), such that, AIC is as low as possible and k_m/k_{off} is as close as possible to the number of mRNAs 20(i.e. observed value column in this experiments).

Tables 5.2 to 5.8 and Tables 5.9 to 5.15 show maximum likelihood parameter estimates for model parameters k_{on}, k_{off}, k_m. Tables 5.2 to 5.8 and Tables 5.9 to 5.15 show k_m/k_{off} and AIC score for various values of τ and Number Of States, using experimentally observed values in the glutaminase dataset for Delay-Bursty MCEM and Clumped-MCEM respectively. From these rows of Tables 5.2 to 5.8 and Tables 5.9 to 5.15, it is seen that, setting $\tau = 4.75$ and Number Of States = 15 give the best fit when compared to experimentally observed time-series data (with mRNAs = 20) with the lowest value of AIC. These results further support the hypothesis, that a model with the larger number of OFF states (15 in our case) is able to better explain the experimentally observed values during transcriptional bursting.

To further evaluate the robustness of these inference techniques, data from a model is generated, and used to carry out parameter inference, to determine if the inferred parameters agree with the original values used during data generation. These details are presented in Appendix A.2. Results in highlighted text describe the models that best fits the data. Simulation results, using synthetic and glutaminase promoter data, show that (i) bursting kinetics are promoter specific (i.e. it depends on the number

of promoter OFF states) and (ii) mRNA production is consistent with multiple promoter OFF states.

Table 5.2: Delay-Bursty MCEM parameter inference using glutaminase promoter time-series data for Model 4.7.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC
1	0.5	2.97	3.57	66.02	18.49	20	3478.17
1	0.75	3.44	4.08	71.18	17.44	20	3471.10
1	1.10	4.27	3.79	72.24	19.06	20	3467.85
1	1.40	4.32	3.54	73.13	20.65	20	3470.18
1	1.65	6.84	3.89	73.82	18.97	20	3468.89
1	2.10	8.80	3.56	68.94	19.36	20	3469.89
1	3.40	18.77	3.13	67.21	21.47	20	3468.24
1	4.00	19.79	3.23	68.75	21.28	20	3469.25
1	4.10	26.20	3.28	69.09	21.06	20	3468.55
1	**4.75**	**45.33**	**3.50**	**65.60**	**18.74**	**20**	**3467.55**
1	5.00	50.70	2.96	63.54	21.46	20	3469.31

Table 5.3: Delay-Bursty MCEM parameter inference using glutaminase promoter time-series data for Model 4.8.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC
2	0.5	3.13	3.73	66.06	17.71	20	3476.68
2	0.75	3.22	3.60	70.91	19.69	20	3473.79
2	1.10	4.43	3.87	72.62	18.76	20	3470.25
2	1.40	4.89	3.64	72.89	20.02	20	3465.37
2	1.65	6.58	3.89	74.95	19.26	20	3468.70
2	2.10	8.43	3.64	70.71	19.42	20	3467.16
2	3.40	18.79	3.55	68.62	19.32	20	3463.77
2	4.00	22.34	3.28	67.65	20.62	20	3466.19
2	4.10	25.79	3.48	71.08	20.42	20	3464.17
2	**4.75**	**48.37**	**3.40**	**64.36**	**18.92**	**20**	**3462.45**
2	5.00	45.30	3.12	65.53	21.00	20	3463.45

Table 5.4: Delay-Bursty MCEM parameter inference using glutaminase promoter time-series data for Model 4.9.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC
3	0.5	3.00	3.63	66.58	18.34	20	3476.51
3	0.75	3.23	3.44	70.46	20.48	20	3473.48
3	1.10	4.36	3.71	72.00	19.40	20	3470.32
3	1.40	4.98	3.62	73.30	20.24	20	3467.78
3	1.65	6.53	3.74	73.70	19.70	20	3469.18
3	2.10	7.56	3.45	70.52	20.44	20	3465.09
3	3.40	19.43	3.48	67.95	19.52	20	3467.50
3	4.00	24.18	3.26	67.62	20.74	20	3465.99
3	4.10	24.85	3.44	71.70	20.84	20	3465.65
3	**4.75**	**47.50**	**3.17**	**67.30**	**21.23**	**20**	**3463.78**
3	5.00	41.69	3.30	65.38	19.81	20	3465.42

Table 5.5: Delay-Bursty MCEM parameter inference using glutaminase promoter time-series data for Model 4.10.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC
4	0.5	2.93	3.69	66.55	18.03	20	3475.98
4	0.75	3.42	3.67	69.33	18.89	20	3474.17
4	1.10	4.27	3.78	73.13	19.34	20	3470.04
4	1.40	5.27	3.65	73.20	20.05	20	3467.86
4	1.65	6.88	3.97	74.11	18.66	20	3468.22
4	2.10	6.90	3.30	70.40	21.33	20	3465.86
4	3.40	18.53	3.39	68.51	20.20	20	3464.76
4	4.00	25.93	3.22	67.32	20.90	20	3463.94
4	4.10	24.61	3.45	70.84	20.53	20	3465.4
4	**4.75**	**44.46**	**3.20**	**67.49**	**21.09**	**20**	**3462.74**
4	5.00	44.09	3.23	65.51	20.28	20	3469.26

Table 5.6: Delay-Bursty MCEM parameter inference using glutaminase promoter time-series data for Model 4.11.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC
5	0.5	3.00	3.81	67.04	17.59	20	3476.03
5	0.75	3.09	3.55	69.46	19.56	20	3475.37
5	1.10	4.38	3.85	73.13	18.99	20	3469.86
5	1.40	5.10	3.75	72.33	19.28	20	3468.99
5	1.65	6.70	3.87	72.91	18.83	20	3468.71
5	2.10	7.22	3.29	69.00	20.97	20	3468.97
5	3.40	18.86	3.32	67.64	20.37	20	3464.22
5	4.00	26.24	3.20	65.66	20.51	20	3464.53
5	4.10	25.87	3.51	70.77	20.16	20	3465.15
5	**4.75**	**46.20**	**3.43**	**66.06**	**19.25**	**20**	**3463.14**
5	5.00	40.66	3.11	67.41	21.67	20	3464.64

Table 5.7: Delay-Bursty MCEM parameter inference using glutaminase promoter time-series data for Model 4.12.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC
10	0.5	3.03	4.03	70.86	17.58	20	3477.00
10	0.75	3.52	3.84	70.59	18.38	20	3472.88
10	1.10	4.45	3.89	73.12	18.79	20	3471.10
10	1.40	5.32	3.76	72.88	19.38	20	3469.35
10	1.65	6.26	3.68	72.45	19.68	20	3467.27
10	2.10	8.45	3.39	69.04	20.36	20	3469.83
10	3.40	19.40	3.47	69.17	19.93	20	3465.22
10	4.00	25.55	3.45	69.99	20.28	20	3464.87
10	4.10	27.82	3.14	67.16	21.38	20	3465.08
10	**4.75**	**42.57**	**3.19**	**65.60**	**20.56**	**20**	**3463.33**
10	5.00	45.59	3.15	66.10	20.98	20	3464.39

Table 5.8: Delay-Bursty MCEM parameter inference using glutaminase promoter time-series data for Model 4.13.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC
15	0.5	3.02	4.04	70.67	17.49	20	3476.14
15	0.75	3.49	3.84	70.92	18.46	20	3473.14
15	1.10	4.29	3.79	72.96	19.25	20	3469.84
15	1.40	5.25	3.71	72.57	19.56	20	3468.60
15	1.65	6.13	3.71	72.89	19.64	20	3467.4
15	2.10	8.48	3.55	70.51	19.86	20	3467.18
15	3.40	19.58	3.46	69.27	20.02	20	3464.02
15	4.00	27.58	3.46	70.01	20.23	20	3463.78
15	4.10	31.01	3.43	68.88	20.08	20	3465.34
15	**4.75**	**41.64**	**3.15**	**63.36**	**20.11**	**20**	**3461.25**
15	5.00	45.25	3.29	67.68	20.57	20	3464.53

Table 5.9: Clumped-MCEM parameter inference using glutaminase promoter time-series data for Model 4.7.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC
1	0.5	3.00	3.77	69.03	18.31	20	3478.16
1	0.75	3.38	4.11	74.32	18.08	20	3476.71
1	1.10	3.76	3.68	76.03	20.66	20	3469.46
1	1.40	5.16	3.76	73.41	19.52	20	3472.05
1	1.65	7.60	4.43	74.76	16.87	20	3470.29
1	2.10	10.06	3.84	69.37	18.06	20	3467.80
1	3.40	22.61	3.31	67.41	20.36	20	3467.89
1	4.00	26.20	2.97	62.70	21.11	20	3471.95
1	4.10	23.44	3.38	69.44	20.54	20	3468.16
1	**4.75**	**49.22**	**3.35**	**67.50**	**20.14**	**20**	**3464.20**
1	5.00	44.51	3.40	69.64	20.48	20	3464.79

Table 5.10: Clumped-MCEM parameter inference using glutaminase promoter time-series data for Model 4.8.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC
2	0.5	2.89	3.95	70.85	17.93	20	3474.89
2	0.75	3.26	3.80	75.22	19.79	20	3478.29
2	1.10	4.09	3.92	75.20	19.18	20	3472.57
2	1.40	5.36	3.84	75.01	19.53	20	3467.90
2	1.65	7.67	4.32	76.06	17.60	20	3470.97
2	2.10	10.01	3.50	69.38	19.82	20	3467.70
2	3.40	21.79	3.41	68.69	20.14	20	3464.82
2	4.00	29.96	3.03	62.88	20.75	20	3464.69
2	4.10	26.54	3.53	70.49	19.96	20	3466.18
2	**4.75**	**44.50**	**3.22**	**67.25**	**20.88**	**20**	**3461.93**
2	5.00	40.23	3.32	70.21	21.14	20	3464.84

Table 5.11: Clumped-MCEM parameter inference using glutaminase promoter time-series data for Model 4.9.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC
3	0.5	2.91	3.83	69.49	18.14	20	3476.91
3	0.75	3.29	4.07	73.79	18.13	20	3475.89
3	1.10	4.14	3.89	74.22	19.07	20	3468.52
3	1.40	5.25	3.75	73.65	19.64	20	3465.47
3	1.65	7.88	4.45	77.67	17.45	20	3467.19
3	2.10	10.17	3.85	70.40	18.28	20	3466.82
3	3.40	20.07	3.33	68.45	20.55	20	3462.85
3	4.00	34.38	3.09	62.63	20.26	20	3467.19
3	4.10	29.98	3.59	70.94	19.76	20	3464.23
3	**4.75**	**41.87**	**3.28**	**66.98**	**20.42**	**20**	**3462.38**
3	5.00	42.19	3.54	71.86	20.29	20	3466.86

Table 5.12: Clumped-MCEM parameter inference using glutaminase promoter time-series data for Model 4.10.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC
4	0.5	2.78	3.95	71.39	18.07	20	3477.91
4	0.75	3.50	3.98	72.01	18.09	20	3472.68
4	1.10	4.28	3.57	71.21	19.94	20	3474.63
4	1.40	5.14	3.78	75.41	19.94	20	3470.73
4	1.65	7.11	4.21	77.58	18.42	20	3469.49
4	2.10	8.64	3.44	69.63	20.24	20	3465.24
4	3.40	21.15	3.47	68.59	19.76	20	3466.23
4	4.00	30.94	3.05	61.36	20.11	20	3465.18
4	4.10	27.83	3.65	72.38	19.83	20	3463.49
4	**4.75**	**43.33**	**3.18**	**67.45**	**21.21**	**20**	**3463.40**
4	5.00	40.40	3.60	69.79	19.38	20	3467.09

Table 5.13: Clumped-MCEM parameter inference using glutaminase promoter time-series data for Model 4.11.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC
5	0.5	3.01	4.21	72.52	17.22	20	3479.93
5	0.75	3.63	4.00	70.36	17.59	20	3475.69
5	1.10	4.42	3.90	70.65	18.11	20	3470.04
5	1.40	4.88	3.85	75.56	19.62	20	3467.75
5	1.65	6.76	4.33	77.43	17.88	20	3469.57
5	2.10	9.46	3.67	70.31	19.15	20	3466.92
5	3.40	20.09	3.40	69.19	20.35	20	3465.45
5	4.00	32.06	2.92	62.83	21.51	20	3467.89
5	4.10	28.15	3.88	74.15	19.11	20	3464.93
5	**4.75**	**42.01**	**3.35**	**67.39**	**20.11**	**20**	**3463.33**
5	5.00	43.92	3.64	71.09	19.53	20	3464.54

Table 5.14: Clumped-MCEM parameter inference using glutaminase promoter time-series data for Model 4.12.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC
10	0.5	3.02	4.18	72.65	17.38	20	3476.45
10	0.75	3.54	3.88	70.84	18.25	20	3473.72
10	1.10	4.36	3.86	73.57	19.05	20	3471.29
10	1.40	4.79	3.93	77.28	19.66	20	3468.89
10	1.65	6.42	3.93	75.05	19.09	20	3468.40
10	2.10	9.48	3.53	69.68	19.73	20	3466.36
10	3.40	19.89	3.46	68.77	19.87	20	3464.57
10	4.00	30.66	3.07	63.90	20.81	20	3464.47
10	4.10	28.10	3.56	71.79	20.16	20	3464.55
10	**4.75**	**39.14**	**3.27**	**66.73**	**20.40**	**20**	**3463.56**
10	5.00	35.08	3.29	68.11	20.70	20	3465.42

Table 5.15: Clumped-MCEM parameter inference using glutaminase promoter time-series data for Model 4.13.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC
15	0.5	2.96	4.14	72.70	17.56	20	3475.75
15	0.75	3.45	3.82	71.35	18.67	20	3472.87
15	1.10	4.50	3.79	72.05	19.01	20	3471.24
15	1.40	5.25	3.84	75.16	19.57	20	3469.21
15	1.65	6.31	3.86	74.22	19.22	20	3467.42
15	2.10	8.70	3.54	69.74	19.70	20	3466.53
15	3.40	20.08	3.34	67.92	20.33	20	3463.80
15	4.00	29.72	3.14	64.73	20.61	20	3463.70
15	4.10	27.51	3.51	70.56	20.10	20	3463.88
15	**4.75**	**35.76**	**3.26**	**67.62**	**20.74**	**20**	**3460.27**
15	5.00	44.92	3.22	67.25	20.88	20	3464.48

5.8.2 Comparison of random telegraph model with multistep model

$$
\begin{aligned}
DNA_{off} &\underset{k_{off}}{\overset{k_{on}}{\rightleftharpoons}} DNA_{on} \\
DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\
mRNA &\xrightarrow{.924} \phi \\
mRNA &\xrightarrow{12.6} mRNA + Protein \\
Protein &\xrightarrow{1.98} \phi\ .
\end{aligned}
\tag{5.11}
$$

Table 5.16 show inferred values k_{on}, k_{off}, k_m and AIC values for random telegraph model. When τ is set to 0, it reduces to random telegraph model. Model 5.11 also includes the mRNA degradation, protein translation and degradation reactions with .924 (Suter et al., 2011), 12.6 (Molina et al., 2013), 1.98 (Suter et al., 2011) respectively. It assumes bursting with the correct parameterization. From these rows of Table 5.16, it is seen that the random telegraph model fails to produce bursts consistent with data (mRNAs = 20 in our case). These results further strengthens the conclusions drawn from multistep promoter models.

Table 5.16: Parameter inference values for random telegraph model using glutaminase promoter time-series data for Model 5.11.

Method	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC
Delay-Bursty MCEM	0	1.95	4.36	69.35	15.90	20	3484.46
Clumped-MCEM	0	2.12	5.18	73.89	14.26	20	3482.94

5.8.3 Scaling of inference approach with model complexity

$$
\hat{\theta}^{(n+1)} = arg_{\theta} max \left(\sum_{z \in Z(y)} \left[g(z|y, \hat{\theta}^{(n)}) \times log f_{\theta}(\mathrm{x}_0, z) \right] \right) \tag{5.12}
$$

The theory behind the EM (Dempster et al., 1977) algorithm guarantees that Equation 5.12 will converge to estimates that locally maximize the observed data likelihood, given sufficiently large number of iterations. An explicit

evaluation of the summation is intractable in Equation 5.12, instead, we use Monte Carlo extension of EM that generate reaction trajectories using simulation algorithms. The effective use of these inference approaches involves appropriate selection of the consistent trajectories and iterations. Such selections are done by heuristics that are dependent on the model being analyzed.

5.8.4 Comparison with the literature

The results presented in Tables 5.2 to 5.8 can be produced using Bursty $MCEM^2$ (Daigle et al., 2015) and Delay-Bursty MCEM. The results produced using Clumped-MCEM is presented in Tables 5.9 to 5.15. In this section, the comparison of these approaches is made in terms of efficiency. Numerical experiments have ran on 8 core Intel Xeon CPU ($E52650@2.6GHz$) with 64GB memory. Tables 5.17 and 5.19 show an initial number of trajectories simulated for each method using glutaminase promoter time-series data. The total simulation time for Delay-Bursty MCEM and Clumped-MCEM using glutaminase data is given in Table 5.18 and 5.20. The total simulation time in Table 5.18 shows that the Bursty $MCEM^2$ takes $\approx$ 8 days to obtain an estimate of the similar numerical accuracy, as the Delay-Bursty MCEM estimate takes $\approx$ 5 days. The computational cost of Delay-Bursty MCEM is reduced by 37.44% as compared to Bursty $MCEM^2$. The total simulation time for Clumped-MCEM in Table 5.20 shows that it takes $\approx$ 3.5 days to obtain similar accuracy as Delay-Bursty MCEM and Bursty $MCEM^2$. The computational cost of Clumped-MCEM is reduced by 57.58% as compared to Bursty $MCEM^2$. Further, the computational cost of Clumped-MCEM is reduced by 32.19% as compared to Delay-Bursty MCEM.

Fig.5.2 compares the execution times of Clumped-MCEM, Delay-Bursty MCEM and Bursty $MCEM^2$ in simulating the multistep promoter model, using glutaminase promoter time-series data respectively.

Table 5.17: Initial number of trajectories simulated for the glutaminase data.

Method	No. of trajectories in CE phase	No. of trajectories in MCEM phase
Delay-Bursty MCEM	-	2500
Bursty MCEM^2	10000	2500

Table 5.18: Execution times for the multistep promoter model using glutaminase data.

Number Of States	Method	CPU(h)
15	Delay-Bursty MCEM	120.34
15	Bursty MCEM^2	192.36

Table 5.19: Initial number of trajectories simulated for the glutaminase data.

Method	No. of trajectories in CE phase	No. of trajectories in MCEM phase
Clumped-MCEM	10000	1500
Bursty MCEM^2	10000	2500

Table 5.20: Execution times for the multistep promoter model using glutaminase data.

Number Of States	Method	CPU(h)
15	Clumped-MCEM	81.6
15	Bursty MCEM^2	192.36

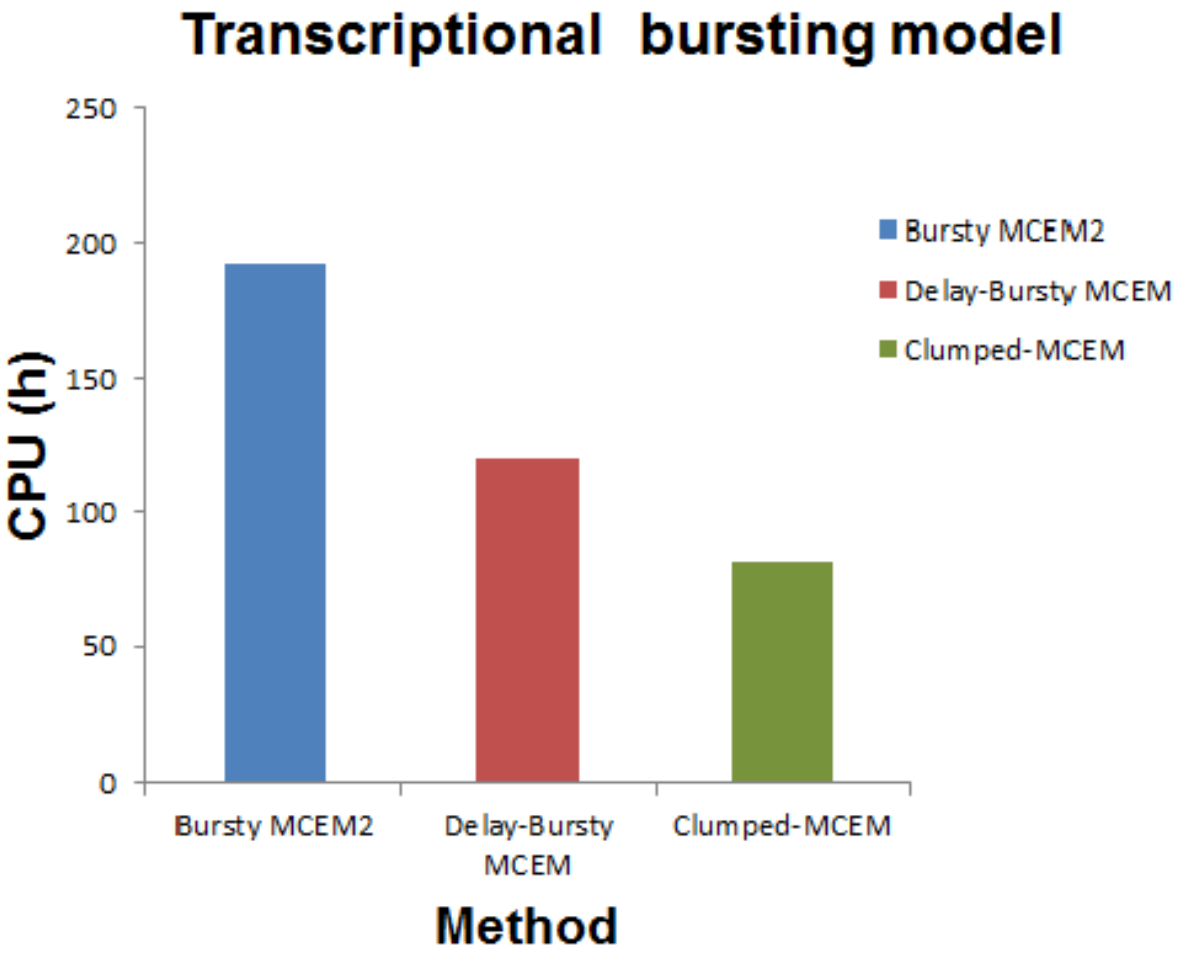


Figure 5.2: Performance of Clumped-MCEM, Delay-Bursty MCEM and bursty $MCEM^2$ in simulating the multistep promoter model using time-series data.

5.9 Additional results

There may be some other models that match the experimental data equally well. To select such models, the relative likelihood value is calculated. The relative likelihood, that any other model is preferable, is given by (Burnham and Anderson, 2002)

$$exp((AIC_{min} - AIC_i)/2) \; . \tag{5.13}$$

where AIC_{min} denotes the minimum AIC score and AIC_i is the score of the model under consideration. In this work, models with relative likelihoods ≥ 0.368 are considered to constitute probable fits to the data. Tables 5.21 to

5.27 and Tables 5.28 to 5.34 show k_m/k_{off}, AIC score, and relative likelihood, for various values of τ and Number Of States, using glutaminase data for Delay-Bursty MCEM and Clumped-MCEM respectively. The highlighted text indicates the models that best fits the experimental data.

Table 5.21: Delay-Bursty MCEM parameter inference using glutaminase promoter time-series data for Model 4.7.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC	Relative likelihood
1	0.5	2.97	3.57	66.02	18.49	20	3478.17	0.0049
1	0.75	3.44	4.08	71.18	17.44	20	3471.10	0.169
1	**1.10**	**4.27**	**3.79**	**72.24**	**19.06**	**20**	**3467.85**	**0.860**
1	1.40	4.32	3.54	73.13	20.65	20	3470.18	0.268
1	**1.65**	**6.84**	**3.89**	**73.82**	**18.97**	**20**	**3468.89**	**0.511**
1	2.10	8.80	3.56	68.94	19.36	20	3469.89	0.310
1	**3.40**	**18.77**	**3.13**	**67.21**	**21.47**	**20**	**3468.24**	**0.708**
1	**4.00**	**19.79**	**3.23**	**68.75**	**21.28**	**20**	**3469.25**	**0.427**
1	**4.10**	**26.20**	**3.28**	**69.09**	**21.06**	**20**	**3468.55**	**0.606**
1	**4.75**	**45.33**	**3.50**	**65.60**	**18.74**	**20**	**3467.55**	**1**
1	**5.00**	**50.70**	**2.96**	**63.54**	**21.46**	**20**	**3469.31**	**0.414**

Table 5.22: Delay-Bursty MCEM parameter inference using glutaminase promoter time-series data for Model 4.8.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC	Relative likelihood
2	0.5	3.13	3.73	66.06	17.71	20	3476.68	0.000 81
2	0.75	3.22	3.60	70.91	19.69	20	3473.79	0.0034
2	1.10	4.43	3.87	72.62	18.76	20	3470.25	0.020
2	1.40	4.89	3.64	72.89	20.02	20	3465.37	0.232
2	1.65	6.58	3.89	74.95	19.26	20	3468.70	0.043
2	2.10	8.43	3.64	70.71	19.42	20	3467.16	0.094
2	**3.40**	**18.79**	**3.55**	**68.62**	**19.32**	**20**	**3463.77**	**0.516**
2	4.00	22.34	3.28	67.65	20.62	20	3466.19	0.154
2	**4.10**	**25.79**	**3.48**	**71.08**	**20.42**	**20**	**3464.17**	**0.423**
2	**4.75**	**48.37**	**3.40**	**64.36**	**18.92**	**20**	**3462.45**	**1**
2	**5.00**	**45.30**	**3.12**	**65.53**	**21.00**	**20**	**3463.45**	**0.606**

Table 5.23: Delay-Bursty MCEM parameter inference using glutaminase promoter time-series data for Model 4.9.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC	Relative likelihood
3	0.5	3.00	3.63	66.58	18.34	20	3476.51	0.0017
3	0.75	3.23	3.44	70.46	20.48	20	3473.48	0.0078
3	1.10	4.36	3.71	72.00	19.40	20	3470.32	0.038
3	1.40	4.98	3.62	73.30	20.24	20	3467.78	0.135
3	1.65	6.53	3.74	73.70	19.70	20	3469.18	0.067
3	**2.10**	**7.56**	**3.45**	**70.52**	**20.44**	**20**	**3465.09**	**0.519**
3	3.40	19.43	3.48	67.95	19.52	20	3467.50	0.155
3	4.00	24.18	3.26	67.62	20.74	20	3465.99	0.331
3	**4.10**	**24.85**	**3.44**	**71.70**	**20.84**	**20**	**3465.65**	**0.392**
3	**4.75**	**47.50**	**3.17**	**67.30**	**21.23**	**20**	**3463.78**	**1**
3	**5.00**	**41.69**	**3.30**	**65.38**	**19.81**	**20**	**3465.42**	**0.440**

Table 5.24: Delay-Bursty MCEM parameter inference using glutaminase promoter time-series data for Model 4.10.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC	Relative likelihood
4	0.5	2.93	3.69	66.55	18.03	20	3475.98	0.0013
4	0.75	3.42	3.67	69.33	18.89	20	3474.17	0.0032
4	1.10	4.27	3.78	73.13	19.34	20	3470.04	0.025
4	1.40	5.27	3.65	73.20	20.05	20	3467.86	0.077
4	1.65	6.88	3.97	74.11	18.66	20	3468.22	0.0645
4	2.10	6.90	3.30	70.40	21.33	20	3465.86	0.2101
4	3.40	18.53	3.39	68.51	20.20	20	3464.76	0.364
4	**4.00**	**25.93**	**3.22**	**67.32**	**20.90**	**20**	**3463.94**	**0.548**
4	4.10	24.61	3.45	70.84	20.53	20	3465.4	0.264
4	**4.75**	**44.46**	**3.20**	**67.49**	**21.09**	**20**	**3462.74**	**1**
4	5.00	44.09	3.23	65.51	20.28	20	3469.26	0.038

Table 5.25: Delay-Bursty MCEM parameter inference using glutaminase promoter time-series data for Model 4.11.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC	Relative likelihood
5	0.5	3.00	3.81	67.04	17.59	20	3476.03	0.0015
5	0.75	3.09	3.55	69.46	19.56	20	3475.37	0.0022
5	1.10	4.38	3.85	73.13	18.99	20	3469.86	0.034
5	1.40	5.10	3.75	72.33	19.28	20	3468.99	0.053
5	1.65	6.70	3.87	72.91	18.83	20	3468.71	0.0617
5	2.10	7.22	3.29	69.00	20.97	20	3468.97	0.0542
5	**3.40**	**18.86**	**3.32**	**67.64**	**20.37**	**20**	**3464.22**	**0.582**
5	**4.00**	**26.24**	**3.20**	**65.66**	**20.51**	**20**	**3464.53**	**0.499**
5	4.10	25.87	3.51	70.77	20.16	20	3465.15	0.366
5	**4.75**	**46.20**	**3.43**	**66.06**	**19.25**	**20**	**3463.14**	**1**
5	**5.00**	**40.66**	**3.11**	**67.41**	**21.67**	**20**	**3464.64**	**0.472**

Table 5.26: Delay-Bursty MCEM parameter inference using glutaminase promoter time-series data for Model 4.12.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC	Relative likelihood
10	0.5	3.03	4.03	70.86	17.58	20	3477.00	0.001 07
10	0.75	3.52	3.84	70.59	18.38	20	3472.88	0.0084
10	1.10	4.45	3.89	73.12	18.79	20	3471.10	0.0205
10	1.40	5.32	3.76	72.88	19.38	20	3469.35	0.049
10	1.65	6.26	3.68	72.45	19.68	20	3467.27	0.139
10	2.10	8.45	3.39	69.04	20.36	20	3469.83	0.038
10	**3.40**	**19.40**	**3.47**	**69.17**	**19.93**	**20**	**3465.22**	**0.388**
10	**4.00**	**25.55**	**3.45**	**69.99**	**20.28**	**20**	**3464.87**	**0.463**
10	**4.10**	**27.82**	**3.14**	**67.16**	**21.38**	**20**	**3465.08**	**0.416**
10	**4.75**	**42.57**	**3.19**	**65.60**	**20.56**	**20**	**3463.33**	**1**
10	**5.00**	**45.59**	**3.15**	**66.10**	**20.98**	**20**	**3464.39**	**0.588**

Table 5.27: Delay-Bursty MCEM parameter inference using glutaminase promoter time-series data for Model 4.13.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC	Relative likelihood
15	0.5	3.02	4.04	70.67	17.49	20	3476.14	0.000 58
15	0.75	3.49	3.84	70.92	18.46	20	3473.14	0.0026
15	1.10	4.29	3.79	72.96	19.25	20	3469.84	0.0136
15	1.40	5.25	3.71	72.57	19.56	20	3468.60	0.0253
15	1.65	6.13	3.71	72.89	19.64	20	3467.4	0.0461
15	2.10	8.48	3.55	70.51	19.86	20	3467.18	0.0515
15	3.40	19.58	3.46	69.27	20.02	20	3464.02	0.250
15	4.00	27.58	3.46	70.01	20.23	20	3463.78	0.282
15	4.10	31.01	3.43	68.88	20.08	20	3465.34	0.129
15	**4.75**	**41.64**	**3.15**	**63.36**	**20.11**	**20**	**3461.25**	**1**
15	5.00	45.25	3.29	67.68	20.57	20	3464.53	0.193

Table 5.28: Clumped-MCEM parameter inference using glutaminase promoter time-series data for Model 4.7.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC	Relative likelihood
1	0.5	3.00	3.77	69.03	18.31	20	3478.16	0.000 93
1	0.75	3.38	4.11	74.32	18.08	20	3476.71	0.0019
1	1.10	3.76	3.68	76.03	20.66	20	3469.46	0.072
1	1.40	5.16	3.76	73.41	19.52	20	3472.05	0.019
1	1.65	7.60	4.43	74.76	16.87	20	3470.29	0.047
1	2.10	10.06	3.84	69.37	18.06	20	3467.80	0.165
1	3.40	22.61	3.31	67.41	20.36	20	3467.89	0.158
1	4.00	26.20	2.97	62.70	21.11	20	3471.95	0.020
1	4.10	23.44	3.38	69.44	20.54	20	3468.16	0.138
1	**4.75**	**49.22**	**3.35**	**67.50**	**20.14**	**20**	**3464.20**	**1**
1	**5.00**	**44.51**	**3.40**	**69.64**	**20.48**	**20**	**3464.79**	**0.744**

Table 5.29: Clumped-MCEM parameter inference using glutaminase promoter time-series data for Model 4.8.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC	Relative likelihood
2	0.5	2.89	3.95	70.85	17.93	20	3474.89	0.0015
2	0.75	3.26	3.80	75.22	19.79	20	3478.29	0.000 28
2	1.10	4.09	3.92	75.20	19.18	20	3472.57	0.0048
2	1.40	5.36	3.84	75.01	19.53	20	3467.90	0.050
2	1.65	7.67	4.32	76.06	17.60	20	3470.97	0.010
2	2.10	10.01	3.50	69.38	19.82	20	3467.70	0.055
2	3.40	21.79	3.41	68.69	20.14	20	3464.82	0.235
2	4.00	29.96	3.03	62.88	20.75	20	3464.69	0.251
2	4.10	26.54	3.53	70.49	19.96	20	3466.18	0.119
2	**4.75**	**44.50**	**3.22**	**67.25**	**20.88**	**20**	**3461.93**	**1**
2	5.00	40.23	3.32	70.21	21.14	20	3464.84	0.233

Table 5.30: Clumped-MCEM parameter inference using glutaminase promoter time-series data for Model 4.9.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC	Relative likelihood
3	0.5	2.91	3.83	69.49	18.14	20	3476.91	0.000 69
3	0.75	3.29	4.07	73.79	18.13	20	3475.89	0.0011
3	1.10	4.14	3.89	74.22	19.07	20	3468.52	0.046
3	1.40	5.25	3.75	73.65	19.64	20	3465.47	0.213
3	1.65	7.88	4.45	77.67	17.45	20	3467.19	0.090
3	2.10	10.17	3.85	70.40	18.28	20	3466.82	0.108
3	**3.40**	**20.07**	**3.33**	**68.45**	**20.55**	**20**	**3462.85**	**0.790**
3	4.00	34.38	3.09	62.63	20.26	20	3467.19	0.090
3	**4.10**	**29.98**	**3.59**	**70.94**	**19.76**	**20**	**3464.23**	**0.396**
3	**4.75**	**41.87**	**3.28**	**66.98**	**20.42**	**20**	**3462.38**	**1**
3	5.00	42.19	3.54	71.86	20.29	20	3466.86	0.106

Table 5.31: Clumped-MCEM parameter inference using glutaminase promoter time-series data for Model 4.10.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC	Relative likelihood
4	0.5	2.78	3.95	71.39	18.07	20	3477.91	0.0007
4	0.75	3.50	3.98	72.01	18.09	20	3472.68	0.0096
4	1.10	4.28	3.57	71.21	19.94	20	3474.63	0.0036
4	1.40	5.14	3.78	75.41	19.94	20	3470.73	0.026
4	1.65	7.11	4.21	77.58	18.42	20	3469.49	0.047
4	**2.10**	**8.64**	**3.44**	**69.63**	**20.24**	**20**	**3465.24**	**0.398**
4	3.40	21.15	3.47	68.59	19.76	20	3466.23	0.242
4	**4.00**	**30.94**	**3.05**	**61.36**	**20.11**	**20**	**3465.18**	**0.410**
4	**4.10**	**27.83**	**3.65**	**72.38**	**19.83**	**20**	**3463.49**	**0.955**
4	**4.75**	**43.33**	**3.18**	**67.45**	**21.21**	**20**	**3463.40**	**1**
4	5.00	40.40	3.60	69.79	19.38	20	3467.09	0.158

Table 5.32: Clumped-MCEM parameter inference using glutaminase promoter time-series data for Model 4.11.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC	Relative likelihood
5	0.5	3.01	4.21	72.52	17.22	20	3479.93	0.000 24
5	0.75	3.63	4.00	70.36	17.59	20	3475.69	0.002
5	1.10	4.42	3.90	70.65	18.11	20	3470.04	0.034
5	1.40	4.88	3.85	75.56	19.62	20	3467.75	0.109
5	1.65	6.76	4.33	77.43	17.88	20	3469.57	0.044
5	2.10	9.46	3.67	70.31	19.15	20	3466.92	0.166
5	3.40	20.09	3.40	69.19	20.35	20	3465.45	0.346
5	4.00	32.06	2.92	62.83	21.51	20	3467.89	0.102
5	**4.10**	**28.15**	**3.88**	**74.15**	**19.11**	**20**	**3464.93**	**0.449**
5	**4.75**	**42.01**	**3.35**	**67.39**	**20.11**	**20**	**3463.33**	**1**
5	**5.00**	**43.92**	**3.64**	**71.09**	**19.53**	**20**	**3464.54**	**0.546**

Table 5.33: Clumped-MCEM parameter inference using glutaminase promoter time-series data for Model 4.12.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC	Relative likelihood
10	0.5	3.02	4.18	72.65	17.38	20	3476.45	0.0015
10	0.75	3.54	3.88	70.84	18.25	20	3473.72	0.0062
10	1.10	4.36	3.86	73.57	19.05	20	3471.29	0.020
10	1.40	4.79	3.93	77.28	19.66	20	3468.89	0.069
10	1.65	6.42	3.93	75.05	19.09	20	3468.40	0.088
10	2.10	9.48	3.53	69.68	19.73	20	3466.36	0.246
10	**3.40**	**19.89**	**3.46**	**68.77**	**19.87**	**20**	**3464.57**	**0.603**
10	**4.00**	**30.66**	**3.07**	**63.90**	**20.81**	**20**	**3464.47**	**0.634**
10	**4.10**	**28.10**	**3.56**	**71.79**	**20.16**	**20**	**3464.55**	**0.609**
10	**4.75**	**39.14**	**3.27**	**66.73**	**20.40**	**20**	**3463.56**	**1**
10	**5.00**	**35.08**	**3.29**	**68.11**	**20.70**	**20**	**3465.42**	**0.394**

Table 5.34: Clumped-MCEM parameter inference using glutaminase promoter time-series data for Model 4.13.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC	Relative likelihood
15	0.5	2.96	4.14	72.70	17.56	20	3475.75	0.000 43
15	0.75	3.45	3.82	71.35	18.67	20	3472.87	0.0018
15	1.10	4.50	3.79	72.05	19.01	20	3471.24	0.0041
15	1.40	5.25	3.84	75.16	19.57	20	3469.21	0.011
15	1.65	6.31	3.86	74.22	19.22	20	3467.42	0.028
15	2.10	8.70	3.54	69.74	19.70	20	3466.53	0.043
15	3.40	20.08	3.34	67.92	20.33	20	3463.80	0.171
15	4.00	29.72	3.14	64.73	20.61	20	3463.70	0.179
15	4.10	27.51	3.51	70.56	20.10	20	3463.88	0.164
15	**4.75**	**35.76**	**3.26**	**67.62**	**20.74**	**20**	**3460.27**	**1**
15	5.00	44.92	3.22	67.25	20.88	20	3464.48	0.121

5.10 Summary

This chapter presents two inference method developed in this work, namely, Delay-Bursty MCEM and Clumped-MCEM. Application of these algorithms to time-series data of endogenous mouse glutaminase promoter, validates the model assumptions and infer the kinetic parameters. Comparison of Delay-Bursty MCEM and Clumped-MCEM with $BurstyMCEM^2$ reveals that Clumped-MCEM produces same numerical accuracy in less time.

Table 5.35: Summary of parameter inference methods

Method	Description
Bursty MCEM^2	- A novel model reduction using time-dependent functions for multistep promoters along with an efficient computational technique for inferring the unknown parameters from single-cell gene expression data.
Delay-Bursty MCEM	- A novel model reduction using delay distribution for multistep promoters along with an efficient computational technique for inferring the unknown parameters from single-cell gene expression data.
Clumped-MCEM	- A novel model reduction using delay distribution for multistep promoters along with an efficient computational technique for inferring the unknown parameters from single-cell gene expression data. The Clumped-MCEM produces same numerical accuracy as Bursty $MCEM^2$ and Delay-Bursty MCEM in less time.

Chapter 6

Conclusion and Future Work

This thesis focuses on novel model reduction techniques for modeling multistep reactions, along with computational methods for inferring unknown kinetic parameters from single-cell time-series data.

First, a novel model reduction strategy is devised, representing several number of promoter OFF states by a single state, accompanied by specifying a time delay for burst frequency. This model approximates complex promoter switching behavior with Erlang-distributed ON/OFF times. To explore combined effects of parameter inference and simulation, using this model reduction, two inference methods are developed namely, Delay-Bursty MCEM and Clumped-MCEM. Simulation of these methods are performed by modifying two existing simulation algorithms, namely, Delay Stochastic Simulation Algorithm (DSSA) and Modified Cai's Exact SSA Method (MCEM). Both these algorithms are based on the idea of delays, to provide accurate representation of proposed multistep models.

Using these methods, computational cost of inferring parameters in multistep models can be greatly reduced. For example, modeling a promoter switching between five OFF states and single ON state requires six switching parameters and simulation of six reactions per transcription process. However, introducing time delays in transcriptional bursting model reduces model complexity as well as computational cost.

The application of these methods to time-series data of endogenous mouse glutaminase promoter validates the model assumptions and infer

values of kinetic parameters. Simulation results show that: (1) models with multiple OFF states produce behaviour that is most consistent with experimental data and also reveals that bursting kinetics are promoter specific (2) Delay-Bursty MCEM and Clumped-MCEM inference are more efficient for time-series data. The comparison with the state-of-the-art Bursty $MCEM^2$ method shows that Delay-Bursty MCEM and Clumped-MCEM produce the similar numerical accuracy. Further, when these methods are compared in terms of efficiency, it is observed that Delay-Bursty MCEM reduces computational cost by 37.44% as compared to Bursty $MCEM^2$. Clumped-MCEM reduces computational cost by 57.58% and 32.19% as compared to Bursty $MCEM^2$ and Delay-Bursty MCEM respectively.

In conclusion, Delay-Bursty MCEM and Clumped-MCEM reduce the model complexity involved in modeling multistep reactions, and enables efficient simulation and parameter inference. These methods provide faster and more accurate parameter inference and simulation of more complex models. This can open new perspectives in Systems Biology, where researchers have to often balance the accuracy of their parameter inference and simulations with the need of considering complex models.

Scope for future work

Some possibilities for further research are presented below:

- The proposed models assume that switching rates are identical in the multistep reactions. Future work can consider extending this approach to nonidentical switching rates.

- Both Delay-Bursty MCEM and Clumped-MCEM are based on the mass action kinetics. There is scope to extend this research work for other types of multistep reactions.

- Both Delay-Bursty MCEM and Clumped-MCEM consider delayed and nondelayed nonconsuming reactions. These methods can potentially be extended to delayed and nondelayed consuming reactions.

- Finally, extending simulation techniques for systems that are not well-mixed also presents many challenges, for which good solutions are needed.

Appendix A

Other Models

This appendix presents additional experiments carried out for multistep ON states and single OFF state model. It also describes parameter inference performed, using synthethic data for multistep OFF state and single ON state model.

A.1 Multistep ON model : parameter inference using time-series data

$$
\begin{aligned}
DNA_{on_1} &\underset{k_{on}}{\overset{k_{off},\tau}{\rightleftharpoons}} DNA_{off} \\
DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\
mRNA &\xrightarrow{.924} \phi \\
mRNA &\xrightarrow{12.6} mRNA + Protein \\
Protein &\xrightarrow{1.98} \phi\ .
\end{aligned}
\tag{A.1}
$$

$$
\begin{aligned}
DNA_{on_1} \longrightarrow DNA_{on_2} &\xrightarrow{k_{off},\tau} DNA_{off} \\
DNA_{off} &\xrightarrow{k_{on}} DNA_{on} \\
DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\
mRNA &\xrightarrow{.924} \phi \\
mRNA &\xrightarrow{12.6} mRNA + Protein \\
Protein &\xrightarrow{1.98} \phi\ .
\end{aligned}
\tag{A.2}
$$

$$
\begin{aligned}
DNA_{on_1} \longrightarrow \ldots \longrightarrow DNA_{on_3} &\xrightarrow{k_{off},\tau} DNA_{off} \\
DNA_{off} &\xrightarrow{k_{on}} DNA_{on} \\
DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\
mRNA &\xrightarrow{.924} \phi \\
mRNA &\xrightarrow{12.6} mRNA + Protein \\
Protein &\xrightarrow{1.98} \phi\ .
\end{aligned}
\tag{A.3}
$$

$$
\begin{aligned}
DNA_{on_1} \longrightarrow \ldots \longrightarrow DNA_{on_4} &\xrightarrow{k_{off},\tau} DNA_{off} \\
DNA_{off} &\xrightarrow{k_{on}} DNA_{on} \\
DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\
mRNA &\xrightarrow{.924} \phi \\
mRNA &\xrightarrow{12.6} mRNA + Protein \\
Protein &\xrightarrow{1.98} \phi\ .
\end{aligned}
\tag{A.4}
$$

$$
\begin{aligned}
DNA_{on_1} \longrightarrow \ldots \longrightarrow DNA_{on_5} &\xrightarrow{k_{off},\tau} DNA_{off} \\
DNA_{off} &\xrightarrow{k_{on}} DNA_{on} \\
DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\
mRNA &\xrightarrow{.924} \phi \\
mRNA &\xrightarrow{12.6} mRNA + Protein \\
Protein &\xrightarrow{1.98} \phi\ .
\end{aligned}
\tag{A.5}
$$

$$
\begin{aligned}
DNA_{on_1} \longrightarrow \ldots \longrightarrow DNA_{on_{10}} &\xrightarrow{k_{off},\tau} DNA_{off} \\
DNA_{off} &\xrightarrow{k_{on}} DNA_{on} \\
DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\
mRNA &\xrightarrow{.924} \phi \\
mRNA &\xrightarrow{12.6} mRNA + Protein \\
Protein &\xrightarrow{1.98} \phi\ .
\end{aligned}
\tag{A.6}
$$

$$
\begin{aligned}
DNA_{on_1} \longrightarrow \ldots \longrightarrow DNA_{on_{15}} &\xrightarrow{k_{off},\tau} DNA_{off} \\
DNA_{off} &\xrightarrow{k_{on}} DNA_{on} \\
DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\
mRNA &\xrightarrow{.924} \phi \\
mRNA &\xrightarrow{12.6} mRNA + Protein \\
Protein &\xrightarrow{1.98} \phi\ .
\end{aligned}
\tag{A.7}
$$

The unknown kinetic parameters of the model A.1, A.2, A.3, A.4, A.5, A.6, A.7 are $\tau, k_{on}, k_{off}, k_m$. These models include the mRNA degradation, protein translation and degradation reactions with .924 (Suter et al., 2011), 12.6 (Molina et al., 2013), 1.98(Suter et al., 2011) respectively. Model A.1, A.2, A.3, A.4, A.5, A.6, A.7 represents 1, 2, 3, 4, 5, 10, 15 promoter OFF states respectively. These models include bursting, with the correct parameterization. It assumes random bursts production. The unknown parameters of the model are initialized to 1. But c_3 has been initialized to 0.5. The unobserved initial promoter state and number of mRNAs is initialized to DNA_{on} and 20 respectively. The time delay value, when present, ranging from 0.5-5 is selected (denoted as τ). Tables A.1 to A.7 and Tables A.8 to A.14 displays results for multistep ON states and single OFF state for Delay-Bursty MCEM and Clumped-MCEM using glutaminase promoter time-series data respectively. Simulation results reveal, that model with multistep ON states and single OFF state does not agree well with experimental data.

Table A.1: Delay-Bursty MCEM parameter inference using glutaminase promoter time-series data for Model A.1

Number Of States	τ	k_{off}	k_{on}	k_m	AIC
1	0.5	11.03	1.77	69.51	3484.32
1	0.75	18.99	1.87	68.73	3481.57
1	1.10	33.48	1.46	76.44	3483.09
1	1.40	25.30	1.64	58.17	3485.13
1	1.65	59.12	1.64	67.74	3481.36
1	2.10	114.12	1.44	71.58	3482.56
1	3.40	476.82	1.48	63.61	3486.48
1	4.00	1474.37	1.52	65.40	3485.14
1	4.10	1982.29	1.57	68.61	3483.65
1	4.75	1960.33	1.52	63.63	3489.20
1	5.00	2491.61	1.58	59.58	3490.54

Table A.2: Delay-Bursty MCEM parameter inference using glutaminase promoter time-series data for Model A.2

Number Of States	τ	k_{off}	k_{on}	k_m	AIC
2	0.5	10.87	1.75	68.13	3484.50
2	0.75	18.32	1.88	69.72	3483.63
2	1.10	32.60	1.50	76.91	3483.86
2	1.40	26.47	1.60	59.03	3485.92
2	1.65	65.81	1.69	66.33	3477.61
2	2.10	114.87	1.55	70.22	3482.57
2	3.40	504.73	1.65	62.39	3488.62
2	4.00	1425.18	1.56	66.12	3485.31
2	4.10	2040.87	1.61	66.53	3486.94
2	4.75	1907.24	1.46	63.27	3487.61
2	5.00	2774.97	1.68	59.05	3485.93

Table A.3: Delay-Bursty MCEM parameter inference using glutaminase promoter time-series data for Model A.3

Number Of States	τ	k_{off}	k_{on}	k_m	AIC
3	0.5	11.49	1.94	68.21	3484.50
3	0.75	18.02	1.76	71.47	3484.46
3	1.10	31.56	1.51	75.99	3484.07
3	1.40	26.60	1.61	59.60	3484.80
3	1.65	62.26	1.63	66.69	3483.44
3	2.10	119.44	1.52	70.03	3484.34
3	3.40	477.097	1.52	62.58	3489.14
3	4.00	1321.88	1.57	65.29	3485.06
3	4.10	2040.16	1.61	65.97	3486.47
3	4.75	2002.73	1.46	62.79	3487.47
3	5.00	2631.97	1.56	58.60	3487.36

Table A.4: Delay-Bursty MCEM parameter inference using glutaminase promoter time-series data for Model A.4

Number Of States	τ	k_{off}	k_{on}	k_m	AIC
4	0.5	11.14	1.80	68.64	3483.49
4	0.75	17.10	1.67	72.42	3486.40
4	1.10	30.79	1.53	75.95	3483.19
4	1.40	25.90	1.59	60.46	3485.03
4	1.65	58.58	1.67	66.17	3482.59
4	2.10	121.37	1.52	70.94	3481.99
4	3.40	453.26	1.51	64.62	3483.09
4	4.00	1422.13	1.55	67.84	3485.43
4	4.10	1900.97	1.58	67.67	3485.19
4	4.75	1930.04	1.53	60.07	3490.21
4	5.00	2473.80	1.59	60.61	3489.66

Table A.5: Delay-Bursty MCEM parameter inference using glutaminase promoter time-series data for Model A.5

Number Of States	τ	k_{off}	k_{on}	k_m	AIC
5	0.5	10.13	1.66	68.77	3484.38
5	0.75	18.10	1.71	71.25	3484.49
5	1.10	32.05	1.55	75.03	3482.08
5	1.40	27.52	1.59	61.22	3484.63
5	1.65	68.42	1.71	65.78	3485.41
5	2.10	125.41	1.57	70.13	3484.08
5	3.40	481.46	1.49	65.88	3489.02
5	4.00	1584.53	1.58	66.53	3484.47
5	4.10	1862.30	1.54	67.22	3487.40
5	4.75	2295.26	1.59	61.62	3488.46
5	5.00	2798.56	1.53	61.62	3489.80

Table A.6: Delay-Bursty MCEM parameter inference using glutaminase promoter time-series data for Model A.6

Number Of States	τ	k_{off}	k_{on}	k_m	AIC
10	0.5	11.13	1.74	70.46	3483.52
10	0.75	17.99	1.68	72.40	3482.16
10	1.10	32.98	1.67	72.67	3481.35
10	1.40	30.15	1.58	63.06	3482.76
10	1.65	61.64	1.57	70.95	3484.18
10	2.10	112.04	1.51	69.65	3482.88
10	3.40	529.27	1.54	65.49	3485.84
10	4.00	1604.51	1.51	67.86	3485.52
10	4.10	1787.93	1.53	67.68	3486.03
10	4.75	2694.76	1.57	63.34	3487.1
10	5.00	3427.36	1.53	62.95	3487.57

Table A.7: Delay-Bursty MCEM parameter inference using glutaminase promoter time-series data for Model A.7

Number Of States	τ	k_{off}	k_{on}	k_m	AIC
15	0.5	11.36	1.72	71.82	3483.1
15	0.75	16.86	1.65	71.74	3481.81
15	1.10	29.17	1.60	71.41	3481.96
15	1.40	32.33	1.56	64.93	3482.05
15	1.65	53.06	1.53	68.90	3483.53
15	2.10	113.77	1.53	69.35	3482.92
15	3.40	586.16	1.53	66.46	3484.51
15	4.00	1605.07	1.53	67.71	3485.32
15	4.10	1800.45	1.53	67.66	3486.08
15	4.75	2884.39	1.51	64.47	3487.22
15	5.00	4062.69	1.56	63.48	3487.04

Table A.8: Clumped-MCEM parameter inference using glutaminase promoter time-series data for Model A.1

Number Of States	τ	k_{off}	k_{on}	k_m	AIC
1	0.5	9.53	1.60	70.92	3483.30
1	0.75	21.54	1.82	71.70	3478.96
1	1.10	32.55	1.64	74.48	3489.45
1	1.40	29.59	1.64	59.27	3488.62
1	1.65	46.66	1.56	65.71	3485.28
1	2.10	126.31	1.57	71.81	3485.71
1	3.40	423.09	1.59	62.20	3488.58
1	4.00	1405.73	1.66	64.68	3487.17
1	4.10	1750.40	1.52	69.75	3487.48
1	4.75	2121.67	1.45	63.32	3490.49
1	5.00	2664.34	1.56	60.71	3492.88

Table A.9: Clumped-MCEM parameter inference using glutaminase promoter time-series data for Model A.2

Number Of States	τ	k_{off}	k_{on}	k_m	AIC
2	0.5	10.81	1.67	72.19	3480.91
2	0.75	20.76	1.93	71.91	3486
2	1.10	29.20	1.55	72.99	3480.61
2	1.40	29.41	1.60	62.09	3483.30
2	1.65	43.14	1.44	66.48	3486.69
2	2.10	150.19	1.48	72.15	3486.03
2	3.40	404.80	1.64	61.20	3484.01
2	4.00	1314.91	1.63	64.02	3489.78
2	4.10	1961.96	1.56	67.97	3487.51
2	4.75	1957.79	1.41	64.48	3491.60
2	5.00	2702.06	1.41	63.42	3489.18

Table A.10: Clumped-MCEM parameter inference using glutaminase promoter time-series data for Model A.3

Number Of States	τ	k_{off}	k_{on}	k_m	AIC
3	0.5	10.36	1.57	71.78	3480.4
3	0.75	20.02	1.83	72.85	3483.53
3	1.10	32.26	1.60	74.56	3477.75
3	1.40	30.73	1.64	63.18	3481.85
3	1.65	43.93	1.56	65.89	3483.61
3	2.10	162.55	1.57	72.52	3482.49
3	3.40	405.85	1.57	61.22	3486.45
3	4.00	1242.35	1.61	63.84	3487.38
3	4.10	1780.24	1.51	67.73	3485.74
3	4.75	2026.37	1.37	66.29	3488.95
3	5.00	2641.90	1.45	62.35	3491.61

Table A.11: Clumped-MCEM parameter inference using glutaminase promoter time-series data for Model A.4

Number Of States	τ	k_{off}	k_{on}	k_m	AIC
4	0.5	10.90	1.64	72.10	3485.31
4	0.75	20.25	1.82	71.36	3484.57
4	1.10	31.63	1.53	76.17	3483.26
4	1.40	29.25	1.61	62.49	3481.73
4	1.65	47.01	1.54	66.94	3486.35
4	2.10	170.46	1.53	73.65	3482.14
4	3.40	418.10	1.51	62.54	3486.52
4	4.00	1196.37	1.50	64.95	3486.54
4	4.10	1989.89	1.58	66.86	3483.23
4	4.75	2090.92	1.41	64.52	3490.55
4	5.00	2760.17	1.56	61.41	3488.55

Table A.12: Clumped-MCEM parameter inference using glutaminase promoter time-series data for Model A.5

Number Of States	τ	k_{off}	k_{on}	k_m	AIC
5	0.5	11.07	1.71	71.60	3483.41
5	0.75	18.92	1.73	72.07	3483.05
5	1.10	30.65	1.52	76.76	3483.39
5	1.40	30.36	1.57	63.13	3482.71
5	1.65	56.12	1.62	67.72	3482.79
5	2.10	170.19	1.66	74.04	3485.27
5	3.40	447.48	1.53	62.65	3488.09
5	4.00	1109.80	1.52	64.53	3487.45
5	4.10	2155.95	1.61	66.06	3487.58
5	4.75	2445.56	1.37	64.74	3489.05
5	5.00	2558.23	1.54	60.77	3491

Table A.13: Clumped-MCEM parameter inference using glutaminase promoter time-series data for Model A.6

Number Of States	τ	k_{off}	k_{on}	k_m	AIC
10	0.5	11.55	1.79	70.60	3482.89
10	0.75	18.67	1.70	73.91	3483.41
10	1.10	33.13	1.60	75.01	3482.67
10	1.40	34.48	1.62	65.35	3482.58
10	1.65	62.63	1.59	69.50	3482.69
10	2.10	187.39	1.61	75.08	3483.44
10	3.40	467.57	1.53	63.23	3485.91
10	4.00	1011.60	1.53	63.21	3487.39
10	4.10	2516.22	1.55	69.48	3486.11
10	4.75	2847.78	1.54	63.73	3488.83
10	5.00	2256.70	1.54	61.32	3489.57

Table A.14: Clumped-MCEM parameter inference using glutaminase promoter time-series data for Model A.7

Number Of States	τ	k_{off}	k_{on}	k_m	AIC
15	0.5	11.45	1.75	71.42	3482.92
15	0.75	17.84	1.67	73.34	3482.09
15	1.10	31.98	1.62	73.64	3482.05
15	1.40	37.55	1.61	66.52	3482.41
15	1.65	61.48	1.58	69.40	3482.49
15	2.10	172.05	1.56	74.94	3483.15
15	3.40	494.52	1.53	64.74	3485.51
15	4.00	1125.95	1.50	65.71	3487.43
15	4.10	2258.79	1.49	70.60	3485.89
15	4.75	2663.96	1.53	63.60	3487.20
15	5.00	2743.42	1.51	62.43	3488.44

A.2 Parameter inference using synthetic data

The unknown kinetic parameters of the model A.8, A.9, A.10, A.11, A.12, A.13, A.14 are $\tau, k_{on}, k_{off}, k_m$. The mRNA degradation rate is given as .0001 for models A.8, A.9, A.10, A.11, A.12, A.13, A.14. The synthetic data is generated in such a way that the model can produce on an average, 20 mRNAs once per time unit, before switching to first OFF state. The promoter state is initialized to DNA_{off_1} for models A.8, A.9, A.10, A.11, A.12, A.13, A.14 respectively for each simulations. The τ value is set to 4.75 for simulations. These models include bursting, with the correct parameterization. Table A.15 represents inferred kinetic rates, mean burst size(k_m/k_{off}) and AIC values, using synthetic data. Related model number is mentioned adjacent to the number of states in the first column of Table A.15.

$$\begin{aligned} DNA_{off_1} &\underset{k_{off}}{\overset{k_{on},\tau}{\rightleftharpoons}} DNA_{on} \\ DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\ mRNA &\xrightarrow{.0001} \phi \end{aligned} \tag{A.8}$$

$$\begin{aligned} DNA_{off_1} \longrightarrow DNA_{off_2} &\xrightarrow{k_{on},\tau} DNA_{on} \\ DNA_{on} &\xrightarrow{k_{off}} DNA_{off_1} \\ DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\ mRNA &\xrightarrow{.0001} \phi \end{aligned} \tag{A.9}$$

$$\begin{aligned} DNA_{off_1} \longrightarrow \ldots \longrightarrow DNA_{off_3} &\xrightarrow{k_{on},\tau} DNA_{on} \\ DNA_{on} &\xrightarrow{k_{off}} DNA_{off_1} \\ DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\ mRNA &\xrightarrow{.0001} \phi \end{aligned} \tag{A.10}$$

$$\begin{aligned} DNA_{off_1} \longrightarrow \ldots \longrightarrow DNA_{off_4} &\xrightarrow{k_{on},\tau} DNA_{on} \\ DNA_{on} &\xrightarrow{k_{off}} DNA_{off_1} \\ DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\ mRNA &\xrightarrow{.0001} \phi \end{aligned} \tag{A.11}$$

$$
\begin{aligned}
DNA_{off_1} \longrightarrow \ldots \longrightarrow DNA_{off_5} &\xrightarrow{k_{on},\tau} DNA_{on} \\
DNA_{on} &\xrightarrow{k_{off}} DNA_{off_1} \\
DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\
mRNA &\xrightarrow{.0001} \phi
\end{aligned}
\tag{A.12}
$$

$$
\begin{aligned}
DNA_{off_1} \longrightarrow \ldots \longrightarrow DNA_{off_{10}} &\xrightarrow{k_{on},\tau} DNA_{on} \\
DNA_{on} &\xrightarrow{k_{off}} DNA_{off_1} \\
DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\
mRNA &\xrightarrow{.0001} \phi
\end{aligned}
\tag{A.13}
$$

$$
\begin{aligned}
DNA_{off_1} \longrightarrow \ldots \longrightarrow DNA_{off_{15}} &\xrightarrow{k_{on},\tau} DNA_{on} \\
DNA_{on} &\xrightarrow{k_{off}} DNA_{off_1} \\
DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\
mRNA &\xrightarrow{.0001} \phi
\end{aligned}
\tag{A.14}
$$

Table A.15: Delay-Bursty MCEM parameter inference using synthetic data. Results in bold font shows that mRNA number improves with number of OFF states.

Number Of States	τ	k_{on}	k_{off}	k_m	k_m/k_{off}	observed value	AIC
1(A.8)	4.75	5.39	1.23	16.76	13.62	20	244.14
2(A.9)	4.75	5.18	0.97	15.70	16.18	20	245.14
3(A.10)	4.75	4.68	1.12	17.12	15.28	20	243.8
4(A.11)	4.75	4.75	1.09	17.76	16.29	20	241.78
5(A.12)	4.75	4.50	0.80	14.60	18.25	20	242.74
10(A.13)	4.75	3.61	0.78	15.29	19.60	20	242.16
15(A.14)	**4.75**	**3.18**	**0.72**	**14.48**	**20.11**	**20**	**240.18**

The pictorial representation of of the models A.8 to A.14 is depicted in Fig.A.1.

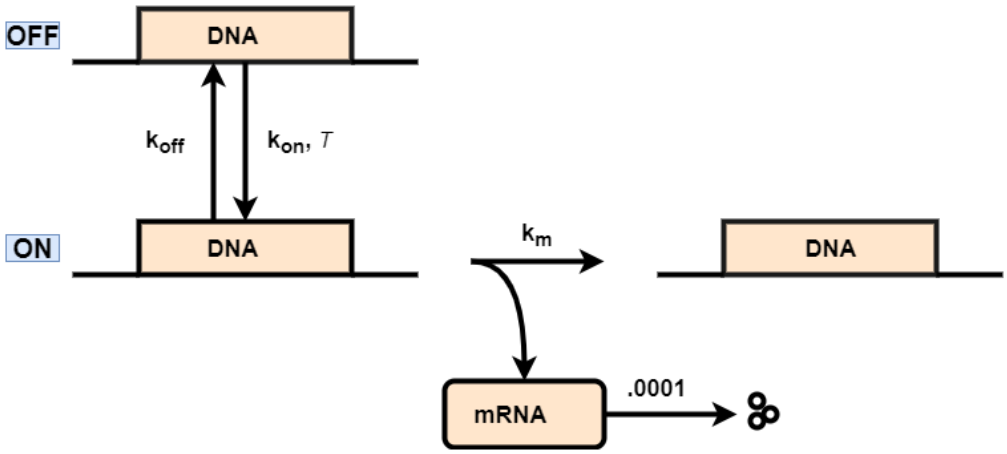


Figure A.1: Diagrammatic representation of model description using time-series data.

A.3 SSA simulation for original formulation

$$\begin{aligned} DNA_{on} &\xrightarrow{5.0} DNA_{on} + mRNA \\ mRNA &\xrightarrow{.0001} \phi \end{aligned} \tag{A.15}$$

Model A.15 produces an average of 5 mRNA molecules per unit of time. mRNA degradation rate is given as .0001. This model exhibits non-bursty production of mRNA. A single trajectory over 100 time units is simulated and recorded the number of mRNA molecules at 400 equally spaced intervals. The initial conditions for this simulation were 0 mRNA molecules and promoter state is set to DNA_{on}.

Using data from Model A.15 (Daigle et al., 2015), unknown parameters are inferred from Model A.16:

$$\begin{aligned} DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\ mRNA &\xrightarrow{.0001} \phi \end{aligned} \tag{A.16}$$

Table A.16: parameter inference using synthetic data for Model A.16.

Model	τ	k_m	observed value	AIC
2	NA	4.81	5	1135.87

τ- delays Not Applicable (NA).

Table A.16 suggests inferred parameter k_m is close to observed value 5. The trajectories of Model A.16 and Model A.17 is simulated using SSA.

$$\begin{aligned} DNA_{on} &\xrightarrow{k_m} DNA_{on} + mRNA \\ mRNA &\xrightarrow{.924} \phi \\ mRNA &\xrightarrow{12.6} mRNA + Protein \\ Protein &\xrightarrow{1.98} \phi\ . \end{aligned} \tag{A.17}$$

Table A.17: parameter inference using glutaminase promoter time-series data for Model A.17.

Model	τ	k_m	observed value	AIC
3	NA	20.53	20	3683.47

τ- delays Not Applicable (NA).

Model A.17 is similar to Model A.16, with protein translation and protein degradation reactions added. The simulation results for glutaminase data inference also suggests that k_m is close to observed value 20.

From the original formulation of Model A.16 and Model A.17, it is clearly understood that Model A.16 and Model A.17 cannot exhibit bursting. To characterize multistep transcriptional bursting model delays are used.

References

Achimescu, S. and Lipan, O. (2006). Signal propagation in nonlinear stochastic gene regulatory networks. *IEE Proceedings - Systems Biology*, 153:120–134.

Akaike, H. (1974). A new look at the statistical model identification. *Automatic Control, IEEE Transactions on 1974*, 19:716–723.

Babtie, A. C., Kirk, P., and Stumpf, M. P. H. (2014). Topological sensitivity analysis for systems biology. *Proceedings of the National Academy of Sciences of the United States of America*, 111:18507–18512.

Barrio, M., Burrage, K., Burrage, P., Leier, A., and Marquez Lago, T. (2010). Computational approaches for modeling intrinsic noise and delays in genetic regulatory networks. *In Das, Sanjoy, Caragea, Doina, Welch, Stephen, Hsu, William H. (Eds.) Handbook of Research on Computational Methodologies in Gene Regulatory Networks. IGI Global, Hershey PA*, pages 169–197.

Barrio, M., Burrage, K., Leier, A., and Tian, T. (2006). Oscillatory regulation of hes1,discrete stochastic delay modelling and simulation. *PLoS Computational Biology*, 2:1017–1030.

Barrio, M., Leier A, and Marquez Lago, T. T. (2013). Reduction of chemical reaction networks through delay distributions. *J Chem Phys.*, 138.

Beaumont, M. A., Zhang, W., and Balding, D. J. (2002). Approximate bayesian computation in population genetics. *Genetics*, 162:2025–2035.

Blake, W. J., KAErn, M., and Collins, J. J. (2003). Noise in eukaryotic gene expression. *Nature*, 422:633–637.

Boeger, H., Griesenbeck, J., and Kornberg, R. D. (2008). Nucleosome retention and the stochastic nature of promoter chromatin remodeling for transcription. *Cell*, 133:716–726.

Boys, R. J., Wilkinson, D. J., and Kirkwood, T. B. L. (2008). Bayesian inference for a discretely observed stochastic kinetic model. *Stat Comput*, 18:125–135.

Burnham, K. and Anderson, D. (2002). Model selection and multimodel inference: a practical information-theoretic approach. *New York: Springer, 2nd edition*.

Burrage, K., Burrage, P., Leier, A., and Marquez Lago, T. (2017). A review of stochastic and delay simulation approaches in both time and space in computational cell biology. *In Holcman, David (Ed.) Stochastic Dynamical Systems in Cellular Biology: Multiscale Modeling, Asymptotics and Numerical Methods. Springer*.

Caffo, B., Jank, W., and Jones, G. L. (2005). Ascent-based monte carlo expectation-maximization. *Series B Statistical Methodology*, 67:235–251.

Cai, L., Dala, l. C. K., and Elowitz, M. B. (2008). Frequency modulated nuclear localization bursts coordinate gene regulation. *Nature*, 455:485–490.

Cai, L., Friedman, N., and Xie, X. S. (2006). Stochastic protein expression in individual cells at the single molecule level. *Nature*, 440:358–362.

Cai, X. (2007). Exact stochastic simulation of coupled chemical reactions with delays. *J. Phys. Chem*, 126.

Caravagna, G. and Hillston J (2010). Modeling biological systems with delays in bio-pepa. *Proceedings of MeCBIC 2010, Electronic Proceedings in Theoretical Computer Science*, 40:85–101.

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

Chubb, J. R. and Liverpool, T. B. (2010). Bursts and pulses: insights from single cell studies into transcriptional mechanisms. *Curr Opin Genet Dev.*, 20:478–484.

Chubb, J. R., Trcek, T., Shenoy, S. M., and Singer, R. H. (2006). Transcriptional pulsing of a developmental gene. *Current biology*, 16:1018–1025.

Daigle, B. J., Soltani, M., Petzold, L., and Singh, A. (2015). Inferring single-cell gene expression mechanisms using stochastic simulation. *Bioinformatics*, 31:1428–1435.

Daigle, B. J. J., Roh, M. K., Petzold, L. R., and Niemi J (2012). Accelerated maximum likelihood parameter estimation for stochastic biochemical systems. *BMC Bioinformatics*, 13:68.

Dar, R. D., Razooky, B. S., Singh, A., Trimeloni, T. V., McCollum, J. M., Cox, C. D., Simpson, M. L., and Weinberger, L. S. (2012). Transcriptional burst frequency and burst size are equally modulated across the human genome. *Proceedings of the National Academy of Sciences of the United States of America*, 109:17454–17459.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *J R Stat Soc Series B (Methodological)*, 39:1–38.

Do, C. B. and Batzoglou, S. (2008). What is the expectation maximization algorithm? *Nature Biotechnology*, 26:897–899.

Dobrzyski, M. and Bruggeman, F. J. (2009). Elongation dynamics shape bursty transcription and translation. *Proceedings of the National Academy of Sciences*, 106:2583–2588.

Eldar, A., Chary, V. K., Xenopoulos, P., Fontes, M. E., and Loson, O. C. (2009). Partial penetrance facilitates developmental evolution in bacteria. *Nature*, 460:510–514.

Evans, M., Hastings, N., and Peacock, B. (2000). Statistical distributions. *Wiley, 3rd edition.*

Fearnhead, P., Giagos, V., and Sherlock C (2014). Inference for reaction networks using the linear noise approximation. *Biometrics*, 70:457–466.

Gandhi, S. J., Zenklusen, D., Lionnet, T., and Singer, R. H. (2011). Transcription of functionally related constitutive genes is not coordinated. *Nat Struct Mol Biol.*, 18:27–34.

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

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem*, 81:2340–2361.

Gillespie, D. T. (2007). Stochastic simulation of chemical kinetics. *Annu Rev Phys Chem*, 58:35–55.

Gillespie, D. T. and Petzold, L. R. (2006). Numerical simulation for biochemical kinetics. *Chapter in:System modeling in cell biology : from concepts to nuts and bolts, MIT Press.*

Golding, I., Paulsson, J., Zawilski, S. M., and Cox, E. C. (2005). Real-time kinetics of gene activity in individual bacteria. *Cell*, 123:1025–1036.

Golightly, A. and Wilkinson, D. J. (2005). Bayesian inference for stochastic kinetic models using a diffusion approximation. *Biometrics*, 61:781–788.

Golightly, A. and Wilkinson, D. J. (2006). Bayesian sequential inference for stochastic kinetic biochemical network models. *Journal of Computational Biology*, 13:838–851.

Golightly, A. and Wilkinson, D. J. (2008). Bayesian inference for nonlinear multivariate diffusion models observed with error. *Computational Statistics Data Analysis*, 52:1674–1693.

Golightly, A. and Wilkinson, D. J. (2011). Bayesian parameter inference for stochastic biochemical network models using particle markov chain monte carlo. *Interface Focus*, 1:807–820.

Halpern, K. B., Tanami, S., Landen, S., Chapal, M., Szlak, L., Hutzler, A., Nizhberg, A., and Itzkovitz, S. (2015). Bursty gene expression in the intact mammalian liver. *Molecular Cell*, 58:147–156.

Harper, C. V., Finkenstadt, B., and Woodcock, D. J. (2011). Dynamic analysis of stochastic transcription cycles. *PLoS Biology*, 9:1–14.

Horvath, A. and Manini, D. (2008). Parameter estimation of kinetic rates in stochastic reaction networks by the em method. *International Conference on BioMedical Engineering and Informatics (BMEI)*, pages 713–717.

Jia, T. and Kulkarni, R. (2011). Intrinsic noise in stochastic models of gene expression with molecular memory and bursting. *Phys. Rev. Lett.*, 106.

Ko M S H (1991). A stochastic model for gene induction. *Journal of Theoretical Biology*, 153:181–194.

Komorowski, M., Finkenstädt, B., Harper, C. V., and Rand D A (2009). Bayesian inference of biochemical kinetic parameters using the linear noise approximation. *BMC Bioinformatics*, 10:343.

Kouyous, R. D., Althaus, C. L., and Bonhoeffer S (2006). Stochastic or deterministic: what is the effective population size of hiv-1? *Trends in Microbiology*, 14:507–511.

Kugler, P. (2012). Moment fitting for parameter inference in repeatedly and partially observed stochastic biological models. *PLoS ONE*, 7:e43001.

Larson, D. R. (2011). What do expression dynamics tell us about the mechanism of transcription? *Curr Opin Genet Dev.*, 21:591–599.

Lee, T. I. and Young, R. A. (2013). Transcriptional regulation and its misregulation in disease. *Cell*, 152:1237–1251.

Leier, A., Barrio, M., and Marquez Lago, T. T. (2014). Exact model reduction with delays: closed-form distributions and extensions to fully bi-directional monomolecular reactions. *J R Soc Interface*, 11.

Liepe, J., Kirk, P., Filippi, S., Toni, T., Barnes, C. P., and Stumpf, M. P. (2014). A framework for parameter estimation and model selection from experimental data in systems biology using approximate bayesian computation. *Nat Protoc.*, 9:439–456.

Lillacci, G. and Khammash, M. (2013). The signal within the noise: efficient inference of stochastic gene regulation models using fluorescence histograms and stochastic simulations. *Bioinformatics*, 29:2311–2319.

Mao, C., Brown, C. R., Falkovskaia, E., Dong, S., Hrabeta Robinson, E., Wenger, L., and Boeger H (2010). Quantitative analysis of the transcription control mechanism. *Mol Syst Biol.*, 6:431–443.

Marchetti, L., Priami, C., and Thanh, V. H. (2017). Simulation algorithms for computational systems biology. *Texts in Theoretical Computer Science. An EATCS Series, Springer International Publishing*.

Mariani, L., Schulz, E. G., Lexberg, M., Helmstetter, C., Radbruch, A., Lhning, M., and Hofer T (2010). Short-term memory in gene induction reveals the regulatory principle behind stochastic il-4 expression. *Mol Syst Biol.*, 6:1–13.

Marjoram, P., Molitor, J., Plagnol, V., and Tavare S (2003). Markov chain monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100:15324–15328.

Miller Jensen, K., Dey, S. S., Schaffer, D., and Arkin, A. P. (2011). Varying virulence:epigenetic control of expression noise and disease processes. *Trends Biotechnol.*, 29:517–525.

Milner, P., Gillespie, C. S., and Wilkinson D J (2013). Moment closure based parameter inference of stochastic kinetic models. *Stat Comput*, 23:287.

Molina, N., Suter, D. M., Cannavo, R., Zoller, B., Gotic, I., and Naef, F. (2013). Stimulus-induced modulation of transcriptional bursting in a single mammalian gene. *Proc Natl Acad Sci*, 110:20563–20568.

Munsky, B. and Khammash, M. (2010). Identification from stochastic cell-to-cell variation: a genetic switch case study. *IET Syst Biol*, 4:356–366.

Munsky, B., Trinh, B., and Khammash, M. (2009). Listening to the noise: random fluctuations reveal gene network parameters. *Mol Syst Biol.*, 5.

Neuert, G., Munsky, B., Tan, R. Z., Teytelman, L., Khammash, M., and Oudenaarden, A. (2013). Systematic identification of signal-activated stochastic gene regulation. *Science*, 339:584–587.

Ochiai, H., Sugawara, T., Sakuma, T., and Yamamoto, T. (2014). Stochastic promoter activation affects nanog expression variability in mouse embryonic stem cells. *Scientific reports*, 4:1–9.

Peccoud, J. and Ycart, B. (1995). Markovian modeling of gene-product synthesis. *Theoretical population biology*, 48:222–234.

Pedraza, J. and Paulsson, J. (2008). Effects of molecular memory and bursting on fluctuations in gene expression. *Science*, 319:339–343.

Pedraza, J. M. and Oudenaarden A (2005). Noise propagation in gene networks. *Science*, 307:1965–1969.

Pritchard, J. K., Seielstad, M. T., Perez Lezaun, A., and Feldman, M. W. (1999). Population growth of human y chromosomes: a study of y chromosome microsatellites. *Mol Biol Evol.*, 16:1791–1798.

Raffard, R. L., Lipan, O., Wong, W. H., and Tomlin, C. J. (2008). Optimal discovery of a stochastic genetic network. *American Control Conference, IEEE*, pages 2773–2779.

Raj, A., Peskin, C. S., Tranchina, D., Vargas, D. Y., and Tyagi S (2006). Stochastic mrna synthesis in mammalian cells. *PLoS Biol*, 4.

Raser, J. M. and OShea, E. K. (2004). Control of stochasticity in eukaryotic gene expression. *Science*, 304:1811–1814.

Ratmann, O., Andrieub, C., Wiufc, C., and Richardsond S (2009). Model criticism based on likelihood-free inference, with an application to protein network evolution. *Proceedings of the National Academy of Sciences*, 106:10576–10581.

Reinker, S., Altman, R. M., and Timmer, J. (2006). Parameter estimation in stochastic biochemical reactions. *Syst Biol (Stevenage)*, 153:168–178.

Robert, C. P. and Casella, G. (2004). Monte carlo statistical methods. *2nd edition, Springer, New York*.

Rubinstein, R. Y. (1997). Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99:89–112.

Rudge, T. and Burrage, K. (2008). Effects of intrinsic and extrinsic noise can accelerate juxtacrine pattern formation. *Bull. Math. Biol.*, 70:971–991.

Ruess, J. and Lygeros, J. (2013). Identifying stochastic biochemical networks from single-cell population experiments: A comparison of approaches based on the fisher information. *52nd IEEE Conference on Decision and Control. IEEE*, pages 2703–2708.

Ruess, J. and Lygeros, J. (2015). Moment-based methods for parameter inference and experiment design for stochastic biochemical reaction networks. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 25.

Sanchez, A., Choubey, S., and Kondev, J. (2013). Stochastic models of transcription: from single molecules to single cells. *Methods*, 62:13–25.

Schilling, C., Bogomolov, S., Henzinger, T. A., Podelski, A., and Ruess J (2016). Adaptive moment closure for parameter inference of biochemical reaction networks. *Biosystems*, 149:15–25.

Scholes, C., DePace, A. H., and Sanchez, A. (2017). Combinatorial gene regulation through kinetic control of the transcription cycle. *Cell systems*, 4:97–108.

Senecal, A., Munsky, B., Proux, F., Ly, N., Braye, F. E., and Zimmer C et al. (2014). Transcription factors modulate c-fos transcriptional bursts. *Cell reports*, 8:75–83.

Shahrezaei, V. and Swain, P. (2008). Analytical distributions for stochastic gene expression. *Proceedings of the National Academy of Sciences*, 105:17256–17261.

Shepherd, D. P., Li, N., Micheva Viteva S N, Munsky B, Hong Geller E, and Werner J H (2013). Counting small rna in pathogenic bacteria. *Anal. Chem.*, 85:4938–4943.

Silk, D., Kirk, P. D. W., Barnes, C. P., Toni, T., and Stumpf, M. P. H. (2014). Model selection in systems biology depends on experimental design. *PLoS Comput Biol*, 10:e1003650.

Singh, A., Razooky, B. S., Dar, R. D., and Weinberger, L. S. (2012). Dynamics of protein noise can distinguish between alternate sources of gene-expression variability. *Mol Syst Biol.*, 8:607.

Singh, A., Razooky, B., Cox, C. D., Simpson, M. L., and Weinberger, L. S. (2010). Transcriptional bursting from the hiv-1 promoter is a significant source of stochastic noise in hiv-1 gene expression. *Biophysical Journal*, 98:L32–L34.

So, L., Ghosh, A., Zong, C., Seplveda, L. A., Segev R, and Golding I (2011). General properties of transcriptional time series in escherichia coli. *Nature genetics*, 43:554–560.

Suel, G. M., Kulkarni, R. P., Dworkin, J., Garcia Ojalvo, J., and Elowitz, M. B. (2007). Tunability and noise dependence in differentiation dynamics. *Science*, 315:1716–1719.

Sunnaker, M., Zamora Sillero, E., Dechant, R., Ludwig, C., Busetto, A. G., Wagner, A., and Stelling J (2013). Automatic generation of predictive dynamic models reveals nuclear phosphorylation as the key msn2 control mechanism. *Science signaling*, 6:ra41.

Suter, D. M., Molina, N., Gatfield, D., Schneider, K., Schibler, U., and Naef, F. (2011). Mammalian genes are transcribed with widely different bursting kinetics. *Science*, 332:472–474.

Tan, R. Z. and Oudenaarden, A. (2010). Transcript counting in single cells reveals dynamics of rdna transcription. *Mol Syst Biol.*, 6:358.

Taniguchi, Y., Choi, P. J., Li, G. W., Chen, H., Babu, M., and Hearn, J. (2010). Quantifying e. coli proteome and transcriptome with single-molecule sensitivity in single cells. *Science*, 329:533–538.

Tavare, S., Balding, D. J., Griffiths, R. C., and Donnelly P (1997). Inferring coalescence times from dna sequence data. *Genetics*, 145:505–518.

Thanh, V. H., Zunino, R., and Priami, C. (2017). Efficient stochastic simulation of biochemical reactions with noise and delays. *Journal of Chemical Physics*, 146.

Tian, T., Xu, S., Gao, J., and Burrage K (2007). Simulated maximum likelihood method for estimating kinetic rates in gene expression. *Bioinformatics*, 23:84–91.

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., and Stumpf, M. P. (2009). Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems. *J R Soc Interface*, 6:187–202.

Toni, T., Ozaki, Y., Kirk, P., Kuroda, S., and Stumpf, M. P. (2012). Elucidating the in vivo phosphorylation dynamics of the erk map kinase using quantitative proteomics data and bayesian model selection. *Mol Biosyst.*, 8:1921–1929.

Trefethen, N. L. and Bau, D. (1997). Numerical linear algebra, siam.

Van Kampen, N. G. (2007). Stochastic processes in physics and chemistry, elsevier science.

Villaverde, A. F., Bongard, S., Mauch, K., Müller, D., Balsa Canto, E., Schmid, J., and Banga JR (2015). A consensus approach for estimating the predictive accuracy of dynamic models in biology. *Computer Methods and Programs in Biomedicine*, 119:17–28.

Wang, Y., Christley, S., Mjolsness, E., and Xie X (2010). Parameter inference for discretely observed stochastic kinetic models using stochastic gradient descent. *BMC Syst Biol.*, 4.

White, J. A., Rubinstein, J. T., and Kay, A. R. (2000). Channel noise in neurons. *Trends Neurosci*, 23:131–137.

Wilkinson, D. (2006). Stochastic modelling for systems biology. *Boca Raton: Taylor and Francis: Chapman and Hall/CRC mathematical and computational biology series*.

Xu, X., Kumar, N., Krishnan, A., and Kulkarni R (2013). Stochastic modeling of dwell-time distributions during transcriptional pausing and initiation. *Decision and Control (CDC), IEEE 52nd Annual Conference on IEEE*, pages 4068–4073.

Yu, J., Xiao, J., Ren, X., Lao, K., and Xie, X. S. (2006). Probing gene expression in live cells, one protein molecule at a time. *Science*, 311:1600–1603.

Yuh, C. H., Bolouri, H., and Davidson, E. H. (1998). Genomic cis-regulatory logic: experimental and computational analysis of a sea urchin gene. *Science*, 279:1896–1902.

Zechner, C., Ruess, J., Krenn, P., Pelet, S., Peter, M., Lygeros, J., and Koeppl, H. (2012). Moment-based inference predicts bimodality in transient gene expression. *PNAS*, 109:8340–8345.

Zenklusen, D., Larson, D. R., and Singer, R. H. (2008). Single-rna counting reveals alternative modes of gene expression in yeast. *Nature Structural and Molecular Biology*, 15:1263–1271.

Zhang, J., Chen, L., and Zhou, T. (2012). Analytical distribution and tunability of noise in a model of promoter progress. *Biophys J.*, 102:1247–1257.

Zoller, B., Nicolas, D., Molina, N., and Naef, F. (2015). Structure of silent transcription intervals and noise characteristics of mammalian genes. *Molecular Systems Biology*, 11.

Zong, C., So, L., Seplveda, L. A., Skinner, S., and Golding, I. (2010). Lysogen stability is determined by the frequency of activity bursts from the fate-determining gene. *Molecular systems biology*, 6:1–12.

List of Publications

Journal Publications

[1] Keerthi S.Shetty and Annappa B. (2018). "Transcriptional processes : Models and Inference". Journal of Bioinformatics and Computational Biology, Vol.16, No.5, 1850023-1 – 1850023-17.

[2] Keerthi S.Shetty and Annappa B "Clumped-MCEM : Inference for multistep transcriptional processes". Computational Biology and Chemistry, Vol.81, 16–20.

Conference Publications

[1] Keerthi S.Shetty and Annappa B. (2017). "Inferring Transcriptional Dynamics with Time-Dependent Reaction Rates Using Stochastic Simulation ". In 5th International Conference on Advanced Computing, Networking, and Informatics(ICACNI 2017), AISC Springer 708, Vol.2, 549–556, June, Goa.

[2] Keerthi S.Shetty and Annappa B. (2016). "Efficient Sampling of Probabilistic Program for Biological Systems". In proc of ICMCB 2016: International Conference on Mathematical and Computational Biology, Vol.3, No.3, 1–5, March, France.

Brief Bio-Data

Name: Keerthi Srinivas Shetty

Address

Keerthi Shetty

Karla Constructions

A S Road

Karkala-574104

Email:keert.cs@gmail.com

Qualification

M. Tech- Software Engineering, Manipal Institute of Technology, Manipal

B.E- Computer Science and Engineering, NMAMIT, Nitte

Previous Work experience

Industry

Innovation Labs, Tata Consultancy Services, Bangalore

TransInn Technologies (MIT Student Startup)

Teaching

NMAMIT, Nitte.