Imperial College London

IMPERIAL COLLEGE LONDON

DEPARTMENT OF MATHEMATICS

MSCI RESEARCH PROJECT

Optimization for Inference of Transcriptional Kinetics from Single Cell Data

Author:
William Hilton

Supervisor(s):
Dr. Philipp Thomas, Zekai Li

Submitted in partial fulfillment of the requirements for the MSci in Mathematics at Imperial College London

June 10, 2024

Abstract

Transcription of DNA into mRNA is a key component of gene expression, the process by which cells produce proteins. Single cell sequencing methods have produced large amounts of experimental data which can be used to infer transcriptional kinetics, the rate and behaviour of transcription, but are affected by significant noise, recording only a small fraction of the true mRNA transcribed. Modelling processes as continuous time Markov chains we develop methods for inference based on optimization of stationary distributions. We investigate the performance of these methods on simulated data and then on single cell sequencing data, exploring new and existing approaches to deal with the noise present.

Acknowledgments

A great thanks to my supervisors Dr. Philipp Thomas and Zekai Li for their support and guidance throughout the project.

Plagiarism statement

The work contained in this thesis is my own work unless otherwise stated.

Signature: William Hilton
Date: June 10, 2024

Contents

Chapter 1

Introduction

Biochemical reactions are essential for life, with thousands taking place inside the cells of all living organisms to carry out their needs. A better understanding of the rates at which these reactions occur, the study of chemical kinetics, is of interest for many applications such as drug discovery in the pharmaceutical industry. Modelling systems of biochemical reactions mathematically, experimental data can be used for inference, producing estimates of kinetic parameters such as the rates of reactions.

Chemical reactions of interacting molecules are often modelled deterministically using mass-action kinetics, producing systems of differential equations that describe the evolution of the concentration of molecules in time. However, when reactions involve small numbers of molecules, such as biochemical reactions occurring inside cells, this macroscopic view breaks down and fails to reproduce stochastic effects that are observed in real systems [1].

Of particular interest is the process of gene expression, a system of chemical reactions inside cells that describes how genes, sections of DNA, produce proteins. The process involves 2 main steps: transcription and translation, in transcription DNA is copied (transcribed) to produce molecules of mRNA, then in translation the mRNA molecules are used by the cell to produce proteins [1]. Gene expression has been shown to be inherently stochastic [2, 3], with both transcription and translation affected significantly by intrinsic and extrinsic noise. Intrinsic noise describes noise sources that create differences within the same cell, such as the stochastic effects caused by low molecule numbers, whereas extrinsic noise sources create differences between cells, such as environmental effects [4].

Models of reactions such as gene expression must then include stochastic effects, as put by [5] "Stochasticity is then mandatory, and we ignore it at out own risk". These stochastic reaction networks (SRNs) are often modelled using continuous time Markov chains whose evolution in time is governed by the chemical master equation (CME). Despite a simple formulation, the CME is not solvable analytically except for simple cases as it is an infinite system of coupled equations [6]. For the 'forward problem' of solving the CME given a fully specified model, stochastic simulation algorithms such as Gillespie (Stochastic simulation algorithm) [7] can be used to simulate sample paths of the reaction network, giving a simple, albeit computationally intensive, method for approximate solutions of the CME. A wide variety of other methods have been developed to produce efficient and accurate approximate solutions of the CME which are discussed at length in [8] and [1].

However, the 'inverse problem' of parameter inference from a reaction network given observed data is still largely an open problem [8] due to the difficulty of solving the CME, an infinite system of coupled equations, as well as the significant noise present in observed data. Standard Bayesian approaches to inference are difficult to use as the likelihood generally requires a closed-form solution to the CME which is not available [9].

A popular method is Approximate Bayesian Computation (ABC) [10], a likelihood-free approach to inference where data is simulated under chosen parameters and compared to observed data to build a picture of parameter values which could have produced the observations. However, this can be computationally expensive due to the large number of simulations required, and comparisons to observed data can be difficult [11]. Another popular approach is based on the moments of the probability distribution that solves the CME. These moments satisfy an infinite system of coupled equations where higher order moments depend on lower orders, which cannot in general be solved, so approximations such as 'moment closure', where a truncation is applied to fix higher order moments to known values, are used to produce approximate solutions [12].

Advancements in single cell RNA sequencing (scRNA-seq) methods have produced "big data" on biochemical reactions such as gene expression which can be used for inference. However, there are still significant challenges as the data provides only a snapshot of the reaction in time, making it difficult to infer dynamics, and contains significant technical noise with only a small percentage of molecules being captured. Additionally, the datasets are large and sparse, often with unknown variation between batches of cells being analysed [5] presenting further problems for analysis.

We explore a method of inference based on the stationary solution to the CME which produces interval estimates for parameters of interest, allowing for uncertainty quantification. Assuming that sufficient time has passed for reactions to exhibit stationary behaviour, we use bootstrap resampling to estimate the stationary distribution of the Markov chain describing the reaction network. Using this estimate and the CME as constraints we optimize using mathematical programming to find the minimum and maximum parameter values that are consistent with the observed data.

In chapter 2 we introduce the theory behind stochastic reaction networks and discuss the setting of the inverse problem. In chapter 3 we introduce the optimization problems used to perform inference. In chapter 4 we present the results of the method, first on synthetic data simulated from several simple models, and then on observed scRNA-seq data where we explore approaches to deal with the low 'capture efficiency' of observations.

Chapter 2

Background

2.1 Stochastic Reaction Networks

Stochastic Reaction Networks (SRN) model the interactions between populations of molecular species, such as proteins and chemical substances. They describe a set of m reactions R_j between n species S_i:

$$R_j : v_{1j}^{-} S_1 + \cdots + v_{nj}^{-} S_n \xrightarrow{k_j} v_{1j}^{+} S_1 + \cdots + v_{nj}^{+} S_n \quad , \quad j = 1, \ldots, m \tag{2.1}$$

where $v_{ij}^{\pm} \in \mathbb{N}$ are the stoichiometric coefficients that give the number of molecules of species S_i which are produced $(+)$ and consumed $(-)$ by reaction R_j. We also define the stoichiometric vector:

$$v_j = (v_{1j}^{+} - v_{ij}^{-}, \ldots, v_{nj}^{+} - v_{nj}^{-}) \in \mathbb{Z}^n \tag{2.2}$$

which describes the net change in the molecule number of each species after reaction R_j [6, 13].

Assuming a system of ’well-mixed’ molecules which react upon collision, the probability for a given combination of reactants to react in the time interval $[t, t + dt)$ is proportional to dt[8], with the constant of proportionality defined as the reaction rate constant $k_j \in [0, \infty)$. Multiplying by the number of possible combinations of reactant molecules in the system at time t (0 if there are not enough) gives the probability for any combination of reactants to react [14]: $\mathbb{P}(\text{Reaction } R_j \text{ occurs in } [t, t + dt)) \propto dt \times$ # reactant combinations. We thus define the propensity function $a_j : \mathbb{N}^n \to [0, \infty)$ which maps from the current state of the system to the propensity, or rate, at which reaction R_j occurs as:

$$a_j(x) = k_j \prod_{i=1}^{n} \binom{x_i}{v_{ij}^{-}} \tag{2.3}$$

A simple example of a reaction system is a birth-death model [13]:

$$\varnothing \xrightarrow{k_1} X \qquad X \xrightarrow{k_2} \varnothing \tag{2.4}$$

The first reaction produces molecules of X from an external source, there are no reactants so the propensity function is simply the reaction rate constant: $a_1(x) = k_1$. The second reaction degrades molecules of X, the reactants are a single molecule of X and so the propensity function is proportional to the amount available: $a_2(x) = k_2 x$.

The state of the system at time t is defined as the discrete random variable $X(t) = (X_1(t), \ldots, X_n(t)) \in \mathbb{N}^n$ which describes the number of molecules of each species S_i at time t and takes values in the state space of the system $S \subseteq \mathbb{N}^n$ [6]. This set usually has infinite cardinality $|S|$, with no limit on the number of molecules, but some species may be constrained to finitely many states such as a switch restricted to the states $\{\text{on}, \text{off}\}$.

The system is modelled as a continuous time Markov chain, with transition rate from state x to state y defined as:

$$q(x,y) = \sum_{j=1}^{m} a_j(x)(1_{x+v_j}(y) - 1_x(y)) \quad , \quad x, y \in S \tag{2.5}$$

where $1_x(y)$ is the indicator function of $x = y$. The transition rate matrix of the chain is then defined as $Q = (q(x,y))_{x,y \in S}$, and can be shown to be totally stable and conservative [13], satisfying the properties:

$$q(x,y) \geq 0 \quad , \quad \forall x \neq y \tag{2.6}$$

$$-q(x,x) = \sum_{y \in S, y \neq x} q(x,y) < \infty \quad , \quad \forall x \in S \tag{2.7}$$

We further assume that Q is regular: the chain cannot leave every finite subset of the state space S in a finite amount of time, meaning in our case that molecule numbers do not reach infinity within finite time [13].

2.1.1 The Chemical Master Equation

We define $p_t(x) = \mathbb{P}(X(t) = x)$ as the probability that the chain is in state $x \in S$ at time $t \geq 0$, and $p_t := (p_t(x))_{x \in S}$ as the **column** vector of the distribution of the chain at time t. Under the assumption that Q is regular, p_t is the only probability distribution solution to the Chemical Master Equation (CME):

$$\frac{dp_t}{dt} = Qp_t \quad , \quad p_0 = \lambda \tag{2.8}$$

where $\lambda = (\lambda(x))_{x \in S}$ is any initial probability distribution of the chain [13].

A probability distribution $p = (p(x))_{x \in S}$ on S is called a stationary distribution of the chain if, once setting the initial distribution to p: $p_0 = p$, the chain remains distributed according to p for all later times: $p_t = p\,, \forall t \geq 0$. Since p does not depend on the time t we have that $\frac{dp}{dt} = 0$ and so:

$$Qp = 0 \tag{2.9}$$

Any distribution satisfying the above property is called a stationary solution of the CME and, under the assumption of regularity, a distribution p is a stationary solution if and only if it is a stationary distribution [13], so we will refer to both properties as 'stationary'. Under further assumptions the stationary distribution determines the long-term behaviour of the chain. For an ergodic Markov chain the distribution p_t converges to the unique stationary distribution p in total variation $\lim_{t\to\infty} \|p_t - p\|_{TV} = 0$ [13], where the total variation distance between two probability distributions is defined as $\|p - q\|_{TV} := \sup_{A \subseteq S} |p(A) - q(A)|$

We will assume that all Markov chains considered are regular and ergodic and so converge to a unique stationary distribution p.

2.2 The Inverse Problem

In the inverse problem we have observations of the state of a system of reactions and modelling the system as a stochastic reaction network our aim is to infer the values of parameters of the system. We focus on model calibration rather than model selection [8], assuming that the stoichiometric coefficients $v_{ij}^{\pm}$ which determine the structure of the system are fixed and estimating the values of kinetic parameters such as reaction rate constants k_j. In this setting we assume the SRN is an accurate model of the observed reaction system and search for the parameter values which could have produced the observed data.

Advancements in single cell sequencing techniques have produced "big data" on the processes and reactions that take place inside cells which can be used for inference [5]. In particular, Single-cell RNA sequencing (scRNA-seq) is a method used to measure the RNA content of a single cell which can be used to investigate the process of transcription in gene expression. However, measuring such tiny amounts of biological material is challenging and the method records only a small fraction of the true material, a 'capture efficiency' often as low as 6%, leading to sparse data with many zero values (dropouts) and additional variation between batches of cells being analysed [15]. We discuss methods to correct for capture efficiency, dropout and batch effects when working with scRNA-seq data in Chapter 4.

Ideally we would be able to observe the state of the reaction system over a period of time, obtaining a trajectory $\{x(t) \,|\, t_1 \leq t \leq t_2\}$ that contains information about the dynamics of the system. However, standard scRNA-seq methods produce only a snapshot of the reaction system in time, an observation of the state $x(t)$ at a single time t which may not be the same across observations of multiple cells [5].

Our approach is to consider the long-term behaviour of the reaction network. Assuming that the SRN converges to its stationary distribution p, after a sufficiently long time T we can consider the distributions p_t s.t. $t \geq T$ as approximations of p. Observations of the reaction network at or after time T can then be treated as approximate samples from p and used to compute empirical distribution estimates. Under the assumption of regularity we also have that $Qp = 0$, where the elements of the rate matrix Q are functions of the reaction rate constants and other parameters of interest. The system of equations $Qp = 0$ and an estimate of p then provide the constraints for an optimization problem to find the set of parameter values which could have produced the observations under the model.

2.2.1 The Bootstrap

We will use Bootstrap resampling to produce interval estimates for the entries of the stationary distribution $p = (p(x))_{x \in S}$. Given n observations $x_1, \dots, x_n$ assumed to be independent samples from the stationary distribution p we compute a 95% bootstrap percentile confidence interval (CI) for each $p(x)$ [16]. For each bootstrap sample $\{x_i^b\}_{i=1}^n$ and state $x \in S$ we use the empirical estimate of $p(x)$ given by:

$$\hat{p}_b(x) = \frac{\text{\# occurrences of x in sample}}{n} = \frac{1}{n} \sum_{i=1}^{n} 1_{\{x_i^b = x\}} \tag{2.10}$$

Which is simply the proportion of observations of the state x in the sample. The bootstrap algorithm is outlined in 1.

```
Algorithm 1: Bootstrap algorithm
  Input:      $\{x_i\}_{i=1}^n$, i.i.d sample from $p$
              $B$, number of bootstrap resamples
              $S$, state space
  Output:     Confidence interval estimates of $p(x)$ for each $x \in S$
  Procedure:
  for $b = 1, \ldots, B$ do
      Samples n times with replacement from $\{x_i\}_{i=1}^n$ for bootstrap sample $\{x_i^b\}_{i=1}^n$
      Compute $\hat{p}_b(x)$ for every $x \in S$ using the bootstrap sample $\{x_i^b\}_{i=1}^n$
  for $x \in S$ do
      Compute $\hat{p}_L(x)$ and $\hat{p}_U(x)$, the 2.5% and 97.5% percentiles of the estimates
      $\{\hat{p}_b(x)\}_{b=1}^B$
  return $[\hat{p}_L(x)\ ,\ \hat{p}_U(x)]$ for each $x \in S$
```

Chapter 3

Methods

3.1 Optimization

We now introduce the optimization problems used to find the set of reaction rate constants which could have produced the observed data under a given reaction network model. As described, given observations of the full state of the system $x \in S$ we use the bootstrap to compute 95% confidence intervals $[\hat{p}_L(x), \hat{p}_U(x)]$ for the stationary distribution $p = (p(x))_{x \in S}$, written in vector form as $\hat{p}_L \leq p \leq \hat{p}_U$. These bounds together with the system of equations $Qp = 0$ give constraints on the values of the reaction rates k_j we wish to infer.

However, the rate matrix Q has dimensions $|S| \times |S|$ and p has dimension $|S|$, where $|S|$ is the usually infinite cardinality of the state space S. Given a finite number of observations $\{x_i\}_{i=1}^n$ we can bound only finitely many entries of p, and cannot work with the infinite system of equations $Qp = 0$ when optimizing numerically.

To address this issue we truncate the state space to a finite subset of states $E \subset S$, working with the truncated distribution $\{p(x)\}_{x \in E}$ and equations of $Qp = 0$ involving only those terms. In practice this amounts to choosing which equations from the infinite system $Qp = 0$ to include as constraints, and we will refer to this finite system of chosen equations as $Q_E p = 0$ (or simply as $Qp = 0$). Additionally, while $\{p(x)\}_{x \in S}$ is a probability distribution over S, the truncation $\{p(x)\}_{x \in E}$ is not necessarily a distribution over E. Combining these conditions we obtain the constraint set $\mathcal{C}$, illustrated in figure 3.1(a):

$$\mathcal{C} = \left\{ \begin{array}{c|c} & Q_E p = 0 \\ p \geq 0, k_j \geq 0 & \\ & \displaystyle\sum_{x \in E} p(x) \leq 1 \\ j = 1, \ldots, m & \\ & \hat{p}_L \leq p \leq \hat{p}_U \end{array} \right\} \tag{3.1}$$

Optimizing for the minimum and maximum values of each reaction rate k_j subject to the constraint set $\mathcal{C}$ produces the solution set:

$$\hat{\Theta} = \left[\min_{\mathcal{C}} k_1, \max_{\mathcal{C}} k_1\right] \times \cdots \times \left[\min_{\mathcal{C}} k_m, \max_{\mathcal{C}} k_m\right] \tag{3.2}$$

This is an estimate of the set of parameters which could have produced the observed data under the model, with statistical error guarantees due to the 95% bootstrap confidence intervals used. In general, the estimate $\hat{\Theta}$ will not give the true set of parameters that satisfy $\mathcal{C}$: $\Theta = \{k_j \mid k_j \in \mathcal{C}\}$ since it does not account for the dependence between the reaction rates, but as illustrated by 3.1(b), satisfies $\Theta \subseteq \hat{\Theta}$.

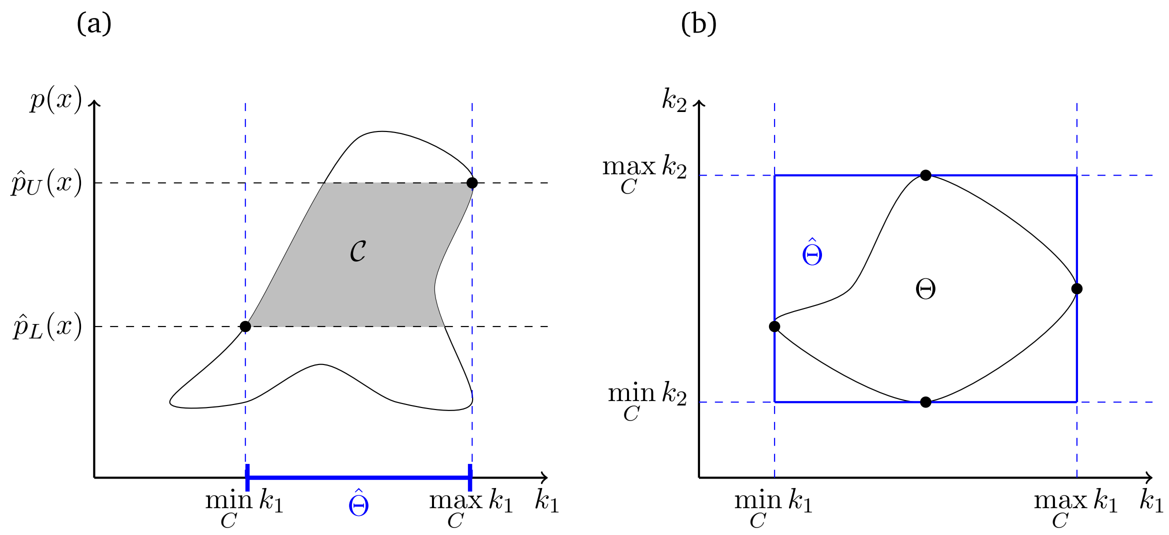


Figure 3.1 Illustrations of a constraint set and solution set for the optimization problem. (a) illustrates the constraint set $\mathcal{C}$ in a 2 dimensional space of stationary distribution value $p(x)$ and reaction rate k_1. The equations $Qp = 0$ and conditions on p constrain the variables to the outlined region and, on observing data, the interval bounds $[\hat{p}_L(x), \hat{p}_L(x)]$ on $p(x)$ restrict to the shaded constraint set $\mathcal{C}$. The minimum and maximum values of k_1 within this set, marked in blue, produce the solution set $\hat{\Theta}$ of values of k_1 that are consistent with the observed data under the model. (b) illustrates how a solution set $\hat{\Theta} = \left[\min_{\mathcal{C}} k_1, \max_{\mathcal{C}} k_1\right] \times \left[\min_{\mathcal{C}} k_2, \max_{\mathcal{C}} k_2\right]$, outlined by the blue box, may in general differ from the true set of values $\Theta = \{k_1, k_2 \in \mathcal{C}\}$ outlined by the black curve.

3.1.1 Linear Programming

The entries of the rate matrix Q are linear in the parameters k_j so the the constraint $Qp = 0$ can be written as the sum $\sum_{j=1}^{m} k_j Q_j p = 0$ for constant matrices Q_j. Since k_j and p are both variables in the optimization problem this is a quadratic equality constraint and so optimization over the constraint set $\mathcal{C}$ is a non-convex problem which are difficult and slow to solve in general.

By introducing the variables $z_j(x) = k_j p(x)$, written in vector form as $z_j = k_j p$, we can obtain a set of linear equality and inequality constraints:

$$\mathcal{C}_{\text{linear}} = \left\{ \begin{array}{l|l} z_j \geq 0, k_j \geq 0 & \displaystyle\sum_{j=1}^{m} Q_j z_j = 0 \\ j = 1, \ldots, m & \displaystyle\sum_{x \in E} z_j(x) \leq k_j \\ & k_j \hat{p}_L \leq z_j \leq k_j \hat{p}_U \end{array} \right\} \tag{3.3}$$

Since the objective function and constraints are linear in the variables k_j and z_j this is a linear program (LP), a special case of convex optimization which can be solved efficiently using standard methods such as the simplex algorithm to produce global optima [17].

However, moving from $\mathcal{C}$ to $\mathcal{C}_{\text{linear}}$ we relax the constraints of the problem as the non-linear equality constraint $z_j = k_j p$ cannot be included in the linear program. As such, the linear program does not enforce linear dependence between the vector variables z_j i.e. that the matrix $Z = [z_1, \ldots, z_m]$ has rank 1, so there is no guarantee that solutions will satisfy this constraint. In practice, given a solution the rank of Z can be computed to assess how close the constraint is to being satisfied.

3.1.2 Marginal observations

In some situations we may not observe the full state of the system $x = (x_1, \dots, x_n) \in S$ but only some of the species, for example consider production of a molecule that is controlled by an on / off switch where we observe the number of molecules but not the state of the switch. Without loss of generality assume we observe the first r species of n total and define S_i as the state space of each species i so that $S := S_{1:n} = S_1 \times \dots \times S_n$ is the full state space and we observe values in $S_{1:r}$.

Given observations we can bootstrap to produce confidence interval bounds on the marginal of the stationary distribution $p(x_1, \dots, x_r)$, and using the law of total probability relate them to the (joint) stationary distribution $p(x_1, \dots, x_n)$:

$$\mathbb{P}(X_1 = x_1, \dots, X_r = x_r) = \sum_{(x_{r+1}, \dots, x_n) \in S_{r+1:n}} \mathbb{P}(X_1 = x_1, \dots, X_n = x_n) \tag{3.4}$$

Defining the column vector of the marginal distribution $\bar{p} = (p(x_1, \dots, x_r))_{(x_1, \dots, x_r) \in S_{1:r}}$, and analogously the vector of the joint distribution p, we can write this as $\bar{p} = Ap$, where A is a matrix with entries $a_{ij} \in \{0, 1\}$ representing the sum. The confidence interval bounds $\bar{p}_L \leq \bar{p} \leq \bar{p}_U$ on the marginal then give the constraints:

$$\bar{p}^L \leq Ap \leq \bar{p}^U \quad , \quad k_j \bar{p}^L \leq Az_j \leq k_j \bar{p}^U \tag{3.5}$$

which take the place of the existing bounds in $\mathcal{C}$ and $\mathcal{C}_{\text{linear}}$

3.1.3 State space truncation

Introducing a state space truncation relaxes the constraints of the optimization problem by reducing an infinite set of constraints to a finite set and so will widen the solution bounds produced. The choice of truncation E is therefore crucial and we will investigate how it affects solutions in various examples.

In general, a larger E gives tighter solution bounds at the cost of longer computation time due to the larger number of constraints. However, using too large of a truncation risks using CI estimates based on small numbers of observations which can lead to infeasible problems or inaccurate solutions.

3.1.4 Implementation

We use GUROBI [18] to compute globally optimal solutions to non-convex optimization problems (NLP) using the python API, and the python package cvxpy [19] to solve linear programs (LP) using a selection of solvers (including GUROBI).

Chapter 4

Results

4.1 Synthetic data

Since single cell sequencing data is affected by significant noise that complicates analysis we will first work with simulated data to investigate our methods. We simulate stochastic reaction networks by using the Stochastic Simulation Algorithm (SSA) developed by Gillespie [7] to generate exact sample paths, realizations of the state of the system up to time T which at time t are exact samples from the distribution p_t [14]. Taking the final state of the sample path $x(T)$ we simulate observations of real cells and, as discussed, for T sufficiently large we can take these values as approximate samples from the stationary distribution p.

We use the Gillespie direct method which simulates the exponentially distributed holding time between reactions and the chance of each reaction to occur:

1. Initialize time $t = 0$ and state $x = x_0$
2. compute transition rates $\{q(x, x + v_j)\}_{j=1}^{m}$
3. simulate holding time until next reaction $\Delta t \sim \text{Exp}(-q(x, x))$
4. sample reaction j with probability $\mathbb{P}(j) = -\frac{q(x,x+v_j)}{q(x,x)}$ for $j = 1, \ldots, m$
5. update time $t \to t + \Delta t$ and state $x \to x + v_j$
6. repeat until $t > T$
7. return final state

This is a simple and exact simulation method but is computationally expensive, becoming infeasible for large reaction systems since every step must be simulated [1]. Many, more computationally efficient, methods have been developed for simulating SRNs and the forward problem of solving the CME which are discussed at length in [1, 8, 14], but they often sacrifice accuracy for performance. Since we are interested in solving the inverse problem, the direct method is sufficient for producing accurate simulated data from simple reaction networks which can then be used for inference.

4.1.1 Full observations

In the setting where we observe the full state of the system we consider one of the simplest reaction networks, a birth-death process [13]. A birth-death process is a single species reaction network with 2 reactions that model the creation ’birth’ and destruction ’death’ of molecules:

$$\varnothing \xrightarrow{k_1} X$$
$$X \xrightarrow{k_2} \varnothing$$

The stoichiometric coefficients are $v_{11}^{-} = 0$, $v_{11}^{+} = 1$, $v_{12}^{-} = 1$, $v_{12}^{+} = 0$ giving the net change of molecules in each reaction $v_1 = (1)$ and $v_2 = (-1)$. The state space is $S = \mathbb{N}$ and for a number of molecules $x \in S$ the propensity functions of each reaction are:

$$a_1(x) = k_1 \binom{x}{0} = k_1 \tag{4.1}$$

$$a_2(x) = k_2 \binom{x}{1} = k_2 x \tag{4.2}$$

The first reaction models production from an external source using no reactants so occurs at a constant rate. The second reaction models the consumption of molecules by an external source with 1 molecule of X as a reactant, so occurs at a rate proportional to the current number of molecules $x \in S$.

The transition rates of the chain are given by:

$$\begin{aligned}
q(x,y) &= \sum_{j=1}^{2} a_j(x)(1_{x+v_j}(y) - 1_x(y)) \\
&= k_1(1_{x+1}(y) - 1_x(y)) + k_2 x(1_{x-1(y)} - 1_x(y)) \\
&= \begin{cases} k_1, & y = x+1 \\ k_2 x, & y = x-1 \\ -(k_1 + k_2 x), & y = x \\ 0, & \text{otherwise} \end{cases}
\end{aligned}$$

Giving the rate matrix of the chain $Q = (q(x,y))_{x \in S}$:

$$\begin{aligned}
Q &= \begin{bmatrix} q(0,0) & q(1,0) & \cdots & q(x,0) & \cdots \\ q(0,1) & q(1,1) & \cdots & q(x,1) & \cdots \\ \vdots & \vdots & \ddots & & \\ q(0,y) & q(1,y) & & q(x,y) & \cdots \\ \vdots & \vdots & & \vdots & \ddots \end{bmatrix} \\
&= \begin{bmatrix} -k_1 & k_2 & 0 & 0 & \cdots \\ k_1 & -(k_1+k_2) & 2k_2 & 0 & \cdots \\ 0 & k_1 & -(k_1+2k_2) & 3k_2 & \\ 0 & 0 & k_1 & \ddots & \ddots \\ \vdots & \vdots & & \ddots & \end{bmatrix} \\
&= k_1 \begin{bmatrix} -1 & & & \\ 1 & -1 & & \\ & 1 & -1 & \\ & & \ddots & \ddots \end{bmatrix} + k_2 \begin{bmatrix} 0 & 1 & & & \\ 0 & -1 & 2 & & \\ & & -2 & 3 & \\ & & & \ddots & \ddots \end{bmatrix} \\
&:= k_1 Q_1 + k_2 Q_2
\end{aligned}$$

and so the constraint sets:

$$\mathcal{C} = \left\{ p \geq 0, k_1, k_2 \geq 0 \,\middle|\, \begin{array}{l} k_1 Q_1 p + k_2 Q_2 p = 0 \\ \sum_{x \in E} p(x) \leq 1 \\ \hat{p}_L \leq p \leq \hat{p}_U \end{array} \right\} \tag{4.3}$$

$$\mathcal{C}_{\text{linear}} = \left\{ z_1, z_2 \geq 0, k_1, k_2 \geq 0 \,\middle|\, \begin{array}{l} Q_1 z_1 + Q_2 z_2 = 0 \\ \sum_{x \in E} z_j(x) \leq k_j \\ k_j \hat{p}_L \leq z_j \leq k_j \hat{p}_U \end{array} \right\} \tag{4.4}$$

Where $[\hat{p}^L, \hat{p}^U]$ are vector bounds on the stationary distribution p computed using the bootstrap on observations of the reaction network after a sufficient time T > 0. As discussed, we truncate the state space to a finite subset $E \subset \mathbb{N}$ where in practice we choose equations of $Qp = 0$ to include as constraints and define the truncation by the states used. The tri-diagonal form of Q means that including equation $n > 0$ we have $\{n-1, n, n+1\} \subseteq E$, and $\{0, 1\} \subseteq E$ for $n = 0$.

However, an important detail to consider is identifiability of the parameters being estimated. The true stationary distribution p of a birth-death process is Poisson(k_1/k_2) distributed [20] which only depends on the ratio between the reaction rates k_1 and k_2. Given observations of p it is not possible to identify the values of the parameters k_1 and k_2, only their ratio is identifiable. To address this we choose to fix $k_2 = 1$, setting the ratio equal to k_1 which then represents the relative balance between birth and death reaction rates. This problem of identifiability occurs in many reaction networks where the stationary behaviour only depends on, and so only gives information about, the ratio between parameters whose individual values then cannot be identified.

Figure 4.1 shows the bounds on the parameter k_1 produced by the linear (LP) and non-linear (NLP) methods when applied to a sample of size n = 1000 simulated from a birth-death reaction network. Both methods produce interval bounds on k_1 which contain the true value and reduce in width as the number of constraints is increased. Initially, adding constraints to the optimization significantly reduces the solution width but the improvements slow until both bounds reach a constant width that does not improve despite additional constraints. This shows that truncating to a finite set of constraints provides enough information to produce good bounds, and suggests that many equations of $Qp = 0$ give the same information and so are redundant.

The LP is an 'outer approximation' of the NLP, with the NLP bounds always closer to the true value than the LP bounds, which is expected since the linear constraint set $\mathcal{C}_{\text{linear}}$ is a relaxation of the non-linear set $\mathcal{C}$, dropping the constraints $z_j = k_j p$. We see that even for large N the LP produces strictly worse results and can check that these optimal solutions do not satisfy the constraints $z_j = k_j p$ and so do not belong to $\mathcal{C}$.

However, the LP is more computationally efficient than the NLP, being significantly faster to compute especially for larger numbers of constraints, a trade off for the wider solution bounds. The diminishing returns and increased computation time when increasing the number of constraints used suggests an optimal balance between results and efficiency, around N = 7 in 4.1. Solving large numbers of optimization problems when working with single cell data in chapter 4 this balance will be vital, and we will use heuristics to choose a truncation.

While the input data is exactly simulated from a birth-death reaction with parameters $k_1 = 5$ and $k_1 = 1$, due to the finite sample size n = 1000 there are more parameters than a single point that are consistent with the observed data (even when fixing $k_2 = 1$ for identifiability)

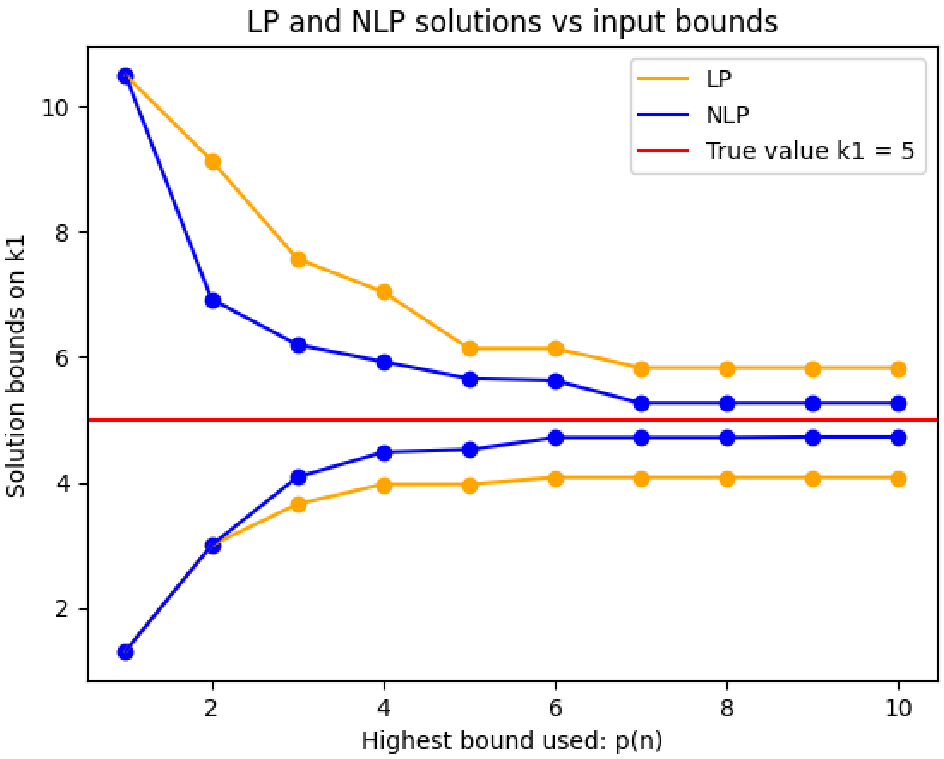


Figure 4.1 Parameter bounds for the Birth-Death model against number of constraints used. Upper and lower bounds on the rate parameter k_1 computed by optimizing over the linear and non-linear constraint sets $\mathcal{C}_{\text{linear}}$ (orange) and $\mathcal{C}$ (blue). Plotted against the number of equations of $Qp = 0$ used as constraints: starting with the 1st equation, then the 1st and 2nd equations and so on, adding one equation each time, equivalently adding to the state space truncation $E = \{0, 1\}$ one state at a time. The red line indicates the true value of the parameter k_1, with the data used n = 1000 samples simulated from a model with parameters $k_1 = 5$ and $k_2 = 1$.

and so are bounded by the LP and NLP. However, as the sample size increases it can be shown that the bounds produced by both methods approach the single point $k_1 = 5$.

4.1.2 Marginal observations

To investigate the setting of marginal observations we consider the 'telegraph model' [5], a popular model of transcription in gene expression describing the bursty production of mRNA:

$$G_{off} \underset{k_{off}}{\overset{k_{on}}{\rightleftharpoons}} G_{on}$$

$$G_{on} \xrightarrow{k_{tx}} G_{on} + M$$

$$M \xrightarrow{k_{deg}} \varnothing$$

A gene switches between an on-state G_{on} in which transcription of mRNA occurs, and an off-state G_{off} in which no transcription occurs. This leads to a pattern of 'transcriptional bursting' [21] where periods of mRNA production are separated by periods of inactivity. We assume that the mRNA content can be observed but not the state of the gene G and aim to estimate the rate of transcription k_{tx}. The on and off rates k_{on} and k_{off} are typically unknown, but the degradation rate of mRNA k_{deg} will be fixed for identifiability.

The state of the system is the tuple (m, g) where $m \in \mathbb{N}$ is the number of mRNA molecules and $g \in \{0, 1\}$ describes the state of the gene with 0 corresponding to G_{off} and 1 to G_{on}, giving the state space $S = \mathbb{N} \times \{0, 1\}$. Marginal observations mean we observe a sample of values of m, and applying the bootstrap can compute confidence interval bounds on the values of the marginal stationary distribution $p(m)$: $\bar{p}_L(m) \leq p(m) \leq \bar{p}_U(m)$. Since $p(m) = p(m, 0) + p(m, 1)$ we then have bounds on sums of pairs of the joint stationary distribution, written in vector form as:

$$\bar{p} = \begin{bmatrix} p(0) \\ p(1) \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 & 1 & & & \\ & & 1 & 1 & \\ & & & & \ddots \end{bmatrix} \begin{bmatrix} p(0,0) \\ p(0,1) \\ p(1,0) \\ p(1,1) \\ \vdots \end{bmatrix} = Ap \tag{4.5}$$

Bounds on the marginal $\bar{p}_L \leq \bar{p} \leq \bar{p}_U$ then give the constraints:

$$\bar{p}_L \leq Ap \leq \bar{p}_U \tag{4.6}$$

$$k_j \bar{p}_L \leq Az_j \leq k_j \bar{p}_U \tag{4.7}$$

For $z_{on} := k_{on}p$, $z_{off} := k_{off}p$, $z_{tx} := k_{tx}p$, $z_{deg} := k_{deg}p$. Reading off the stoichiometric coefficients from the reaction we can compute the transition rates of the chain:

$$\begin{aligned} q((m,g),(m,g+1)) &= k_{on}(1-g) \\ q((m,g),(m,g-1)) &= k_{off}g \\ q((m,g),(m+1,g)) &= k_{tx}g \\ q((m,g),(m-1,g)) &= k_{deg}m \\ q((m,g),(m,g)) &= -(k_{on}(1-g) + k_{off}g + k_{tx}g + k_{deg}m) \\ q((m,g),(x,y)) &= 0, \quad \text{otherwise} \end{aligned}$$

These define the rate matrix $Q = (q(x, y))_{x,y \in S}$, and since the entries are linear in the reaction rate parameters we can decompose Q into a sum over constant matrices:

$$Q = k_{on}Q_{on} + k_{off}Q_{off} + k_{tx}Q_{tx} + k_{deg}Q_{deg} \tag{4.8}$$

The stationary distribution of the mRNA $p(m)$ is a Beta-Poisson mixture distribution [20]

$$m|\rho \sim \text{Poisson}\left(\frac{k_{tx}}{k_{deg}}\rho\right) \qquad \rho \sim Beta\left(\frac{k_{on}}{k_{deg}}, \frac{k_{off}}{k_{deg}}\right) \tag{4.9}$$

where beta-distributed gene activity ρ controls the Poisson-distributed production of mRNA. The stationary distribution of the gene $p(g)$ is given by:

$$p(g) = \begin{cases} \frac{k_{off}}{k_{on}+k_{off}}, & g = 0 \\ \frac{k_{on}}{k_{on}+k_{off}}, & g = 1 \end{cases} \tag{4.10}$$

Where the proportion of time spent in the on / off state is equal to the relative balance between k_{on} and k_{off}. These distributions are invariant under re-scaling of the reaction rates k_j, so we fix $k_{deg} = 1$ to ensure identifiability of the parameters k_{on}, k_{off} and k_{tx} given observations from $p(m)$.

Since we are only able to bound sums of pairs of the stationary distribution $p(m, g)$ we expect that the solution bounds produced by optimization will be wider than if bounds on individual entries were available. However, additional constraints can be derived using the 'Fréchet bounds' [22]:

$$\max\left\{0, \sum_{k=1}^{n} \mathbb{P}(A_k) - (n-1)\right\} \leq \mathbb{P}\left(\bigcap_{i=1}^{n} A_k\right) \leq \min_k \{\mathbb{P}(A_k)\} \tag{4.11}$$

When applied to the events $\{M = m\}$ and $\{G = g\}$ we obtain:

$$\max\{0, p(m) + p(g) - 1\} \leq p(m, g) \leq \min\{p(m), p(g)\} \tag{4.12}$$

Several constraints can be derived which are redundant given the base constraints, but a tighter bound is given by the inequality:

$$p(m, g) \leq p(g) \quad g \in \{0, 1\} \tag{4.13}$$

Multiplying by $(k_{on} + k_{off})$ we obtain the constraints, written in vector form:

$$(k_{on} + k_{off})p \leq kv \tag{4.14}$$

$$z_{on} + z_{off} \leq kv \tag{4.15}$$

For $kv := \begin{bmatrix} k_{off} & k_{on} & k_{off} & k_{on} & \cdots \end{bmatrix}^T$. When k_{on} and k_{off} are fixed / known we can go even further and bound the remaining z_j in the LP:

$$z_{on} = \left(\frac{k_{on}}{k_{off}}\right) z_{off} \tag{4.16}$$

$$(k_{on} + k_{off})z_{tx} \leq k_{tx} kv \tag{4.17}$$

$$(k_{on} + k_{off})z_{deg} \leq k_{deg} kv \tag{4.18}$$

Putting everything together we get the constraint sets:

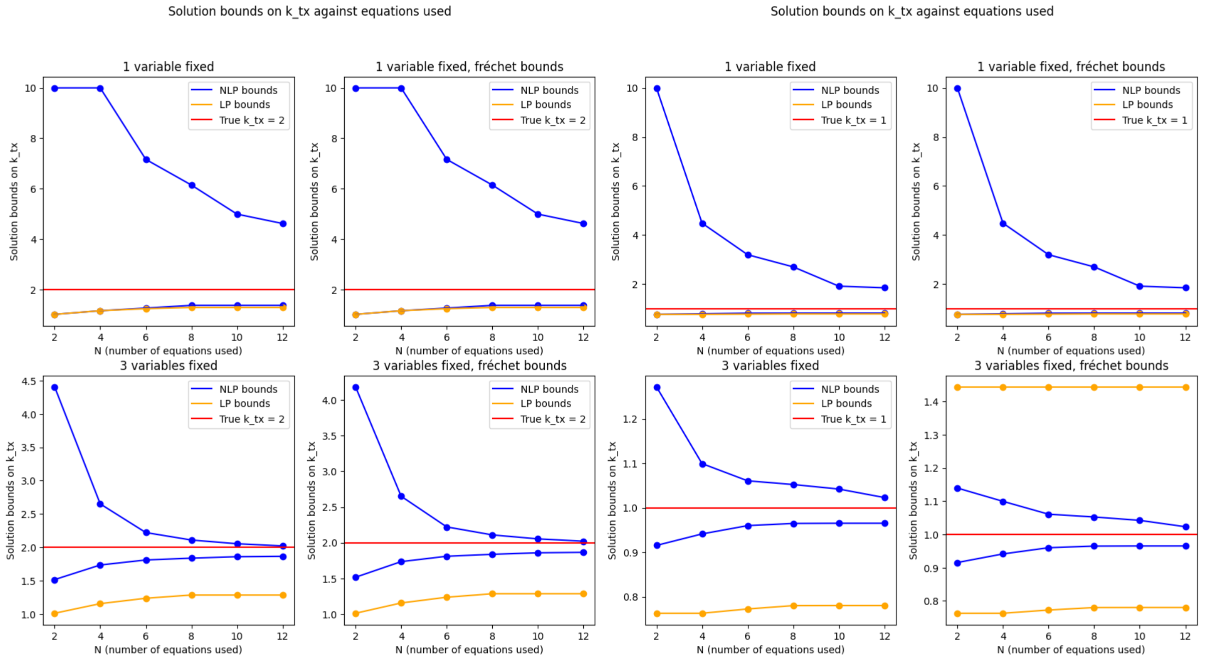
$$\mathcal{C} = \left\{ \begin{matrix} p \geq 0 \\ k_j \geq 0 \quad j \in J \end{matrix} \;\middle|\; \begin{matrix} \sum_{j \in J} k_j Q_j p = 0 \\ \sum_{(m,g) \in E} p(m, g) \leq 1 \\ \bar{p}_L \leq Ap \leq \bar{p}_U \\ (k_{on} + k_{off})p \leq kv \end{matrix} \right\} \tag{4.19}$$

$$\mathcal{C}_{\text{linear}} = \left\{ \begin{matrix} z_j \geq 0 \\ k_j \geq 0 \quad j \in J \end{matrix} \;\middle|\; \begin{matrix} \sum_{j \in J} Q_j z_j = 0 \\ \sum_{(m,g) \in E} z_j(m, g) \leq k_j \quad j \in J \\ k_j \bar{p}_L \leq A z_j \leq k_j \bar{p}_U \quad j \in J \\ z_{on} + z_{off} \leq kv \end{matrix} \right\} \tag{4.20}$$

For the index set $J = \{on, off, tx, deg\}$ and a chosen state space truncation $E \subset S = \mathbb{N} \times \{0, 1\}$ where we only truncate the state space of M, since the state space of G is already finite. We include the additional bounds derived when k_{on} and k_{off} are fixed.

Figure 4.2 shows the bounds produced on k_{tx} when optimizing over the constraint sets $\mathcal{C}$ and $\mathcal{C}_{\text{linear}}$ using data simulated from 2 different telegraph models.

As with the birth-death model in figure 4.1 we see the LP is an outer approximation of the NLP, and the width of the solution bounds for both methods decreases as more constraints are used, although the changes are small for the LP with only slight improvement in the lower bounds. Despite using a larger sample of n = 5000 observations compared to n = 1000 in 4.1 the bounds for both methods are wider, showing that marginal observations can give much less information about the parameters of a reaction compared to observations of the full state.



(a) $k_{on} = 2$, $k_{off} = 1$, $k_{tx} = 2$, $k_{deg} = 1$.

(b) $k_{on} = 5$, $k_{off} = 1$, $k_{tx} = 1$, $k_{deg} = 1$.

Figure 4.2 Solution bounds on $\mathbf{k_{tx}}$ against number of constraints used for the telegraph model. Upper and lower bounds computed by optimizing over $\mathcal{C}$ (NLP in blue) and $\mathcal{C}_{\text{linear}}$ (LP in orange) plotted against the number of equations of $Qp = 0$ used as constraints, starting with the 1st and 2nd equations and adding 2 at a time. Sub-figure (a) shows solutions using, as input data, a sample of n = 5000 simulated from a telegraph model with parameters $\{2, 1, 2, 1\}$ and (b) a model with parameters $\{5, 1, 1, 1\}$. The 4 plots for each show the results when: only k_{deg} fixed (top left), k_{deg} fixed and fréchet bounds used (top right), k_{on}, k_{off} and k_{deg} fixed (bottom left), k_{on}, k_{off} and k_{deg} fixed and fréchet bounds used (bottom right).

In all except the bottom right plot 4.2b the LP does not produce an upper bound: the maximization problem for for k_{tx} under the constraints $\mathcal{C}_{\text{linear}}$ is unbounded, showing that the linear relaxation of $\mathcal{C}$ may not retain enough information in the setting of marginal observations.

The true parameter values can strongly influence the performance of both methods as seen by the differences between 4.2a and 4.2b: both methods perform better in 4.2b and more generally when k_{on} is large relative to k_{off} since the model is 'closer' to a birth-death process (gene always active, alleviating the problem of marginal observations) and the fréchet bounds are tighter. When the values of k_{on} and k_{off} are fixed (set as constants in the optimization), as in the bottom 2 plots of 4.2a and 4.2b, the extra information significantly reduces the width of the solution bounds. Adding fréchet bounds does not give a noticeable improvement, except for the case in the bottom right plot of 4.2b where the LP achieves a finite upper bound.

4.1.3 Bursting model

When k_{off} is large the telegraph model reduces to a model of random bursts [20]:

$$\varnothing \xrightarrow{k_{on}} b_{tx}\text{mRNA}, \quad b_{tx} \sim \text{Geometric}\left(\frac{1}{\bar{b}_{tx}}\right)$$

$$\text{mRNA} \xrightarrow{k_{deg}} \varnothing$$

where we define the transcriptional burst frequency $\bar{a}_{tx} = \frac{k_{on}}{k_{deg}}$ and transcriptional burst size $\bar{b}_{tx} = \frac{k_{tx}}{k_{off}}$. The model describes gradual degradation of mRNA with sudden bursts of transcription, as if the gene briefly switched to its on state. As discussed later, many genes have been shown to exhibit this bursting behaviour, making this an important model for analysing scRNA-seq data. We will consider a slightly more general model with the burst distribution $\{\pi_j\}_{j=1}^{\infty}$ no longer restricted to geometric:

$$\begin{aligned} &\varnothing \xrightarrow{k_1} bX, \quad \mathbb{P}(b=j) = \pi_j \\ &X \xrightarrow{k_2} \varnothing \end{aligned}$$

this can equivalently be written as the infinite system of reactions:

$$\begin{aligned} &\varnothing \xrightarrow{k_1\pi_1} X \quad \varnothing \xrightarrow{k_1\pi_2} 2X \quad \cdots \quad \varnothing \xrightarrow{k_1\pi_j} jX \quad \cdots \\ &X \xrightarrow{k_2} \varnothing \end{aligned}$$

The transition rates on the state space $S = \mathbb{N}$ are given by

$$q(x,y) = \begin{cases} k_2 x, & y = x-1 \\ k_1\pi_j, & y = x+j \\ -k_2x - k_1, & y = x \\ 0, & \text{otherwise} \end{cases} \tag{4.21}$$

and so the rate matrix can be decomposed into the sum

$$\begin{aligned} Q &= k_2 \begin{bmatrix} 0 & 1 & & \\ & -1 & 2 & \\ & & -2 & \ddots \\ & & & \ddots \end{bmatrix} + k_1 \begin{bmatrix} -1 & & & & \\ \pi_1 & -1 & & & \\ \pi_2 & \pi_1 & -1 & & \\ \pi_3 & \pi_2 & \pi_1 & -1 & \\ & \vdots & & & \ddots \end{bmatrix} \\ &:= k_2 Q_2 + k_1 Q_1 + \sum_{j=1}^{\infty} k_1 \pi_j Q_{1j} \end{aligned}$$

Where $Q_1 = -I$, Q_{1j} is a matrix of 1's on the jth lower diagonal and Q_2 is the first matrix shown. There are 2 key differences compared to the models seen so far: (1) Q involves non-linear terms, products of parameters $k_1\pi_j$. This can be solved by fixing $k_1 = 1$, which is valid as the full set of parameters is not identifiable. (2) The parameter π_j is only present in the jth or 'lower' row of Q, so the choice of constraints / truncation will control the number of terms of the burst distribution that can be bounded.

The constraint set for non-linear optimization is then:

$$\mathcal{C} = \left\{ \begin{array}{l|l} & k_2Q_2p + Q_1p + \sum_{j=1}^{J} \pi_j Q_j p = 0 \\ p \geq 0 & \\ k_2 \geq 0 & \sum_{x=0}^{J} p(x) \leq 1 \\ \pi_j \geq 0 \quad j \in \{1,\dots,J\} & \\ & \sum_{j=1}^{J} \pi_j \leq 1 \\ & \hat{p}_L \leq p \leq \hat{p}_U \end{array} \right\} \tag{4.22}$$

Defining the cross terms $z_2 = k_2 p$, $y_j = \pi_j p$ for $j \in \{i, \dots, J\}$ and $z_1 = k_1 p = p$ (for consistency of notation) the linear constraint set is:

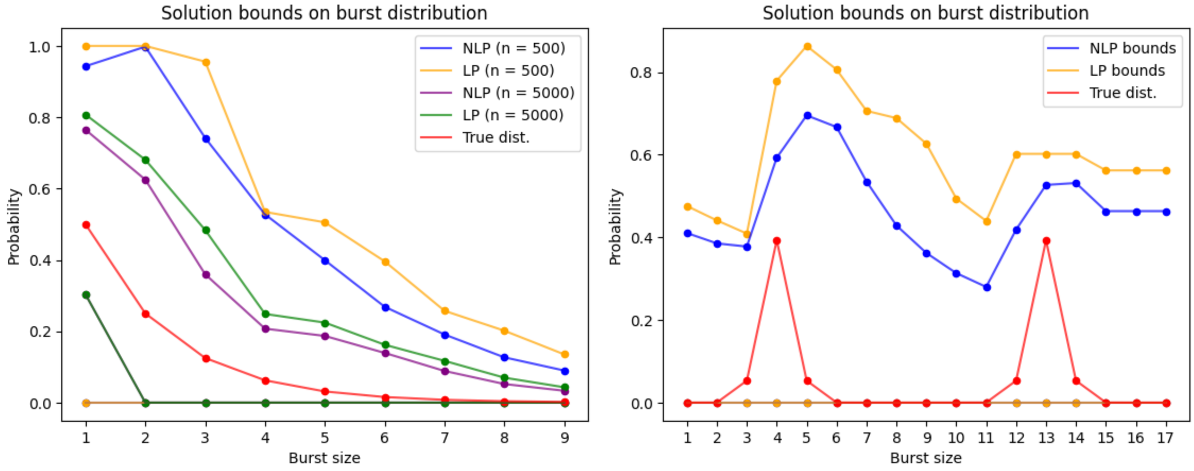
$$\mathcal{C}_{\text{linear}} = \left\{ \begin{array}{r l} z_1, z_2 \geq 0 & \\ k_2 \geq 0 & \\ \pi_j, y_j \geq 0 & j \in \{1, \dots J\} \end{array} \left| \begin{array}{l} Q_2 z_2 + Q_1 z_1 + \sum_{j=1}^{J} Q_j y_j = 0 \\ \sum_{x=0}^{J} z_1(x) \leq 1 \\ \sum_{x=0}^{J} z_2(x) \leq k_2 \\ \sum_{x=0}^{J} y_j(x) \leq \pi_j \quad j \in \{1, \dots, J\} \\ \sum_{j=1}^{J} y_j(x) \leq z_1(x) \quad x \in \{0, \dots, J\} \\ \sum_{j=1}^{J} \pi_j \leq 1 \\ \hat{p}_L \leq z_1 \leq \hat{p}_U \\ k_2 \hat{p}_L \leq z_2 \leq k_2 \hat{p}_U \\ \pi_j \hat{p}_L \leq y_j \leq \pi_j \hat{p}_U \quad j \in \{1, \dots, J\} \end{array} \right. \right\} \tag{4.23}$$

Where $[\hat{p}_L, \hat{p}_U]$ are the confidence interval bounds on the stationary distribution p produced by the bootstrap, and we use the first $N = J + 1$ equations of $Qp = 0$, equivalent to the state space truncation $E = \{0, \dots, J\}$. Minimizing and maximizing over $\mathcal{C}$ or $\mathcal{C}_{\text{linear}}$ we can bound the parameters k_2 and $\{\pi_j\}_{j=1}^{J}$ with k_1 fixed to 1.

Figure 4.3 shows the results of optimization for 2 different birth distributions. For all results the input data was a sample of n (approximately) stationary observations, simulated from a bursting reaction network of the true parameters.

Figure 4.3a shows the effect of increasing sample size: while all 4 bounds follow the shape of the geometric distribution,the results for $n = 5000$ (green and purple) are much tighter, even producing a non-zero lower bound on π_1, which helps identify the geometric shape. In general a larger sample produces tighter bootstrap CI's on p, tighter constraints which lead to tighter solution bounds, and may also allow the estimation of more parameters π_j since more terms of p can be bounded. Figure 4.3b shows the bounds produced on a bimodal burst distribution for a large sample ($n = 5000$): both methods identify the 2 peaks, but the NLP gives a clearer picture of the shape. In general more complicated burst distributions are harder to estimate, as seen by the difference between the results for $n = 5000$ in plots (a) and (b), but both methods still perform well for non-geometric distributions.

Computing the burst distribution bounds we solve a minimization and maximization problem for each parameter π_j. While for a single problem the LP and NLP have similar solution times, for large J the increased size and number of problems to solve can lead to significant differences in computation time e.g. in 4.3b the LP took 5 seconds but the NLP took over 80. Applied to data with potentially hundreds of states (and so J) the computational efficiency of the LP is a key advantage, despite the wider bounds it produces.



(a) Geometric(0.5) burst distribution

(b) Bimodal burst distribution

Figure 4.3 Solution bounds on the burst distribution π for geometric and bimodal bursts. (a) Upper and lower bounds on a Geometric(0.5) burst distribution for 2 samples of different sizes, computed by optimizing over $\mathcal{C}$ and $\mathcal{C}_{\text{linear}}$ with $J = 9$. Blue and orange (NLP and LP) bounds computed for a sample of n = 500 observations simulated from the model, purple and green (NLP and LP) bounds computed for a sample of n = 5000. (b) Upper and lower bounds on a bimodal burst distribution, computed using $J = 17$ and a sample of n = 5000 observations simulated from a bimodal bursting model.

4.2 Single cell data

We now turn to single cell data, large datasets of individual observations of real cells, and discuss how to combat the significant noise present. We focus on the transcription of DNA, using transcript counts from single cell RNA sequencing to investigate the kinetics of the process. 'Transcriptional bursting', where discrete bursts of transcription are separated by periods of inactivity, has been found in genes for a wide range of settings such as mouse embryonic stem cells (mESC) and other mammalian cells [21]. A study on mammalian cells [23] found that "[transcriptional] bursting kinetics are highly gene-specific" with variation in the size and frequency of bursts. These results suggest the bursting model (above) is an accurate model of transcription and our optimization methods, which can estimate a wide variety of bursting kinetics, are suitable for analysis.

Datasets produced by single-cell RNA sequencing are $m \times n$ matrices, describing m genes observed in n cells. The i, jth entry x_{ij} is the transcript count recorded for the ith gene in the jth cell [15]. For each gene i we assume the counts $\{x_{ij}\}_{j=1}^{n}$ are approximate samples from the stationary distribution of a bursting model, so applying the bootstrap and optimizing we can estimate the bursting kinetics of the gene.

However, measuring such small amounts of biological material as the RNA content of individual cells is challenging and so scRNA-seq records only a small fraction of the true amount, a 'capture efficiency' often as low as 6% [15]. This fraction varies between cells and batches (groups of cells being sequenced) and causes 'dropouts', zero or missing count values, leading to sparse datasets. As shown by the aptly named [24] "modelling capture efficiency of single cell RNA sequencing data improves inference of transcriptome wide burst kinetics", suggesting it is important to account for capture efficiency when analysing scRNA-seq data. We investigate 3 approaches, introduced below, to analyse single cell data within the framework of our optimization method:

Approach 1: Observed data Ignoring capture efficiency, estimate parameters using the observed transcript counts.

Approach 2: BayNorm Use bayNorm [15] to produce a normalized count dataset which is then used for analysis

Approach 3: B-method Normalize within the optimization.problem using a binomial model of capture efficiency.

4.2.1 Approach 1: Observed Data

For a gene i we assume the observed transcript counts are approximate samples from the stationary distribution p of a bursting model. Applying the bootstrap we compute confidence intervals on $p(x)$ for all counts x in the sample. The only choice to be made is the state space truncation of the optimization. A naive approach is to include all observed counts, but CI's based on counts with few observations are unreliable, causing infeasibility due to conflicting constraints, and outlying observations lead to large, slow optimization problems.

In practice, a good heuristic is to set a threshold τ and take the largest observed count x_τ with more than τ occurrences in the sample as the truncation point: $J = x_\tau$, as illustrated in figure 4.4. For counts $x < x_\tau$ with fewer than τ observations e.g. $x = 7, 8$ in the figure, we set the bounds on $p(x)$ to $[0, 1]$. Note: this is necessary when x is not observed, as the bootstrap does not provide an estimate of $p(x)$ and we cannot assume it is exactly zero. The choice of τ will depend on the sample size, and we have found that $\tau = 5$ performs well for sample sizes around $n = 200$.

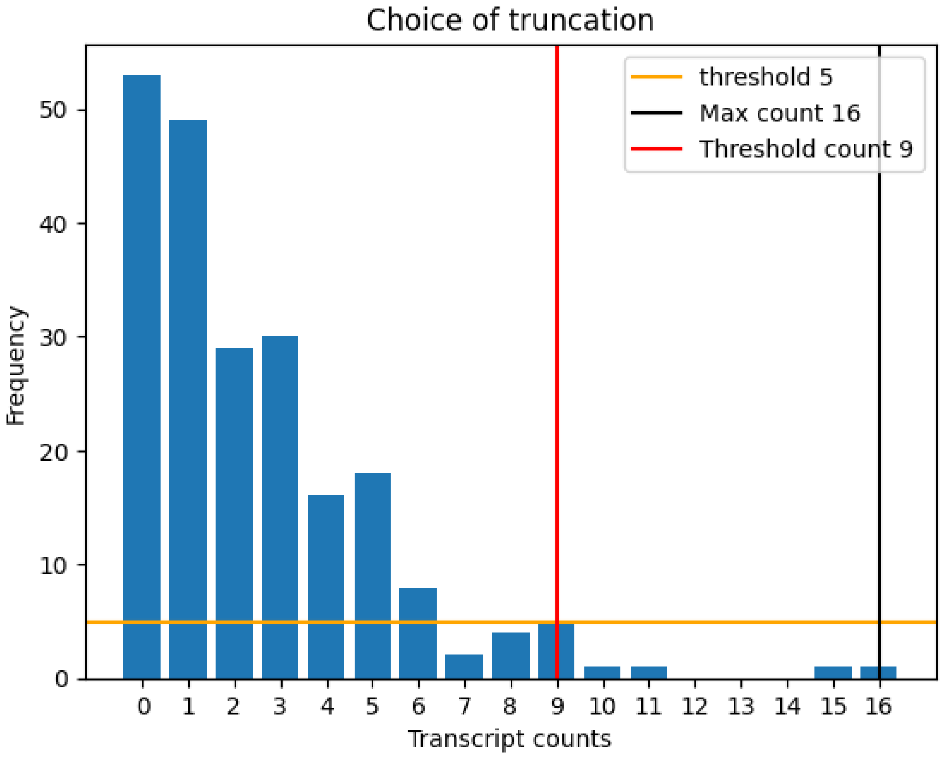


Figure 4.4 Heuristic choice of state space truncation. Frequency plot for a sample of $n = 224$ transcript counts from a single gene. A threshold of $\tau = 5$ on frequency (orange) is used to select the largest count considered in the optimization (red), which is much lower than the largest observed count (black).

4.2.2 Approach 2: BayNorm

Baynorm is a bayesian method for normalization: scaling counts to adjust for capture efficiency, imputation of dropouts, and correcting for batch effects [15]. Given the 'original' count x_{ij}^0, each observed count x_{ij} is assumed to be binomially distributed with $n = x_{ij}^0$ and $p = \beta_j$, the capture efficiency interpreted as the probability that each original transcript in the cell is observed. Each original count x_{ij}^0 is assumed to follow a negative binomial prior distribution with gene specific parameters: mean μ_i and size ϕ_i, estimated from the data:

$$x_{ij}|x_{ij}^0 \sim \text{Binomial}(x_{ij}^0, \beta_j) \tag{4.24}$$

$$x_{ij}^0 \sim \text{NB}(\mu_i, \phi_i) \tag{4.25}$$

Using Bayes rule gives a shifted negative binomial posterior $x_{ij}^0|x_{ij}$. From the posterior we can take the mode as estimate of each original count, called the 'maximum a posteriori probability' (MAP), producing a normalized dataset of the same size as the input. Or, we can sample e.g. $s = 20$ counts from each posterior, producing an $m \times ns$ normalized dataset where for each gene i we imagine observations from ns 'artificial' cells. These datasets, denoted as MAP and PS counts, can then be analysed as in approach 1, described above.

Capture efficiency is assumed to be cell specific, ignoring gene specific effects, defined as β_j for each cell j. The values are estimated by normalizing the total transcript counts in each cell $T_j = \sum_{i=1}^m x_{ij}$ to a mean capture efficiency $\bar{\beta}$, typically set to 6% [15], and taking the normalized values of T_j as βj. The assumption of a negative binomial prior for the counts x_{ij} implicitly assumes the data follows a model of geometric bursts, since this model has a negative binomial stationary distribution. Using the prior parameters μ_i and ϕi we can estimate this geometric burst, with a point estimate of the mean burst size:

$$\bar{b} = \frac{\mu_i}{\phi_i} \tag{4.26}$$

which can be compared to the results of our optimization method.

4.2.3 Approach 3: B-method

As in approach 1, we use the observed counts from a gene i to produce confidence interval bounds on the distribution of observed counts:

$$\hat{p}_L \leq p_{obs} \leq \hat{p}_U \tag{4.27}$$

Estimating the capture efficiencies β_j as described in approach 2 we then form a matrix of binomial probabilities B to relate these bounds to the distribution of original counts p^0:

$$\hat{p}_L - \frac{\mathbb{E}[X]^U}{\bar{\beta}(x_{max}^0 + 1)}\mathbb{1} \leq Bp^0 \leq \hat{p}_U \tag{4.28}$$

where we truncate the state space of observed counts considered to the set $E = \{0, \ldots, x_{max}\}$, original counts considered to the set $E^0 = \{0, \ldots, x_{max}^0\}$, and $\mathbb{E}[X]^U$ is the upper bound of a 95% confidence interval on the mean of the observed counts. See the details and derivation in appendix A.1.

As usual we assume that the original transcript counts are approximate samples from the stationary distribution p of a bursting reaction model, i.e. that p^0, the distribution of original counts, is an approximation of p. To estimate the parameters of the model we simply take the constraint set $\mathcal{C}$ for the bursting model and replace the constraint:

$$\hat{p}_L \leq p \leq \hat{p}_U \tag{4.29}$$

with the constraint:

$$\hat{p}_L - \frac{\mathbb{E}[X]^U}{\bar{\beta}(x^0_{max}+1)}1 \leq Bp \leq \hat{p}_U \tag{4.30}$$

With analogous replacement for the constraint set $\mathcal{C}_{\text{linear}}$. Optimizing over these new constraints we can estimate the parameters of the model.

The choice of truncation points x_{max} and x^0_{max} provides great flexibility, allowing the number of CI bounds used (via x_{max}) and number of burst probabilities π_j estimated (via x^0_{max}) to be varied independently. In practice we found that thresholding to choose x_{max}, as described in approach 1, and scaling by the average capture efficiency $x^0_{max} = \frac{x_{max}}{\bar{\beta}}$ are sensible choices that perform well.

4.2.4 Analysis

We analyse scRNA-seq data from Larsson et al. [25]: transcript counts for each allele (CAST, c57) for 10,728 genes in 224 primary mouse fibroblast cells (concat), and for 10,928 genes in 188 mouse embryonic stem cells (mesc). The data contains missing values, due to dropout and genes that were not expressed in certain cells. Genes with low average counts and many missing values do not have enough information for inference [24], so we remove those with mean count < 1 (of non-missing values) and those with a significant proportion of missing values ($> 25\%$). This results in 4 datasets, separated by allele and cell type[1]:

Dataset	$\mathbf{m \times n}$
cast concat	3963×224
cast mesc	3439×188
c57 concat	3924×224
c57 mesc	3635×94

Table 4.1 Dataset sizes.

Categorization

Following approach 1, we use the observed counts to bound the burst distributions of each gene in all 4 datasets. We then categorize genes according to the properties of these bounds, displaying the full results in figure 4.6 and results for the LP and NLP across all datasets in the sankey diagrams 4.5. The categories, defined below, were chosen to give an overview of the shape of bounds produced by the method, building a picture of the bursting behaviour of genes in the data:

Infeasible the optimization problem is infeasible due to conflicting constraints

Feasible the optimization problem produced a feasible solution (complement of infeasible)

Zero lb the lower bounds for all π_j are 0

Non-zero lb at least one π_j has a non-zero lower bound

Flat ub the upper bounds for all π_j are constant

Increasing ub the upper bounds on π_j's only increase as j increases

[1] We note that in 'c57 mesc' transcript counts are only available for 94 cells

Decreasing ub the upper bounds on π_j's only decrease as j increases

Mixture ub the upper bounds on π_j's both increase and decrease as j increases

Baynorm consistent the geometric burst distribution estimated from the BayNorm prior parameters lies within the bounds for each π_j

Inconsistent at least one π_j has bounds which do not contain the geometric estimate

The sankey plots in figure 4.5 give some insight into the bursting behaviour of genes in the datasets, and the performance of our optimization methods. The NLP is infeasible slightly more often than the LP, expected due to tighter constraints, but overall infeasibility is low suggesting a bursting model of transcription is consistent with many of the genes. Large proportions of genes with all zero lower bound estimates suggest this is a 'harder' problem than bounding above, supported by similar results on simulated data in 4.3. The LP produces a much greater proportion of flat upper bounds than the NLP, these are mostly results with un-informative bounds of $\pi_j \in [0,1]$ for all j, showing the linear relaxation of constraints often does not retain enough information to infer kinetics.

Most genes with non-zero lower bounds have a mixed or decreasing upper bound, potential indications of a geometric burst distribution, such as in 4.3a, where smaller bursts are more common than larger ones. While the majority of solutions are consistent with the geometric burst distribution produced by BayNorm this is slightly misleading as BayNorm accounts for capture efficiency, leading to 'flat' distributions spread over hundreds of burst sizes, whereas observed counts lead to distributions over much smaller bust sizes. As such, the inconsistent genes are mostly those with any non-zero lower bound. Finally, both methods produce sizeable proportions of solutions with increasing upper bounds, potential indications of non-geometric behaviour where larger bursts are more common than smaller ones.

The table in figure 4.6 presents the breakdown of categories by dataset and optimization type. We see infeasibility is significantly more common in the 'cast mesc' dataset, with more than 20% of NLP optimization problems infeasible, which could indicate that some cast alleles in mouse embryonic stem cells do not follow a bursty model of transcription. Again we see the LP produces many 'flat' upper bounds, but also how the smaller sample size of the 'c57 mesc' dataset affects this, more than doubling the percentage. Overall, the smaller sample size does not appear to affect the NLP more than standard differences between datasets, but appears to significantly affect the LP results, something not seen on simulated data where the LP and NLP were affected similarly 4.3a.

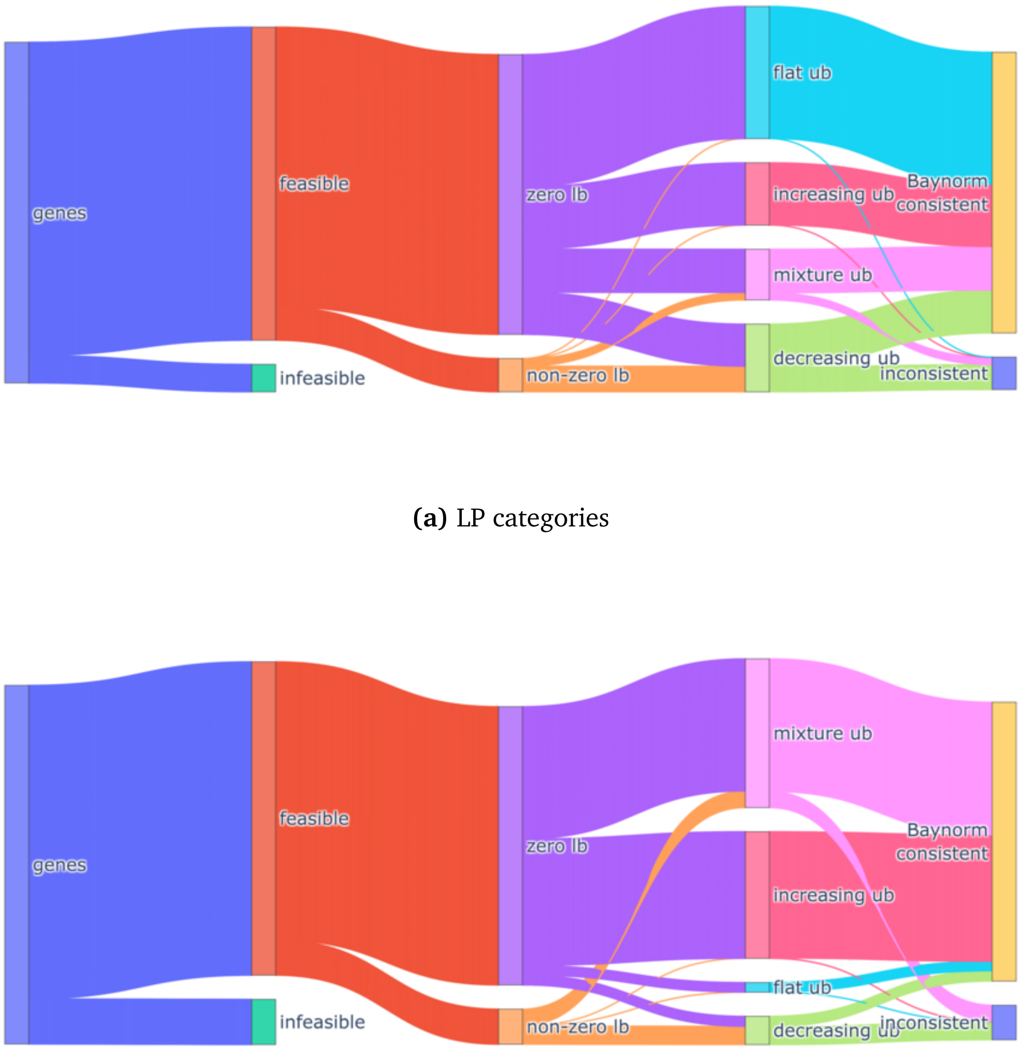
BayNorm and the B-method

Considering a gene categorized as infeasible by analysis of observed counts we follow approaches 2 and 3, using BayNorm MAP and PS normalized counts and the B-method to obtain bounds on its burst distribution, shown in figure 4.7. The bounds produced appear to follow a geometric-like shape, and in fact are all consistent with the geometric burst distribution estimated from the BayNorm prior with mean burst size $\bar{b} \approx 20$, despite no such assumption from the B-method. So accounting for capture efficiency, we are able to estimate the bursting kinetics of genes that we could not analyse using observed data.

However, accounting for capture efficiency significantly increases the size of the optimization problems and so the time taken to solve them, especially for the NLP. The time taken to produce the bounds in 4.7c for the NLP was 30 minutes compared to only 20 seconds for the LP, with similar behaviour for 4.7b. The NLP is thus impractical for large scale analysis and we focus attention on the LP. Taking one gene as an example from each combination of upper and lower bound categories[2] we estimate the burst distribution using all approaches discussed, solving the LP for observed counts, BayNorm MAP counts, BayNorm PS counts and the B-method. The results are displayed in figures 4.8 to 4.14.

In many cases the bounds produced do not provide any information and are simply $[0, 1]$ for all π_j. This is a major drawback of the B-method, with the only informative bounds being those in figure 4.7 for the infeasible example. The burst distributions estimated using BayNorm and the B-method typically range over much larger burst sizes than those estimated from observed data, making direct comparisons difficult. However, in several examples, such as in figure 4.14, the shape of the burst distribution estimated on the observed counts is similar to the shape estimated on the normalized counts. This suggests that the conclusions drawn from the shape based categorization of burst distributions estimated from observed counts could provide insight into the true bursting kinetics of genes.

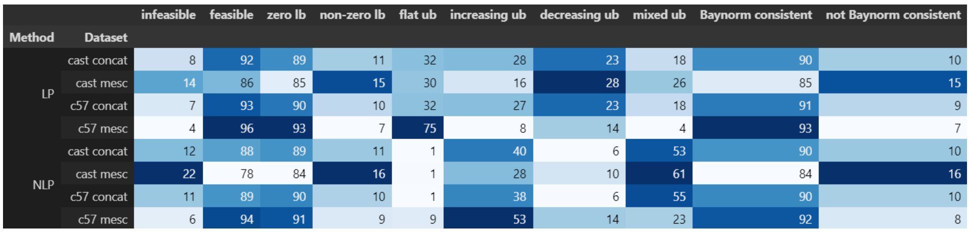
[2] There was no suitable data for flat upper bounds and non-zero lower bounds.



(a) LP categories

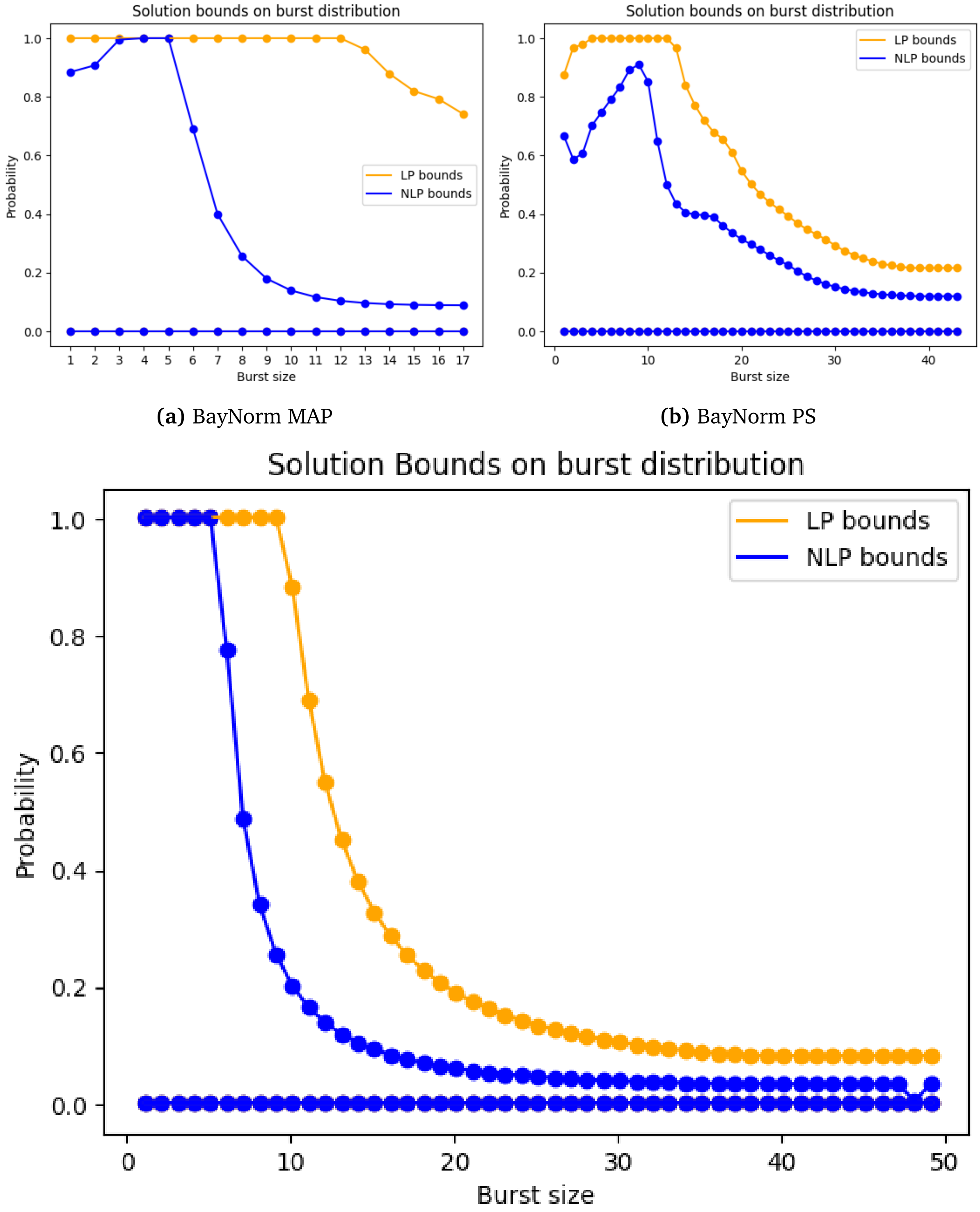
(b) NLP categories

Figure 4.5 Sankey diagrams categorizing genes by estimated bursting behaviour. An estimate of the burst distribution for each gene across all 4 datasets was produced from observed data, and the resulting LP and NLP bounds were categorized according to 4.2.4. Diagrams (a) and (b) show the category membership of all genes for their LP and NLP bounds respectively.



		infeasible	feasible	zero lb	non-zero lb	flat ub	increasing ub	decreasing ub	mixed ub	Baynorm consistent	not Baynorm consistent
Method	**Dataset**										
LP	cast concat	8	92	89	11	32	28	23	18	90	10
	cast mesc	14	86	85	15	30	16	28	26	85	15
	c57 concat	7	93	90	10	32	27	23	18	91	9
	c57 mesc	4	96	93	7	75	8	14	4	93	7
NLP	cast concat	12	88	89	11	1	40	6	53	90	10
	cast mesc	22	78	84	16	1	28	10	61	84	16
	c57 concat	11	89	90	10	1	38	6	55	90	10
	c57 mesc	6	94	91	9	9	53	14	23	92	8

Figure 4.6 Category percentages of results separated by dataset and optimization type. Percentages, per row, of genes belonging to each category of 4.2.4. Darker blues in each column indicate a higher percentage compared to other rows.

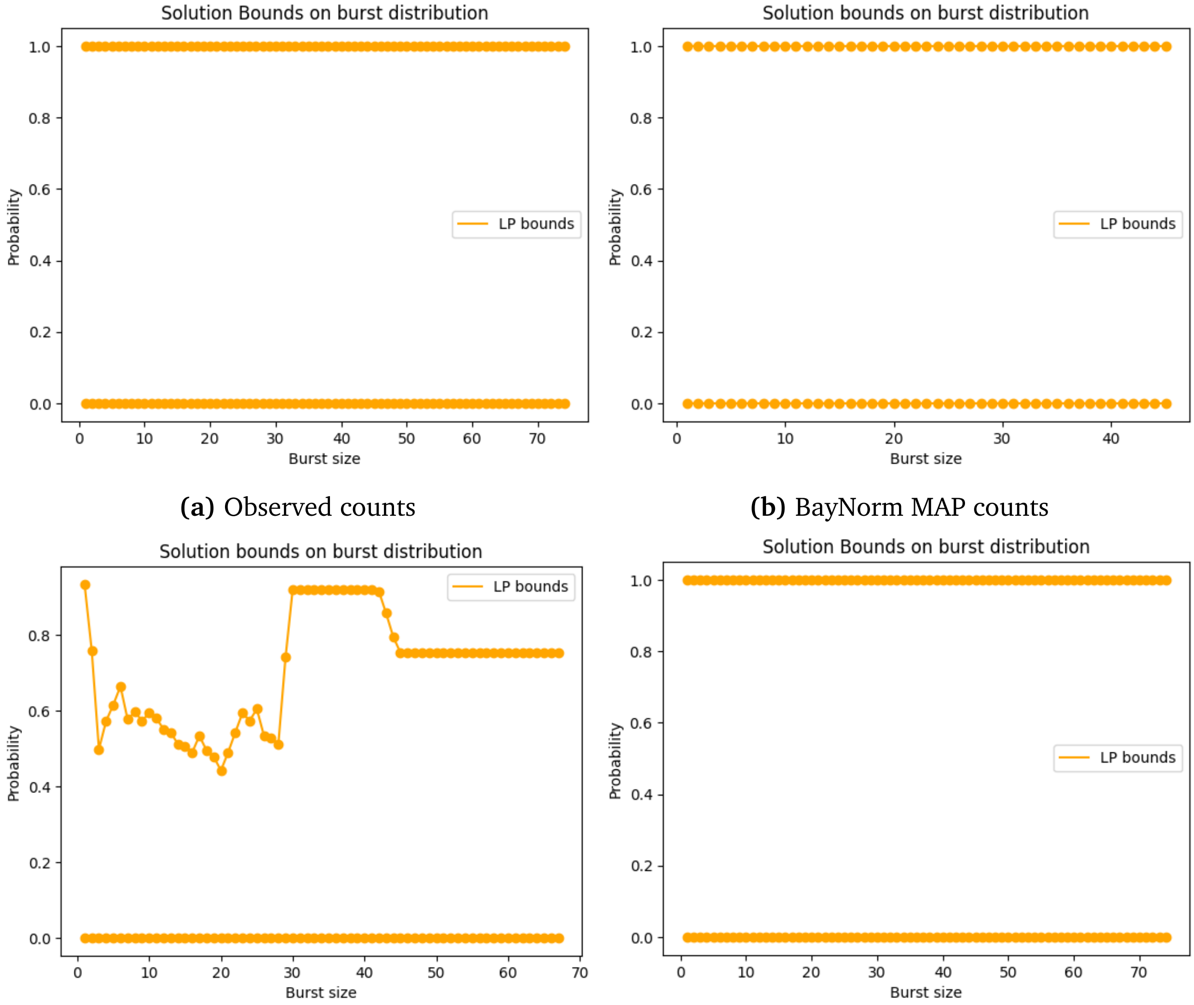


(a) BayNorm MAP

(b) BayNorm PS

(c) B-method

Figure 4.7 Baynorm and B-method burst distribution bounds for an infeasible example. Plots of the bounds computed for the burst distribution of a gene categorized as infeasible using observed counts. Plot (a) gives bounds using BayNorm MAP counts. Plot (b) gives bounds using BayNorm PS counts. Plot (c) gives bounds using the B-method.



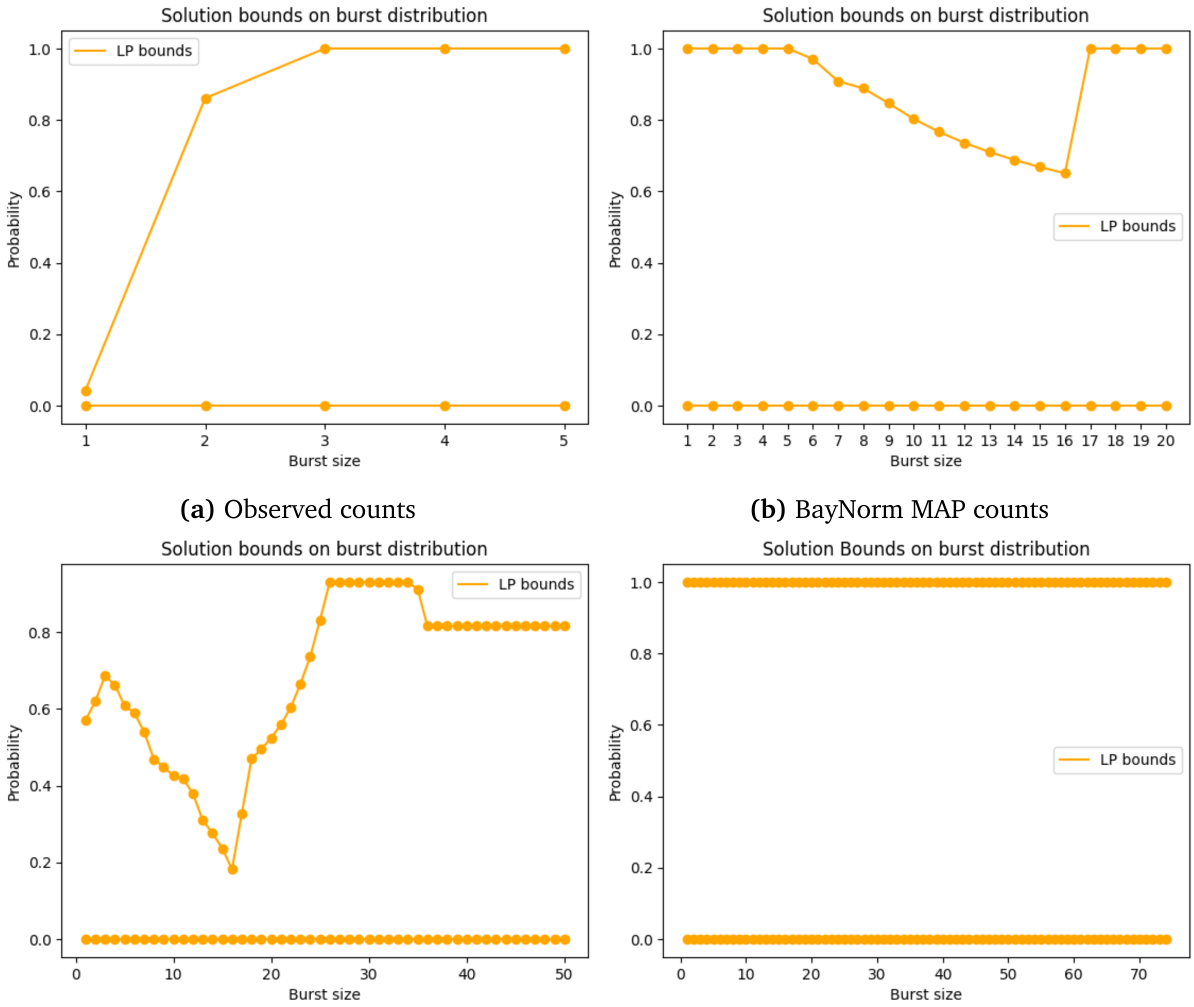
(a) Observed counts

(b) BayNorm MAP counts

(c) BayNorm PS counts

(d) B-method

Figure 4.8 Flat upper bound, zero lower bound example



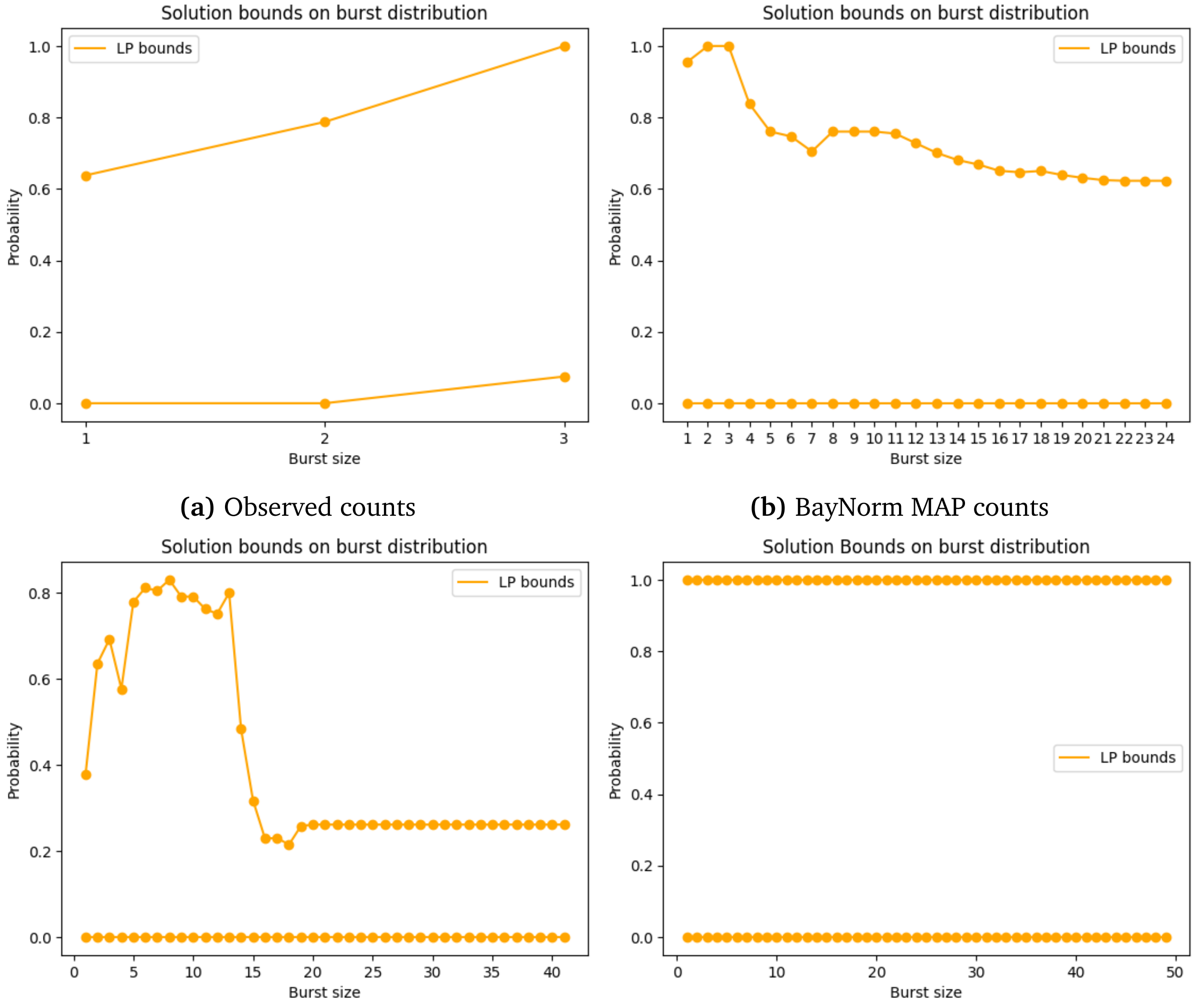
(a) Observed counts

(b) BayNorm MAP counts

(c) BayNorm PS counts

(d) B-method

Figure 4.9 Increasing upper bound, zero lower bound example



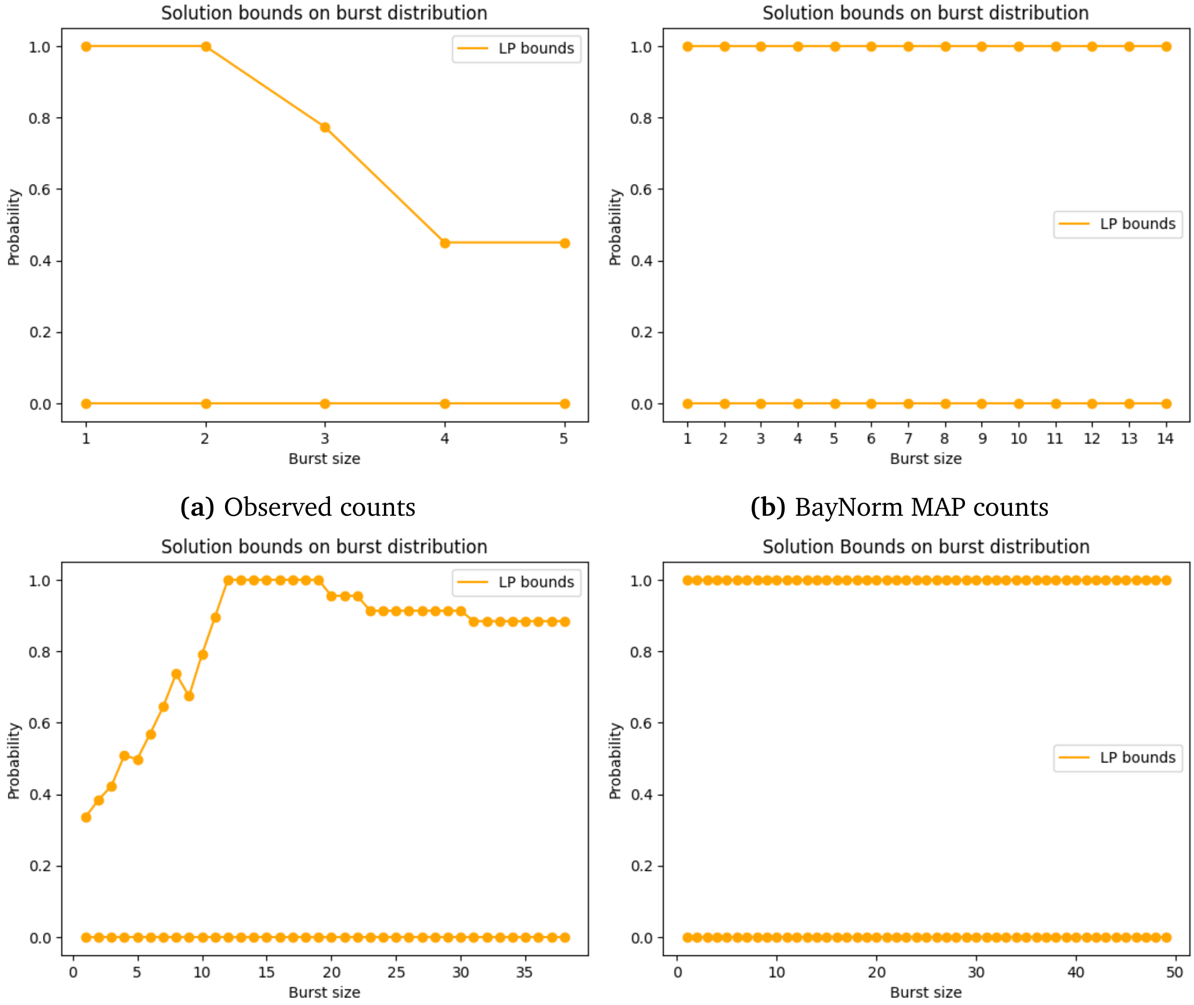
(a) Observed counts

(b) BayNorm MAP counts

(c) BayNorm PS counts

(d) B-method

Figure 4.10 **Increasing upper bound, non-zero lower bound example**



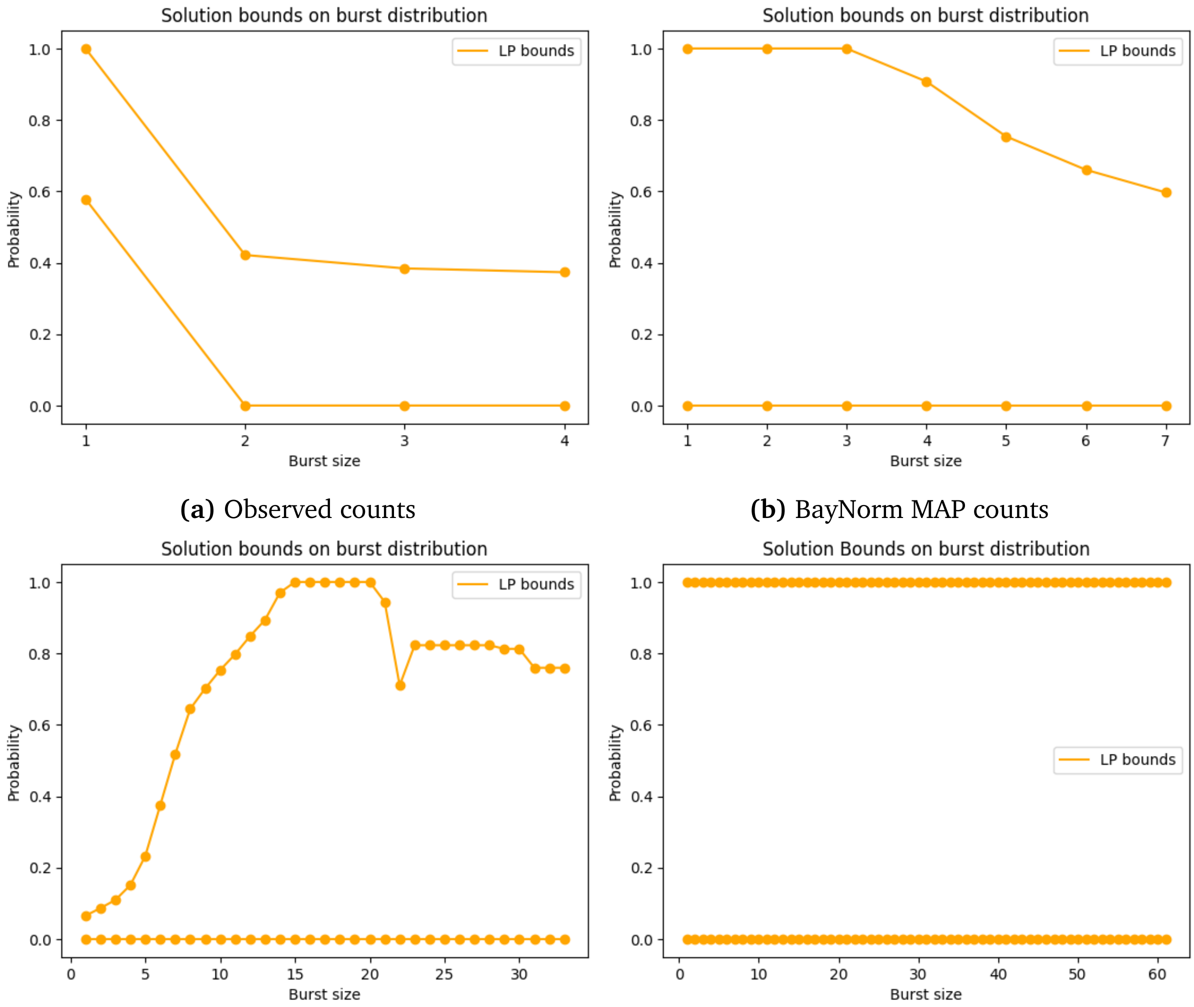
(a) Observed counts

(b) BayNorm MAP counts

(c) BayNorm PS counts

(d) B-method

Figure 4.11 Decreasing upper bound, zero lower bound example



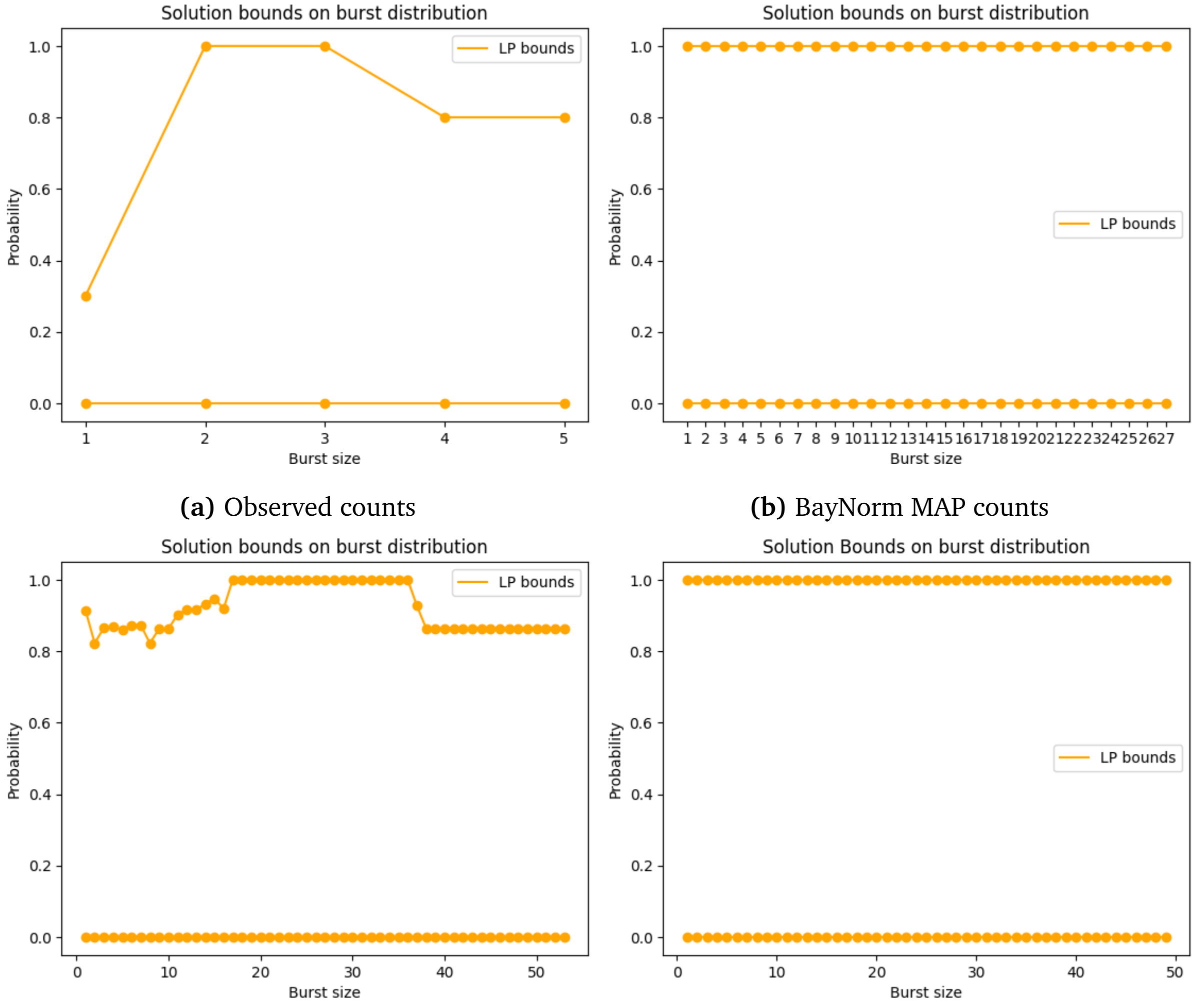
(a) Observed counts

(b) BayNorm MAP counts

(c) BayNorm PS counts

(d) B-method

Figure 4.12 Decreasing upper bound, non-zero lower bound example



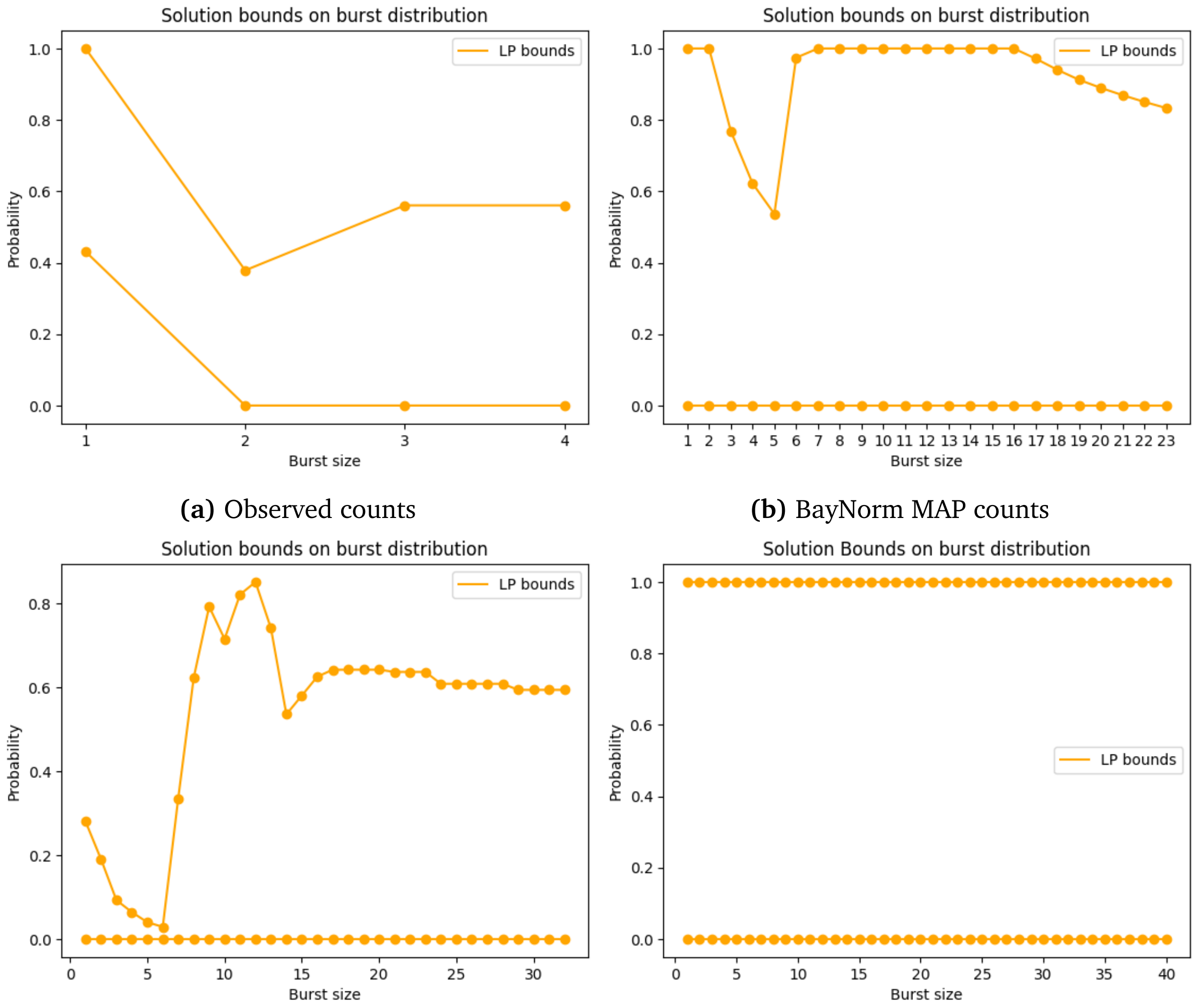
(a) Observed counts

(b) BayNorm MAP counts

(c) BayNorm PS counts

(d) B-method

Figure 4.13 Mixed upper bound, zero lower bound example



(a) Observed counts

(b) BayNorm MAP counts

(c) BayNorm PS counts

(d) B-method

Figure 4.14 Mixed upper bound, non-zero lower bound example

Chapter 5

Conclusion

We introduced a method for parameter inference in stochastic reaction networks based on linear and non-linear optimization over the stationary distribution, demonstrating its performance on data simulated from several reaction networks. We then focused on inference of transcriptional bursting kinetics, discussing 3 approaches to combat the capture efficiency problems of single cell sequencing data when applying the method.

For the naive approach, we used linear and non-linear optimization to categorize the bursting kinetics of genes using observed data. Through a small number of examples we saw the performance of the more advanced ’BayNorm’ and ’B-method’ approaches, and the potential connection to results obtained from observed data.

However, the computational demand of these approaches, especially for non-linear optimization, prevented large scale analysis within the limits of time and compute available. As an extension of the project, it would be of interest to perform a similar categorization of bursting kinetics using the ’BayNorm’ and ’B-method’ approaches, investigating their performance and relationship between the different categorizations.

Appendix A

First Appendix

A.1 B-method derivation

Consider a cell with capture efficiency β, observed transcript count X and original count X^0. As in BayNorm[15] we assume each original transcript has probability β of observation so that:

$$X|X^0, \beta \sim \text{Binomial}(X^0, \beta) \tag{A.1}$$

where we define:

$$B^{\beta}_{x,x^0} := \mathbb{P}(X = x|X^0 = x^0, \beta) = \begin{cases} \binom{x^0}{x} \beta^x (1-\beta)^{x^0 - x}, & x \leq x^0 \\ 0, & x > x^0 \end{cases} \tag{A.2}$$

The case of $x > x^0$ is when the observed number of transcripts in a cell is higher than the original number (the truth) which we assume cannot happen. Taking S and S^0 as the state spaces of observed and original counts respectively, the probability of observing $X = x \in S$ is:

$$\mathbb{P}(X = x) = \sum_{x^0 \in S^0} \mathbb{P}(X = x|X^0 = x^0)\mathbb{P}(X^0 = x^0) \tag{A.3}$$

For a distribution of capture efficiencies $\beta \sim p(\beta)$:

$$\mathbb{P}(X = x|X^0 = x^0) = \int_\beta \mathbb{P}(X = x|X^0 = x^0, \beta)p(\beta)d\beta \tag{A.4}$$

which together give the probability of observing $X = x \in S$, in any cell, as:

$$\begin{aligned} \mathbb{P}(X = x) &= \int_\beta \sum_{x^0 \in S^0} \mathbb{P}(X = x|X^0 = x^0, \beta)\mathbb{P}(X^0 = x^0)p(\beta)d\beta && \text{(A.5)} \\ &= \mathbb{E}_\beta \left[\sum_{x^0 \in S^0} \mathbb{P}(X = x|X^0 = x^0, \beta)\mathbb{P}(X^0 = x^0) \right] && \text{(A.6)} \\ &= \mathbb{E}_\beta \left[\sum_{x^0 \in S^0} B^{\beta}_{x,x^0} \mathbb{P}(X^0 = x^0) \right] && \text{(A.7)} \end{aligned}$$

Defining the matrix $B^\beta = \left\{ B^{\beta}_{x,x^0} \right\}_{x \in S, x^0 \in S^0} \in \mathbb{R}^{|S| \times |S^0|}$, of binomial probabilities, and the column vectors $p_{obs} = \{\mathbb{P}(X = x)\}_{x \in S}$ and $p^o = \left\{\mathbb{P}(X^0 = x^0)\right\}_{x^0 \in S^0}$, of the observed and original count distributions respectively, we can write this in vector form as:

$$p_{obs} = \mathbb{E}_\beta \left[B^\beta p^0 \right] = \mathbb{E}_\beta \left[B^\beta \right] p^0 \tag{A.8}$$

Taking p^0 out of the expectation, since the original counts do not depend on the capture efficiency.

However, we do not have access to the true distribution of capture efficiencies $p(\beta)$. Instead, we use the capture efficiencies, estimated as described in approach 2 (BayNorm), of the sample $\{\beta_j\}_{j=1}^n$ as an empirical distribution, replacing the expectation by the sample mean to give:

$$p_{obs} = \left(\frac{1}{n} \sum_{j=1}^{n} B^{\beta_j} \right) p^0 := Bp^0 \tag{A.9}$$

The matrix $B \in \mathbb{R}^{|S| \times |S^0|}$ can then be computed, and relates the distribution of observed counts to the distribution of original counts. Bounds on the observed distribution, such as from the bootstrap, give bounds on the original distribution:

$$\hat{p}_L \leq p_{obs} \leq \hat{p}_U \implies \hat{p}_L \leq Bp^0 \leq \hat{p}_U \tag{A.10}$$

[Note that B has a very high condition number, so it is not sensible to invert for direct bounds on p^0 as the inverse B^{-1} is very sensitive to small changes in the observed bounds]

Yet again we face the problem of an infinite state space: both $|S|$ and $|S^0|$ are typically infinite, and we cannot bound the infinite vector p_{obs} using a finite sample, or optimize over the infinite vector p^0. We truncate both state space to the sets:

$$E = \{0, \dots, x_{max}\} \subset S \tag{A.11}$$
$$E^0 = \{0, \dots, x^0_{max}\} \subset S^0 \tag{A.12}$$
$$\tag{A.13}$$

Consider the xth row of $p_{obs} = Bp^0$, since B is upper triangular this is a sum over all entries of p^0, those in the truncation E^0 and those outside:

$$\mathbb{P}(X = x) = \sum_{x^0=0}^{x^0_{max}} B_{x,x^0} \mathbb{P}(X^0 = x^0) + \sum_{x^0 \geq x^0_{max}+1} B_{x,x^0} \mathbb{P}(X^0 = x^0) \tag{A.14}$$

The truncation error t_e is given by the second term which sums over the counts outside the truncation E^0. Using the fact that $B_{x,x^0} \leq 1$ because it is an average of the binomial probabilities $B^{\beta_j}_{x,x^0}$ we have that:

$$t_e \leq \sum_{x \geq x^0_{max}+1} \mathbb{P}(X^0 = x^0) \tag{A.15}$$
$$= \mathbb{P}(X^0 \geq x^0_{max} + 1) \tag{A.16}$$

Since X^0 is a non-negative random variable we can apply Markov's inequality, and note that t_e is a sum of non-negative terms to give:

$$0 \leq t_e \leq \frac{\mathbb{E}[X^0]}{x^0_{max} + 1} \tag{A.17}$$

This inequality on the truncation error holds for every row of $p_{obs} = Bp^0$ so by rearranging we obtain:

$$p_{obs} - \frac{\mathbb{E}[X^0]}{x^0_{max} + 1} \underline{1} \leq Bp^0 \leq p_{obs} \tag{A.18}$$

for the truncated matrix $B = \{B_{x,x^0}\}_{x \in E, x^0 \in E^0}$ and distributions $p_{obs} = \{\mathbb{P}(X = x)\}_{x \in E}$ and $p^o = \left\{\mathbb{P}(X^0 = x^0)\right\}_{x^0 \in E^0}$. Using the law of total expectation, the fact that $X|X^0, \beta \sim \text{Binomial}(X^0, \beta)$ and that the original count X^0 and capture efficiency β are independent we have that:

$$\mathbb{E}[X] = \mathbb{E}[\mathbb{E}[X|X^0, \beta]] = \mathbb{E}[X^0\beta] = \mathbb{E}[X^0]\mathbb{E}[\beta] \tag{A.19}$$

and so:

$$\mathbb{E}[X^0] = \frac{\mathbb{E}[X]}{\bar{\beta}} \tag{A.20}$$

where the mean capture efficiency $\bar{\beta}$ is typically 6%. To estimate $\mathbb{E}[X]$ we use the sample of observed counts to compute a 95% confidence interval for the mean, taking the upper bound (as we require an upper bound on the truncation error), denoted by $\mathbb{E}[X]^U$.

Bibliography

[1] D. Schnoerr, G. Sanguinetti, and R. Grima. Approximation and inference methods for stochastic biochemical kinetics—a tutorial review. *Journal of Physics A: Mathematical and Theoretical*, 50(9):093001, 2017.

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

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

[4] J. M. Raser and E. K. O'shea. Noise in gene expression: origins, consequences, and control. *Science*, 309(5743):2010–2013, 2005.

[5] G. Gorin, J. J. Vastola, and L. Pachter. Studying stochastic systems biology of the cell with single-cell genomics data. *Cell Systems*, 14(10):822–843, 2023.

[6] J. Kuntz, P. Thomas, G.-B. Stan, and M. Barahona. Bounding the stationary distributions of the chemical master equation via mathematical programming. *The Journal of chemical physics*, 151(3), 2019.

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

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

[9] K. Öcal, M. U. Gutmann, G. Sanguinetti, and R. Grima. Inference and uncertainty quantification of stochastic gene expression via synthetic models. *Journal of The Royal Society Interface*, 19(192):20220153, 2022.

[10] M. A. Beaumont, W. Zhang, and D. J. Balding. Approximate bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, 2002.

[11] F. Fröhlich, P. Thomas, A. Kazeroonian, F. J. Theis, R. Grima, and J. Hasenauer. Inference for stochastic chemical kinetics using moment equations and system size expansion. *PLoS computational biology*, 12(7):e1005030, 2016.

[12] R. Grima. A study of the accuracy of moment-closure approximations for stochastic chemical kinetics. *The Journal of chemical physics*, 136(15), 2012.

[13] J. Kuntz, P. Thomas, G.-B. Stan, and M. Barahona. Stationary distributions of continuous-time markov chains: a review of theory and truncation-based approximations. *SIAM Review*, 63(1):3–64, 2021.

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

[15] W. Tang, F. Bertaux, P. Thomas, C. Stefanelli, M. Saint, S. Marguerat, and V. Shahrezaei. baynorm: Bayesian gene expression recovery, imputation and normalization for single-cell rna-sequencing data. *Bioinformatics*, 36(4):1174–1181, 2020.

[16] L. Wasserman. *All of nonparametric statistics*. Springer Science & Business Media, 2006.

[17] D. G. Luenberger, Y. Ye, et al. *Linear and nonlinear programming*, volume 2. Springer, 1984.

[18] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023.

[19] S. Diamond and S. Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[20] P. Thomas. Stochastic modeling approaches for single-cell analyses. In O. Wolkenhauer, editor, *Systems Medicine*, pages 45–55. Academic Press, Oxford, 2021.

[21] X. Luo, F. Qin, F. Xiao, and G. Cai. Bisc: accurate inference of transcriptional bursting kinetics from single-cell transcriptomic data. *Briefings in Bioinformatics*, 23(6):bbac464, 2022.

[22] L. Rüschendorf. Sharpness of fréchet-bounds. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 57(2):293–302, 1981.

[23] D. M. Suter, N. Molina, D. Gatfield, K. Schneider, U. Schibler, and F. Naef. Mammalian genes are transcribed with widely different bursting kinetics. *science*, 332(6028):472–474, 2011.

[24] W. Tang, A. C. S. Jørgensen, S. Marguerat, P. Thomas, and V. Shahrezaei. Modelling capture efficiency of single-cell rna-sequencing data improves inference of transcriptome-wide burst kinetics. *Bioinformatics*, 39(7):btad395, 2023.

[25] A. J. Larsson, P. Johnsson, M. Hagemann-Jensen, L. Hartmanis, O. R. Faridani, B. Reinius, Å. Segerstolpe, C. M. Rivera, B. Ren, and R. Sandberg. Genomic encoding of transcriptional burst kinetics. *Nature*, 565(7738):251–254, 2019.