Modelling and Experimental Results on Stochastic Model Reduction, Protein Maturation, Macromolecular Crowding, and Time-varying Gene Expression.

by

Guangqiang Dong

A thesis submitted in conformity with the requirements for the degree of Doctor of Philosophy
Graduate Department of Physics
University of Toronto

Copyright © 2009 by Guangqiang Dong
Gene expression, which connects genomic information to functional units in living cells, has received substantial attention since the completion of The Human Genome Project. Quantitative characterization of gene expression will provide valuable information for understanding the behavior of living cells, and possibilities of building synthetic gene circuits to control or modify the behavior of naturally occurring cells. Many aspects of quantitative gene expression have been studied, including gene expression dynamics and noise in *E. coli*. The gene expression process itself is stochastic, and modelling approaches have been broadly used to study gene expression noise; however, stochastic gene expression models are usually large and time intensive to simulate. To speed up simulations, we have developed a systematic method to simplify gene expression models with fast and slow dynamics, and investigated when we can ignore the gene expression from the background genome when modelling the gene expression from plasmids. When modelling the noise in gene expression, one usually neglected aspect is the slow maturation process of fluorescent proteins, necessary for the protein to give out fluorescence after it is produced. By modelling, we show that the maturation steps can bring large changes to both the mean protein number and the noise in the model. An unstudied aspect of gene expression dynamics is the time dependent gene expression behavior in *E. coli* batch culture. Contrary to the usual assumption, we have found, in *E. coli* batch culture gene expression, that there is no steady state in terms of both the mean number of proteins and the noise. Negative feedback is thought to be able to reduce the noise in a system, and experiments have shown that negative feedback indeed suppresses the noise in gene expression, but the modelling shows that negative feedback will increase the noise. We have found that the increase of noise by feedback is due...
to the exclusion of extrinsic noise from the model, and that negative feedback will suppress the extrinsic noise while increasing the intrinsic noise. Living cells are crowded with macromolecules, which will, predicted by modelling, make the reaction constant time dependent. Our experimental observation has confirmed this prediction.
Acknowledgements

I would like to express my deep and sincere gratitude to my supervisor, Professor David McMillen, for his detailed and constructive comments, and for his support and guidance throughout this work. I would also like to thank Sangram Bagh and Sumon (Mostafizur) Mazumder for helping me learn biology. Many other people have provided helpful advice during this thesis: I would especially like to thank Professor R.J. Dwayne Miller and Professor Virginijus Barzda.
Contents

1 Introduction

1.1 Background and motivation .............................................. 1
  1.1.1 Systems biology and synthetic biology ............................ 1
  1.1.2 Gene expression ................................................ 2
  1.1.3 Gene expression dynamics in living cells ...................... 3
  1.1.4 Noise in gene expression ........................................ 5

1.2 Experimental methods on gene expression .......................... 6
  1.2.1 Green Fluorescent Protein ...................................... 6
  1.2.2 Fluorescence microscopy ....................................... 8
  1.2.3 Flow cytometry ................................................ 8

1.3 Modelling of gene expression ......................................... 11
  1.3.1 Deterministic modelling ......................................... 11
  1.3.2 Stochastic modelling ............................................ 12
  1.3.3 Fitting and optimization ........................................ 16
  1.3.4 Distributed computation ....................................... 18
  1.3.5 Stochastic simulation of a population of cells ............... 18

1.4 The focus of my research ............................................. 19
  1.4.1 Simplification of stochastic chemical reaction models with fast and slow dynamics .................................................. 19
  1.4.2 Increasing the efficiency of bacterial transcription simulations: when to exclude the genome without loss of accuracy 20
  1.4.3 Effect of protein maturation on the noise in gene expression 20
  1.4.4 Time series gene expression dynamics in \textit{E. coli} batch culture 21
  1.4.5 Effect of feedback on the noise in gene expression ........... 21
  1.4.6 Macromolecular crowding effect on the dynamics in gene expression 22
2 Simplification of stochastic chemical reaction models 23
  2.1 Introduction ................................................................. 23
  2.2 Methods ................................................................. 24
  2.3 Applications ............................................................... 33
    2.3.1 Signal cascade model ............................................... 33
    2.3.2 Detailed gene expression model ................................... 42
  2.4 Discussion ............................................................... 53

3 Increasing the efficiency of transcription simulations 55
  3.1 Introduction and background ........................................... 55
  3.2 Method ................................................................. 56
    3.2.1 *E. coli* gene expression model .................................. 56
    3.2.2 Modelling approach and software ................................ 63
    3.2.3 Additional deterministic simulations ............................ 64
    3.2.4 Parameter sweeping ................................................. 65
    3.2.5 Interpolation of on-rate thresholds ............................... 69
  3.3 Results and discussion ................................................ 69
    3.3.1 Percent difference of reporter transcript averages between models ........................................ 69
    3.3.2 Accuracy of the interpolated on-rates ................................ 70
    3.3.3 Time reduction via genome exclusion .............................. 73
    3.3.4 Relationship between the parameters .............................. 74
    3.3.5 Potential extensions ................................................ 74
  3.4 Conclusion ............................................................... 78

4 Effect of protein maturation on gene expression noise 80
  4.1 Introduction ............................................................... 80
  4.2 Gene expression model ................................................ 81
    4.2.1 Analysis of steady-state statistics ............................... 81
    4.2.2 Noise power spectra ................................................ 95
  4.3 Conclusion ............................................................... 101

5 Time series gene expression dynamics 104
  5.1 Introduction ............................................................... 104
  5.2 Method ................................................................. 105
    5.2.1 Plasmid construction ............................................... 105
    5.2.2 Cell culture .......................................................... 105
    5.2.3 Flow cytometry ....................................................... 105
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2.4 Microscopy</td>
<td>105</td>
</tr>
<tr>
<td>5.2.5 Protein number quantification</td>
<td>106</td>
</tr>
<tr>
<td>5.2.6 Data analysis</td>
<td>106</td>
</tr>
<tr>
<td>5.3 Result</td>
<td>106</td>
</tr>
<tr>
<td>5.4 Modelling</td>
<td>112</td>
</tr>
<tr>
<td>5.4.1 Modelling by tracking all cells</td>
<td>112</td>
</tr>
<tr>
<td>5.4.2 Modelling by tracking only one random daughter cell</td>
<td>113</td>
</tr>
<tr>
<td>5.5 Conclusion</td>
<td>118</td>
</tr>
<tr>
<td>6 Effect of feedback on the noise in gene expression</td>
<td>120</td>
</tr>
<tr>
<td>6.1 Introduction</td>
<td>120</td>
</tr>
<tr>
<td>6.2 Modelling</td>
<td>121</td>
</tr>
<tr>
<td>6.2.1 Detailed gene expression model with feedback</td>
<td>121</td>
</tr>
<tr>
<td>6.2.2 Simple gene expression model with feedback</td>
<td>122</td>
</tr>
<tr>
<td>6.3 Experimental verification</td>
<td>125</td>
</tr>
<tr>
<td>6.4 Conclusion</td>
<td>127</td>
</tr>
<tr>
<td>7 Macromolecular crowding</td>
<td>128</td>
</tr>
<tr>
<td>7.1 Introduction</td>
<td>128</td>
</tr>
<tr>
<td>7.2 Method</td>
<td>129</td>
</tr>
<tr>
<td>7.3 Result</td>
<td>131</td>
</tr>
<tr>
<td>7.4 Modelling of gene expression in crowded environments</td>
<td>133</td>
</tr>
<tr>
<td>7.5 Conclusion</td>
<td>137</td>
</tr>
<tr>
<td>8 Conclusion</td>
<td>138</td>
</tr>
<tr>
<td>Bibliography</td>
<td>141</td>
</tr>
</tbody>
</table>
## List of Tables

2.1 Brief Description of Species in Signal Cascade Model .......................... 35
2.2 Brief Description of species in Gene Expression Model ........................ 42

3.1 List of Species Names in the Gene Expression Model ............................ 57
3.2 List of Reactions in the Gene Expression Model with Gene Expression from Genome ............................ 57
3.3 List of Reactions in the Gene Expression Model without Gene Expression from Genome ............................ 58

5.1 Description of plasmids used in this project. ...................................... 107

7.1 List of names of variable ................................................................. 134
## List of Figures

1.1 Gene expression steps. ............................................................. 3
1.2 Cell growth in *E. coli* batch culture. ........................................... 4
1.3 *E. coli* Cells expressing EGFP. .................................................. 7
1.4 Wide-field epi-fluorescence microscopy ......................................... 9
1.5 Diagram of Flow Cytometry ...................................................... 10
1.6 Result of BioNetS simulation. .................................................... 16

2.1 Comparisons of the ODE and stochastic solutions ............................. 31
2.2 Schematic of the ShcA signalling cascade. ...................................... 36
2.3 Full signal cascade model, as entered into the BioNetS user interface ...... 37
2.4 Reduced signal cascade model .................................................. 38
2.5 Comparisons of statistical parameters as a function of time for the ERKPP species 39
2.6 Comparisons of statistical parameters as a function of time for the ERK species 40
2.7 Comparisons of statistical parameters as a function of time for the ERKP species 41
2.8 Detailed gene expression model. ................................................. 43
2.9 Full gene expression model (71 reactions, 47 species) ........................ 45
2.10 Comparisons of protein numbers in the gene expression system ............. 46
2.11 Comparisons of statistical parameters as a function of time for protein numbers in the gene expression system ........................................ 47
2.12 First reduction of the gene expression model (28 reactions, 29 species) ... 48
2.13 Second reduction of the gene expression model (10 reactions, 10 species) ... 49
2.14 Comparisons of statistical parameters as a function of time for mRNA numbers in the gene expression system ........................................... 50
2.15 Comparisons of the steady-state frequency histogram for mRNA and protein in the gene expression system .............................................. 51
2.16 Reduction of the gene expression model with negative feedback (12 reactions, 11 species) ............................................................... 51
2.17 Comparisons of statistical parameters as a function of time for Protein numbers in the gene expression system. ................................................................. 52

3.1 Schematic of the two versions of the model system. ................................. 58
3.2 Comparison of stochastic and deterministic simulation outputs. ............. 66
3.3 Typical time series generated by the model. ......................................... 67
3.4 Parameter sweeping ................................................................................ 68
3.5 The effect of the genome on the reporter transcript output. .................. 71
3.6 The ratio of simulation time between models with the genome to those excluding it 72
3.7 Number of reaction steps simulated. ....................................................... 73
3.8 Dependence of promoter strength on gene length and plasmid number. .... 75
3.9 Dependence of promoter strength on plasmid number and mRNA half-life. .... 76
3.10 Dependence of promoter strength on mRNA half-life and gene length. ...... 77

4.1 Schematic depiction of the simplified gene expression model considered here. 83
4.2 Time series of one realization of the stochastic process .......................... 84
4.3 Mean number of matured proteins versus protein maturation time. ......... 90
4.4 Coefficient of variation (η) of the fully matured protein species versus translation rate (c3) .............................................................. 91
4.5 Coefficient of variation (η) of the fully matured protein species versus mean protein number ............................................................... 92
4.6 Coefficient of variation (η) of the fully matured protein species versus mean protein number ............................................................... 93
4.7 Noise power spectra for different species in the model .......................... 96
4.8 Noise power spectra for different species in the model .......................... 97
4.9 Noise power spectra for species Protein-cyclized, Protein-matured, and Protein-matured minus the intrinsic noise power spectrum of the conversion reaction ... 98
4.10 Passing ratio of the upstream extrinsic noise through the final maturation reaction step (oxidation) .............................................................. 100
4.11 Noise power spectrum for different maturation times while transcription rate and translation rate are kept constant ................................. 101

5.1 Results for 4 different plasmid systems. ................................................. 108
5.2 Cell growth at different growth rates and division rates .......................... 109
5.3 Comparison of CV of regulated and unregulated systems. ..................... 111
5.4 The change of protein number histogram over time for one of the pZ12EGFP run. 112
5.5 The change of cell size histogram over time for one of the pZ12EGFP run. .... 112
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.6</td>
<td>Time series of protein number from model that tracks all cells and model that</td>
</tr>
<tr>
<td></td>
<td>only tracks one sibling cell.</td>
</tr>
<tr>
<td></td>
<td>113</td>
</tr>
<tr>
<td>5.7</td>
<td>Time series of protein number from model that tracks all cells and model that</td>
</tr>
<tr>
<td></td>
<td>only tracks one sibling cell.</td>
</tr>
<tr>
<td></td>
<td>114</td>
</tr>
<tr>
<td>5.8</td>
<td>Fitting of cell growth using Eq. 5.3 to experimental OD curve.</td>
</tr>
<tr>
<td></td>
<td>116</td>
</tr>
<tr>
<td>5.9</td>
<td>Fitting of cell size using Eq. 5.4 to experimental cell size curve.</td>
</tr>
<tr>
<td></td>
<td>117</td>
</tr>
<tr>
<td>5.10</td>
<td>Modeled protein number time series vs experimental protein number time series.</td>
</tr>
<tr>
<td></td>
<td>118</td>
</tr>
<tr>
<td>6.1</td>
<td>Time series of protein number for the model without feedback.</td>
</tr>
<tr>
<td></td>
<td>121</td>
</tr>
<tr>
<td>6.2</td>
<td>Time series of protein number for the model with feedback.</td>
</tr>
<tr>
<td></td>
<td>121</td>
</tr>
<tr>
<td>6.3</td>
<td>Gene expression model with extrinsic noise.</td>
</tr>
<tr>
<td></td>
<td>122</td>
</tr>
<tr>
<td>6.4</td>
<td>Noise vs Feedback strength for different binding affinities.</td>
</tr>
<tr>
<td></td>
<td>123</td>
</tr>
<tr>
<td>6.5</td>
<td>CV of protein vs number of protein for models without feedback, with positive</td>
</tr>
<tr>
<td></td>
<td>feedback and with negative feedback</td>
</tr>
<tr>
<td></td>
<td>124</td>
</tr>
<tr>
<td>6.6</td>
<td>Frequency response to extrinsic noise for models without feedback, with positive</td>
</tr>
<tr>
<td></td>
<td>or with negative feedback.</td>
</tr>
<tr>
<td></td>
<td>125</td>
</tr>
<tr>
<td>6.7</td>
<td>Plasmid expressing EGFP and EYFP</td>
</tr>
<tr>
<td></td>
<td>126</td>
</tr>
<tr>
<td>6.8</td>
<td>Plasmid expressing tetR-EGFP and tetR-EYFP</td>
</tr>
<tr>
<td></td>
<td>126</td>
</tr>
<tr>
<td>7.1</td>
<td>Volume exclusion effect in crowded environments.</td>
</tr>
<tr>
<td></td>
<td>129</td>
</tr>
<tr>
<td>7.2</td>
<td>Time series expression of EGFP in vitro with and without crowding agents.</td>
</tr>
<tr>
<td></td>
<td>132</td>
</tr>
<tr>
<td>7.3</td>
<td>Time series expression of EGFP in vitro with and without crowding agents.</td>
</tr>
<tr>
<td></td>
<td>133</td>
</tr>
<tr>
<td>7.4</td>
<td>Fitting of experimental data to proposed models.</td>
</tr>
<tr>
<td></td>
<td>136</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Background and motivation

1.1.1 Systems biology and synthetic biology

The recent advent and refinement of sophisticated techniques for probing and manipulating the internal workings of living cells has led to the emergence of two new fields: systems biology [1–5] and synthetic biology [6–10].

Systems biology aims to assemble the new data sets into models for the behavior of sets of coupled genes, to understand the behavior of a living organism, from the simplest cell *Escherichia coli* (*E. coli*) to more complex living organisms, such as humans. On the molecular level, the extraordinary advances in biochemistry and molecular biology have led to an unprecedented understanding of molecular interactions in living cells. The complexity of cellular pathways and networks often makes it difficult or impossible to understand or reliably predict the behavior of a system from knowledge of its components, and thus there is considerable interest in formulation of quantitative, predictive mathematical models of cellular functions. The hope, after reliable models describing the behavior of biological systems are constructed, is to predict the effects of drugs or other interventions on the state of diseased cells, and enhance our fundamental understanding of how cells respond to stimuli and regulate their internal environments. On the other hand, for a complex organism such as a human, it is impossible to build or simulate a molecular level model that describes the function of the whole organism. Thus, models at higher levels, treating each cell or each tissue as an entity, are necessary to describe the behavior of these kinds of complex organisms. To connect the behavior of the biological system at different levels, multi-scale modelling will be necessary, with the hope that the simulation results at a lower level, such as molecular interactions, will be able to predict the effect of the lower level events on the higher level function, such as the effect of molecular events on the function of a
whole cell, a whole organ, or even a whole organism.

Synthetic biology has as one of its aims the development of a discipline of biological engineering, allowing systematic control to be exerted over living cells [2,9,11]. This ability would have profound implications, allowing medical interventions to be carried out at the cellular level. Work on building synthetic controllers inside cells has already been done: bistable toggle switches, feedback networks, oscillators, and cell-to-cell communication systems for pattern formation and intercellular coordination have all been implemented in bacteria. Working in bacteria has allowed researchers to explore a simplified environment while developing the background understanding necessary for future biological engineering projects. Prokaryotes are also of interest in their own right, as a platform for the creation of robust, self-replicating, microscopic “machines” programmed to carry out human-designed tasks.

The two fields share a focus on quantitative, mathematical modelling of biological processes, and both require correspondingly quantitative experimental data able to provide accurate reporting of these processes as they proceed inside the cell, and the proper modelling technique able to accurately explain and predict the behavior of a biological system.

1.1.2 Gene expression

With the advancement of biotechnologies, DNA sequencing is becoming easier and faster with time, and nowadays, the genome DNA sequence of biological species of interest can be relatively easily obtained. While the genomic DNA contains the information that determines the behavior of a biological system, the information stored inside DNA must be transferred into the functional units, proteins in most cases, to perform the function that DNA defines. The structure and function of a particular protein is determined by the sequence of the amino acids that form the protein, which is in turn determined by the sequence of DNA which codes the protein. To produce a protein from a certain DNA sequence, a series of biochemical reactions is necessary, called ‘gene expression’, including transcription and translation steps. Since the DNA sequence can be easily obtained, to understand the function of a living cell, how the gene is expressed and how the proteins fold and function will be the two key questions to answer. During my Ph.D. research, I focused on the first question.

The key steps of gene expression have been well studied, as shown in Figure 1.1. Roughly, the process is divided into three steps: First, the RNA polymerase binds to the DNA, aided by some activator proteins or hindered by inhibitor proteins, after which RNA polymerase will switch from a closed complex to an open complex, ready to transcribe the DNA. Lots of research has been focused on this part to understand how proteins expressed from some gene regulate, activate or inhibit the expression of other genes, called the gene expression network. Second, the RNA polymerase moves along the DNA and transcribes the DNA into mRNA. Third, the
ribosome binds to the transcribed mRNA and proteins are produced, after which proteins fold into their native structure and start functioning.

![Gene expression steps diagram](image)

**Figure 1.1: Gene expression steps.**

Qualitative studies of the gene expression process might be sufficient to explain and understand some of the simple behavior of gene expression networks. Quantitative study is requisite to understand the behavior of complex interactions between DNA and proteins, and necessary for accurately predicting the behavior of novel systems. My study is focused on two quantitative aspects of the gene expression process in *E. coli*, gene expression dynamics and noise, which are explained below.

### 1.1.3 Gene expression dynamics in living cells

Living cells grow and divide constantly, sometimes reaching a dynamic steady state in environments such as chemostats (the birth and death or the growth and division reach equilibrium,
Chapter 1. Introduction

Figure 1.2: Cell growth in *E. coli* batch culture.

making the total number of cell constant or the cell size and protein number distribution stay unchanged over time), sometimes without a steady state, in environments such as batch culture, where cells are grown in a fixed volume of nutrient culture medium. The growth of *E. coli* cells in batch culture is thought to be composed of four consecutive phases: lag phase, exponential phase, stationary phase, and death phase, as shown in Figure 1.2. In lag phase, cells are adapting to a new environment; after this, they start to grow and divide with maximum speed, called the exponential phase; the nutrition is then gradually used up, and the cells enter into the stationary phase; after some time, the nutrition is totally used up and the cells start to die, entering the death phase. Studies have been done to monitor the expression of fluorescent proteins over time under microfluidic channels, with the nutrition constantly flushing in and out, similar to the chemostats, where the cells grow and divide at roughly constant rates [12]. However, there has been no systematic study of the time dependent gene expression dynamics and noise in *E. coli* batch culture. The cells in *E. coli* batch culture are usually thought to be in steady state during exponential growth. However, is there a true steady state in *E. coli* batch culture? How do the gene expression dynamics look in *E. coli* batch culture? By systematically studying the time dependent dynamics in *E. coli* batch culture, we address those questions.

Mathematically, gene expression is described and modeled by separating the whole process
into a series of biochemical reactions, each of which can be described either deterministically through differential equations or stochastically through master equations. No matter which method is used, the rate at which a certain biochemical reaction occurs will be determined by a rate constant and the concentration of species involved in the biochemical reaction. For reactions occurring in beakers, the reaction rates are described by the law of mass action - the reaction rate equal to the rate constant multiplied by the concentrations of species on the left side of that reaction, when assuming that the reaction environment is a well mixed environment together with some other conditions: the internal degrees of freedom of each molecule are in thermal equilibrium, and the temperature is constant. Those assumptions might be well satisfied for reactions in beakers. But inside living cells, a large fraction of the total volume is occupied by big molecules: it is inappropriate to assume that the reaction environment is a well mixed environment. Then, these question follows: what are the reaction dynamics in living cells? Particularly, what are the gene expression dynamics in living cells, in such crowded environments? To answer the latter question, I have observed the time dependence of gene expression dynamics in crowded and non-crowded environments.

1.1.4 Noise in gene expression

One widely studied aspect of gene expression dynamics is the stochasticity or “noise” associated with the process. Processes inside the cell are driven by sets of highly coupled biochemical reactions, and the resulting dynamics may be approximated as deterministic [13,14]. There is considerable evidence, however, that biological processes, and gene expression in particular, are subject to significant fluctuations and are not purely deterministic [15–20]: a genetically identical population of bacteria growing under identical conditions, for example, will exhibit a substantial range of gene expression levels even from simplified synthetic constructs [21–25]. This variability arises partly from the fluctuations in reaction rates that occur when small numbers of molecules participate in the reaction steps, called the “intrinsic” noise of the expression of a given gene of interest, and partly from cell-to-cell differences in the background against which the gene of interest is expressed, incorporating effects such as variations in numbers of available enzymes, called the “extrinsic” noise [14–16,18–20,23–29].

Intrinsic noise is inherent to a system, arising from the non-negligible fluctuations in reaction rates in regimes where small numbers of reactant molecules are interacting (as is often taken to be the case inside cells). Extrinsic noise represents random variations in environmental factors outside the system of interest, but varying between cells. Within a single cell, the extrinsic noise affects the gene expression of all the genes in the same way, while the intrinsic noise makes the gene expression from different copies of the same DNA sequences in this cell be different. The distinction between intrinsic and extrinsic effects depends on the extent of the system being
modeled: to a single-gene model, variation in a second regulatory gene may be an extrinsic source of noise, while a two-gene model explicitly representing the production of the regulatory gene would view that same variability as intrinsic to the system.

One example of gene expression noise is shown in Figure 1.3: a fluorescence image of *E. coli* cells expressing Enhanced Green Fluorescence Protein (EGFP). The cells have vastly different sizes and brightness. And even the cells with similar sizes still show large differences in terms of brightness. Note, these cells all have the same DNA sequence, but the numbers of EGFP, thought to be proportional to the brightness of the cell, are different for different cells.

Many studies have been carried out to analyze where the noise comes from. It has been found that extrinsic noise (cell size and plasmid copy number variation) are the dominant source of noise in *E. coli* cells expressing EGFP from unregulated promoters on plasmids [25,30]. The contribution of different reaction steps to the gene expression noise, particularly to the intrinsic noise, has also been well studied, both by modelling and experiments [17]: Studies find that the noise in most of the *Saccharomyces cerevisiae* proteins is proportional to the square root of the mean number of proteins, which is thought to mainly come from transcription due to the birth and degradation of mRNA or promoter fluctuations [22,31]. The study of noise in *Bacillus subtilis* has shown, by varying both transcription and translation rates, that the translation rate is the more dominant factor than the transcription rate [23]. Although those extensive studies have been done to figure out the source of noise in gene expression, there are still some unstudied aspects: What would be the effect of protein maturation steps on the noise in gene expression? How does feedback change the noise in gene expression? I have performed two projects to answer these questions.

### 1.2 Experimental methods on gene expression

To study the gene expression dynamics and noise in living cells, we need to monitor the protein expression levels *in vivo*, without perturbing too much the behavior of living cells. One useful way to monitor the gene expression level of a certain target protein in living cells is to use fluorescent proteins, either by expressing the fluorescent protein together with the target protein of interest, or by fusing the fluorescent protein to the target protein.

#### 1.2.1 Green Fluorescent Protein

Green Fluorescent Protein (GFP), a natural fluorophore derived from the jellyfish *Aequorea victoria*, and related fluorescent proteins, have found wide application in systems and synthetic biology [1–10,32,33]. By providing a fluorescent signal from a protein that may be incorporated into a living, healthy cell, and detected from outside without causing cell death, these proteins
Figure 1.3: *E. coli* Cells expressing EGFP. This image is taken using our Nikon TE2000U fluorescence microscope.
offer a key advantage over other protein monitoring techniques such as Western blotting or microarray technologies, wherein cells must be destroyed in the process of the assay, making real-time observation of living cells impossible. Both systems and synthetic biology have made extensive use of plasmid-borne systems expressing fluorescent proteins as reporters in single-celled organisms, either as deliberately simplified testing grounds for their models [14,15,19,22–24,28,29,34] or as the basis for controllers able to function inside growing cells and alter their behavior [6,7,9,21,35–40].

By mutating GFP, variants of green fluorescent protein have been made, with enhanced fluorescence intensity, different excitation and emission spectra, or enhanced stability.

1.2.2 Fluorescence microscopy

Fluorescence microscopy is one of the popular approaches to detect the fluorescence signals from cells expressing fluorescent proteins. The diagram of a fluorescence microscope is shown in Figure 1.4. Either a laser, a lamp, or an LED can serve as the light source: the laser provides the most stable excitation light with a single wavelength component; the white light lamp has a spectrum covering the whole visible range and needs to be filtered by the excitation filter to provide excitation photons of the proper wavelength, however the light intensity of the white light bulb usually fluctuates with time, making it unsuitable for time series observation of fluorescence levels; colored LEDs provide stable excitation within a certain frequency range, while white LEDs provide stable excitation over the full frequency range. Excitation light is focused by a lens (L), sometimes passed through an excitation filter (F), to the back focal plane of the objective, reflected by a dichroic beamsplitter (D) into the microscope objective (O), and focused by the objective on the specimen. The fluorescent component in the specimen will then be excited, and the fluorescent light will be collected by the same objective, passed through the dichroic mirror and the emission filter, and captured by a CCD camera. The emission filter serves to filter out all the non-fluorescence light, including light scattered by the specimen.

1.2.3 Flow cytometry

Another approach to measure the fluorescence signals from cells is flow cytometry, the diagram of which is shown in Figure 1.5. The solution containing cells is sucked up, mixed with sheath fluid and pushed through the flow channel inside the flow cell. The flow channel is illuminated by a laser or an LED. When a cell passes through the laser spot, the fluorescent proteins inside the cell are excited, and excitation photons are scattered in all directions. The scattered light in the same direction as the laser beam is collected by the PMT marked as “FS”, while the scattered signal in the direction perpendicular to the input laser beam is reflected by the dichroic mirror,
Chapter 1. Introduction

Figure 1.4: Wide-field epi-fluorescence microscopy. Excitation light is focused by a lens (L) at the back focal plane of the objective, reflected by a dichroic beamsplitter (D) into the microscope objective (O), and focused by the objective on the specimen. The fluorescent component of the specimen is excited, and the fluorescent light is collected by the same objective, passed through the dichroic mirror and the emission filter, and captured by a CCD camera.
filtered by a band pass filter and collected by the PMT1 detector: scattered light in both
directions is thought to be related to the cell size. The green fluorescence signal passes through
the first dichroic mirror, is directed by the second dichroic mirror into another band pass filter,
and collected by PMT2. Similarly, yellow and red fluorescence light will be collected by PMT3
and PMT4 respectively. By changing the excitation laser and the filters, flow cytometry can be
configured to detect fluorescent proteins of different colors.

Compared with microscopy, the advantage of using flow cytometry to measure the fluores-
cence signal from cells lies in the ability to analyze a large number of cells in a relatively short
time. With normal settings, we can analyze the fluorescence intensity and cell size of one million
cells in just about tens of minutes. However, the disadvantage is that, as no images of cells are
collected, it is quite possible that some non-cell background particles or air bubbles are treated
as cells, or that two cells pass through the flow channel at the same time and are treated as a
single cell.

Figure 1.5: Diagram of Flow Cytometry [41].

To analyze the data from flow cytometry, I have written a series of MATLAB scripts, which
analyze the data from a single flow cytometry run, from a series of flow cytometry runs, or
from repeated series of flow cytometry runs. On the basic level, functions are written to process
the data from a single flow cytometry run: to convert the raw data format generated by flow
cytometry to MATLAB format; to threshold the data with predefined lower limit and upper limit of forward scattering, side scattering and PMT2 (collects fluorescence from GFP) by running sheath fluid and non-fluorescent cells; to further process the data by eliminating the cells according to certain criteria, such as being more than a certain number of standard deviations away from the mean, or falling in the top or bottom x% of the cell population; to calculate the statistics of forward and side scattering signal, and the fluorescence signals; to plot user-defined histograms or scatter plots. Based on those functions, to analyze the results from a series of flow cytometry runs, I have defined another set of functions to read the label, containing the time, optical density and other information about that particular run, to analyze the series of data using the basic level functions, and to plot the time dependent change of user-defined parameters. One more level up, a class of functions is written to collect, analyze and plot the data from different series of flow cytometry runs.

1.3 Modelling of gene expression

Recent advances in both experimental and computational methods have combined to offer the prospect of developing a quantitative, systematic understanding of biological systems, and this has driven a surge in recent interest in the formulation of mathematical models in biology, especially of the molecular-level details of biochemical kinetics [14,19]. The massive accumulation of experimental data in molecular and cellular biology has made modelling an increasingly vital tool in the effort to understand and analyze biological processes [42]. Many methods of modelling the internal states of cells have been proposed [42,43]. Deterministic modelling treats concentrations of biochemical species as continuous state variables, uses standard chemical kinetics to obtain a set of differential equations describing the system dynamics, and solves these equations to analyze the behavior of the system. Stochastic models treat each species as an integer number of discrete molecules, and treat reactions as discrete, random events. Both of these approaches neglect spatial information, approximating cells as well-mixed systems, and we will maintain this approximation here, while noting that other modelling approaches are being developed to address spatial inhomogeneities in cells [44,45].

1.3.1 Deterministic modelling

If the reaction environment is a well-mixed environment, or assuming this, the reaction rate for each reaction to occur can be written down using the law of mass action: the reaction rate of a chemical reaction is equal to the rate constant multiplied by the product of the concentrations of species that participate in this reaction. With a reaction rate associated with each reaction, the rate of change of the concentration of every species is equal to the rate for this species to
be produced minus the rate of this species being consumed or degraded. Writing down the
differential equation for the rate of change of concentrations of every species, we get a set of
coupled ordinary differential equations (ODEs) governing the time evolution of the system we
are modelling.

With certain initial conditions, we will be able to get the time evolution of the system of
interest by solving the set of differential equations, or we can use those equations to calculate
the steady state of the system, by setting the rate of change of each species to be zero and
solving the algebraic equations. Take a simple biochemical reaction model for example: namely
protein produced and degraded at constant rates, as shown in reactions 1.1.

$$\emptyset \xrightarrow{k_1} P$$

$$P \xrightarrow{k_2} \emptyset$$

(1.1)

The differential equation for this simple model, denoting the concentration of $P$ by $c_p$, can
be written as:

$$\frac{dc_p(t)}{dt} = k_1 - k_2 \cdot c_p(t)$$

(1.2)

Solving this equation with initial concentration $c_p(0) = c_0$, we get the time evolution of the
concentration of protein:

$$c_p(t) = \frac{1}{k_2}(k_1 - (k_1 - k_2c_0)e^{-k_2t})$$

(1.3)

By setting $\frac{dc_p(t)}{dt}$ to be zero in Eq. 1.2, or letting $t \to \infty$ in Eq. 1.3, we can calculate the
protein concentration at steady state: $\frac{k_1}{k_2}$.

1.3.2 Stochastic modelling

As discussed in the first section of this chapter, noise in gene expression is usually not negli-
gible [15–20]. Formulating biochemical models by writing out sets of reactions and translating
them into deterministic sets of ordinary differential equations (ODEs) does not capture this
biological variability, so stochastic modelling approaches are often applied to intracellular bio-
logical models [19,46]. By simulating the fluctuations in small-number regimes, such models can
capture the intrinsic noise in biological systems, while extrinsic noise sources are not explicitly
handled and must be added by other means.

The stochastic model of a biochemical reaction network can be solved, either analytically
using master equations approach, or numerically using the Gillespie algorithm. Due to the
complexity of master equations and stochastic simulations, approximations to both these two
approaches have been developed: for master equations approach, the Langevin equation approach gives an approximate solution; for the Gillespie algorithm, $\tau - \text{leap}$ and other similar approaches approximate the exact Gillespie simulation [47–49]. As master equations can only be solved explicitly for certain linear systems, with the Langevin equation approach suitable for a somewhat larger set of problems, for nonlinear models, computer simulation is a valuable approach to get exact or approximate solutions for those stochastic biochemical reaction models. However, the simulation of stochastic biochemical reaction models is usually very time and resource intensive. To overcome this difficulty, I have developed a systematic approach to reduce the complexity of biochemical reaction models with fast and slow dynamics, and investigated when we need to incorporate gene expression from the genome when modelling the gene expression from plasmids.

**Analytical approach**

The master equations describe the time evolution of the probability distribution for a Markov process. For a biochemical reaction network, an ordinary differential equation can be written down for each state of the system [50]. Basically, the rate of change for the system to be in a certain state will be equal to the rate for the system to enter this state from other states, minus the rate for the system to leave this state. A general form of master equation is shown in Eq. 1.4, where $N_k, N_l$ represent different states of the system of interest, and $T_{kl}, T_{lk}$ represent the transition probabilities for the system from state $N_k$ to $N_l$, and $N_l$ to $N_k$ respectively. For a biochemical reaction network, the state of the system can be fully characterized by the number of molecules. Thus, $N_k$ can be written as $(n_1, n_2, n_3, \ldots)$, where $n_1, n_2, n_3$ represent the number of the first, second, and third species in the model.

\[
\frac{dp(N_k, t)}{dt} = \sum_l (T_{kl}p(N_l, t) - T_{lk}p(N_k, t))
\]  

(1.4)

Take reactions 1.1 for example. The master equation for this system can be written down as shown in Eq. 1.6. Note that the equation involves the number of proteins $n_p$ instead of the concentration of proteins, and the rate constant for the change of number will be denoted as $K_1 = V \cdot k_1$ and $K_2 = k_2$ as shown in Eq. 1.5, where $V$ is the volume of the system.

\[
\frac{dc_p(t)}{dt} = k_1 - k_2 \cdot c_p(t) \Rightarrow \frac{d(n_p(t)/V)}{dt} = k_1 - k_2 \cdot (n_p(t)/V)
\]

\[
\Rightarrow \frac{dn_p(t)}{dt} = k_1 \cdot V - k_2 \cdot n_p(t) \Rightarrow \frac{dn_p(t)}{dt} = K_1 - K_2 \cdot n_p(t)
\]

(1.5)
Chapter 1. Introduction

\[
\begin{align*}
\frac{dp(n_p, t)}{dt} &= K_1 p(n_p - 1, t) + K_2 (n_p + 1) p(n_p + 1, t) - K_1 p(n_p, t) - K_2 n_p p(n_p, t) \\
\frac{dp(n_p + 1, t)}{dt} &= K_1 p(n_p, t) + K_2 (n_p + 2) p(n_p + 2, t) - K_1 p(n_p + 1, t) - K_2 (n_p + 1) p(n_p + 1, t)
\end{align*}
\]

The master equations for this simple system cannot be solved directly, as there are an infinite number of equations for the infinite number of possible states of the system. To solve the master equations, the generating functional approach can be used [50]. In this approach, both sides of the master equations will be multiplied by \( z_1^{n_1} z_2^{n_2} z_3^{n_3} \cdots \). Summing over all the master equations and collecting the terms, we can derive the equation for the generating functional:

\[
G(z, t) = \sum_{n_i=0}^{\infty} z_1^{n_1} z_2^{n_2} \cdots p(N, t)
\]

where \( N = (n_1, n_2, n_3 \cdots) \), \( z = (z_1, z_2, z_3 \cdots) \), and \( z_i \) is an arbitrary variable.

To solve Eq. 1.6, we multiply both sides of the master equations by \( z^{n_p} \), and sum over them to get the equation for \( G(z, t) = \sum_{n_p=0}^{\infty} z^{n_p} p(n_p, t) \):.

\[
\frac{\partial G(z, t)}{\partial t} = K_1 (z - 1) G(z, t) - K_2 (z - 1) \frac{\partial G(z, t)}{\partial z} 
\]

Partial differential equations are difficult to solve analytically. However, often we are more interested in the steady state behavior of a system. Setting the time dependent term on the left side of Eq. 1.8 to zero, and solving the ordinary differential equation, we get the steady state distribution of the system. The result is shown in Eq. 1.9, and corresponds to a Poisson distribution with mean \( \frac{K_1}{K_2} \).

\[
G(z) = e^{K_1 \frac{z}{K_2}} e^{-\frac{K_1}{K_2}} = e^{-\frac{K_1}{K_2}} \sum_{n=0}^{\infty} z^n \left( \frac{K_1}{K_2} \right)^n
\]

\[
\Rightarrow p(n) = \frac{\left( \frac{K_1}{K_2} \right)^n}{n!} e^{-\frac{K_1}{K_2}}
\]

Sometimes, we are more interested in the statistical parameters, such as mean and standard deviation, and using some tricks, we can get these and higher order moments without solving for the distribution. The generating function has the property that if we take the derivatives of the generating function, we get the moments from the derivatives:
\[
\frac{\partial G(z,t)}{\partial z} = \sum_{n=0}^{\infty} nz^{n-1}p(n,t) \Rightarrow <n_p> = \frac{\partial G(z,t)}{\partial z} \bigg|_{z=1}
\]

\[
\frac{\partial^2 G(z,t)}{\partial^2 z} = \sum_{n=0}^{\infty} n(n-1)z^{n-1}p(n,t) \Rightarrow <n_p^2> = \frac{\partial^2 G(z,t)}{\partial^2 z} \bigg|_{z=1} + <n_p> \quad (1.10)
\]

Taking the derivative of Eq. 1.8, letting \( z = 1 \), and using Eq. 1.10, we can derive the equations for the time evolution of the moments:

\[
\frac{d<n_p>}{dt} = K_1 - K_2 <n_p>
\]

\[
\frac{d<n_p^2>}{dt} = K_1 + 2K_1<n_p> - 2K_2<n_p^2> + K_2<n_p> \quad (1.11)
\]

Solving these equations, we can get the time evolution of statistical parameters. The steady state solution of the mean number and the second order moment is given by:

\[
<n_p> = \frac{K_1}{K_2}
\]

\[
<n_p^2> = \frac{K_1}{K_2} + \frac{K_1^2}{K_2^2} \quad (1.12)
\]

**Numerical approach**

Analytical solutions for the chemical master equations only exist for the case of monomolecular reactions or some simple linear systems [50]. For nonlinear systems, we can approximate the equation with a Langevin equation or solve the master equation numerically. The Gillespie Algorithm gives an exact solution to the master equation, based on the distribution shown in Eq. 1.13, which gives the probability distribution for the \( \mu \)th reaction to occur in time \( t \).

\[
p(\mu,t)dt = r_\mu e^{-t\sum_j r_j} dt \quad (1.13)
\]

This distribution can be broken into two separate distributions, as shown in Eq. 1.14 and Eq. 1.15, using which the next reaction time \( t \) and the next reaction number \( \mu \) can be generated separately. This method is named the first-reaction method.

\[
p(t)dt = e^{-t\sum_j r_j} \sum_j r_j dt \quad (1.14)
\]

\[
p(\mu) = \frac{r_\mu}{\sum_j r_j} \quad (1.15)
\]
Given a state of the system, the reaction rate $r_j$ is calculated for each forward and backward reaction, using which we can generate a next reaction time $t$ using Eq. 1.14. With the next reaction time chosen, we then choose which reaction to occur using the probability distribution Eq. 1.15, after which, the state of the system is updated. This process is repeated step by step until a certain time is reached.

A number of software programs exist for implementing this method. I use BioNetS (Biochemical Network Stochastic Simulator) to do all my simulations [51]. The BioNetS simulation result of reactions 1.1 is shown in Figure 1.6. BioNetS is able to simulate reactions using either the first reaction method shown above [52,53], a continuum approximation treating the species numbers as continuous rather than discrete quantities, or a hybrid of the two. In my research, I have used only the exact, discrete method for simulating my models.

![Figure 1.6: Result of BioNetS simulation. Shown is protein number vs time, with the center line representing the average number of protein.](image)

1.3.3 Fitting and optimization.

Two different methods have been used to fit user defined functions to experimental or modelling results: the fminsearch function in MATLAB, and the simulated annealing approach.

fminsearch uses the simplex search method. If the vector length of the parameter to be
fitted is n, a simplex in n-dimensional space is formed by n+1 points, within which there will be a worst point that gives the biggest error. In two-dimensional space, a simplex is a triangle; in three-dimensional space, it is a pyramid. At each step of the search, a new point in or near the current simplex is generated, usually by reflecting the worst point according to the central point of all the points on the simplex. The function’s value at this new point is compared with the function’s values at the vertices of the simplex and, based on the function’s value at this point, a new point is calculated and used to replace the worst point on the simplex. Those steps are repeated until the diameter of the simplex is less than the specified tolerance. One useful way to avoid local minima using this approach is to start the minimization from several different initial sets of points, far away from each other.

Simulated annealing is a probabilistic global optimization technique formulated in analogy with the cooling of real materials. When a physical substance is gradually cooled (“annealed”) in an effort to reach the global minimum energy state, it starts at high temperatures where the system is relatively free to visit higher-energy states, but as it is slowly cooled, lower energies are increasingly favored, until eventually it solidifies into a minimum state that should be a good approximation to the global minimum; if the substance is cooled too rapidly, it can instead be caught in a local minimum far from the global minimum. In simulated annealing, any desired criterion may be used to represent the energy (here, we use the value generated by the error function), and the temperature represents the probability of making transitions between two states with energy difference $\Delta E$, according to the Boltzmann distribution: $P(\text{transition}) = \exp(-\Delta E/kT)$. At each step, a neighboring point is generated randomly. The energy of this point is calculated, for our case, using the pre-defined error function. The $P(\text{transition})$ is compared to a random number between 0 and 1: if $P(\text{transition})$ is greater than this random number, the search will start from this new point, otherwise, the search will start from the old point again. This step is repeated until the user required requirement is satisfied. A number of simulated annealing implementations are freely available for download; we use one by Corpus [54] that provides a script in MATLAB (The MathWorks).

The fminsearch function is faster than the simulated annealing approach, but might lead to a local minimum. It is suitable when the error function surface is not too rough. No matter which approach is used, the first step of the optimization problem is to define an error function, which, when given a set of parameters, will generate an error value by comparing the calculated result using that set of parameters and the experimental or modelling result which we want to fit. The error functions that are used in this thesis are defined to be the sum or weighted sum of the square of the difference between each calculated point and data point, or the percentage difference between each calculated point and data point.
1.3.4 Distributed computation

I wrote a simple but robust script using Unix to run BioNetS simulations on about 100 Mac computers: In the initialization stage, the input files for BioNetS together with the simulation program (a Unix executable) are copied using the scp command over to these Mac computers, and the simulations are started remotely. After the initialization, the script loops through all those computers to check the status of the simulations by checking the output files of the simulations. If the simulation is finished, the result is copied back to the home computer, and a new simulation is copied to and started on that computer. The loop is run over and over, until all the simulations are finished.

1.3.5 Stochastic simulation of a population of cells

In laboratories, E. coli cells are grown under many different conditions, generally characterized into two different types: closed environments, where nutrition is added at the beginning and consumed over time, and open environments, where nutrition is constantly pumped in and out. In a closed environment, the number of cells grows exponentially with time for the first few hours, while in an open environment, at first the number grows exponentially, and then stays approximately constant, as a certain number of cells are also pumped out of the system when the nutrition is pumped out. To model those laboratory cells, several different ways can be used to represent the population of cells. No doubt, the most realistic way is to track each cell individually and track both sibling cells when cell dividing. The disadvantage of this method is that tracking all cells will be very resource-intensive, since, if the death rate is low, the number of cells will grow exponentially: supposing initially we have \( m \) cells, after \( n \) generations, the total number of cells will become near \( m \cdot 2^n \). As \( n \) becomes large, the number of total cells can easily become intractable.

To overcome this difficulty, we can simulate only a small part of the cells and track only one sibling cell at each cell division, representing the whole cell population. Using this method, the number of cells we keep tracking is always the same as the initial number of cells we simulate, thus we will not have the problem of an intractable number of cells. However, whether the selected part we have chosen to simulate will represent the whole population faithfully will depend on the specific situations we are interested in. Generally, if the properties of the population of cells spread over a wide range, and the cells divide randomly, this method might give a good representation of the whole population. The example we look at in Section 4 shows that this method gives a very similar prediction compared to the tracking all cells method.

The simplest method to simulate the property of a population of cells is to simulate only one in silico cell over an extensive time period, and calculate the statistics from the values sampled
from different time points. This method is only appropriate when the population of cells we want to model is in a steady state, so that the statistics calculated using the data points at different times are the same as the statistics calculated by simulating a bunch of cells which are in the steady state. Moreover, when using this approach, we have to make sure the sample frequency is low enough that the correlations between the sampled points are low, so that we can approximate the values at different time points to be the values from different cells. Otherwise, we will get very biased statistical parameters. This is the most simple, efficient, and widely used method, but we should keep in mind that this method can only describe the behavior of a system in a steady state.

1.4 The focus of my research

Although the gene expression process has been studied for many years, the quantitative aspects of gene expression dynamics and noise have only been studied in recent years. Many aspects of gene expression noise and gene expression dynamics have been studied, including the contribution of promoter binding step, transcription step or translation step to the noise in gene expression, or the dynamics of gene expression under constant nutrition flow. There are, however, still many unanswered questions. What would be the effect of protein maturation step, which is necessary before the proteins can be detected, on the noise in gene expression? Is there any way to speed up the simulation by reducing the complexity of the model while still preserving the behavior of the model? Will the gene expression from the background genome greatly affect the gene expression from single genes on plasmids? What would be the effect of feedback on the intrinsic and extrinsic noise in gene expression? How does macromolecular crowding, usually present in living cells, change the reaction dynamics, when breaking the law of mass action? What are the time dependent gene expression dynamics in \textit{E. coli} batch culture? All those questions have not been answered by previous studies. To answer those questions, I combine experiment, modelling, theoretical analysis and computer simulation approaches. Below is a list of the projects I have finished to answer those questions:

1.4.1 Simplification of stochastic chemical reaction models with fast and slow dynamics

Biological systems often involve chemical reactions occurring in low-molecule-number regimes, where fluctuations are not negligible and thus stochastic models are required to capture the system behavior. The resulting models are generally quite large and complex, involving many reactions and species. For clarity and computational tractability, it is important to be able to simplify these systems to equivalent ones involving fewer elements. While many model sim-
plification approaches have been developed for deterministic systems, there has been limited work on applying these approaches to stochastic modelling. Here, we describe a method that reduces the complexity of stochastic biochemical network models, and apply this method to the reduction of a mammalian signalling cascade and a detailed model of the process of bacterial gene expression. Our results indicate that the simplified model gives an accurate representation for not only the average numbers of all species, but also for the associated fluctuations and statistical parameters. This work will be discussed in Chapter 2.

1.4.2 Increasing the efficiency of bacterial transcription simulations: when to exclude the genome without loss of accuracy

When considering the transcription of a gene of interest inside an *E. coli* cell, the number of RNA polymerases (RNAPs) that are available to initiate transcription is a key point, and one that is not currently accessible experimentally; experimental investigations have to date yielded only average total numbers of RNAPs per cell. We have previously investigated a method of simulating the effect of the genome on the availability of RNAPs, but the method introduces a significant number of additional reactions into the system, and thus comes at the cost of increased computational time. Here, we show that for some parameter regimes, the effect of the host cell genome on the transcription of a gene from a plasmid-borne promoter may be negligible, allowing one to simulate the system without the additional computational load of the genome. The key parameter is the on-rate of RNAP binding to the promoter ($k_{on}$), and we compare the total number of transcripts generated as a function of this rate constant, for two versions of our gene expression model, one incorporating the host cell genome and one excluding it. By sweeping parameters, we identify the $k_{on}$ range for which the difference between the genome and no-genome models drops below 5%. We assess the effect of the genome over a four-dimensional parameter space, considering: $24 \text{ min} \leq$ bacterial doubling time $\leq 100\text{min}$; $10 \leq$ plasmid copy number $\leq 1000$; $2\text{min} \leq$ mRNA half-life $\leq 14\text{min}$; and $10 \text{bp} \leq$ gene length $\leq 104\text{bp}$. A simple MATLAB user interface generates an interpolated $k_{on}$ threshold for any point in this range. Exclusion of the genome is shown to yield less than 5% difference in transcript numbers over wide ranges of values, and computational speed is improved by two to 24 times by excluding explicit representation of the genome. This work will be discussed in Chapter 3.

1.4.3 Effect of protein maturation on the noise in gene expression

Fluorescent proteins are frequently used as reporters for gene expression in living cells, either by being expressed in tandem with a protein of interest, or through the creation of fusion proteins. The data yielded by the fluorescence output is of considerable interest in efforts to
formulate quantitative models of cellular behavior underway in fields such as systems biology and synthetic biology. An often neglected aspect of these proteins, however, is their maturation: before a fluorescent protein can generate a fluorescent signal, it must mature through a series of steps (folding, cyclization, and oxidation) that may take from many minutes to over a day. The presence of these maturation steps creates a distinction between the observed gene expression profile and the actual profile. We examine this effect through a simplified gene expression model and conclude that fluorescent protein maturation can have significant effects on estimates of both the mean protein levels and the variability in gene expression. The model shows that in many regimes, the observed variability will be increased by the maturation process, but indicates the existence of regimes in which the observed variability will actually be less than the true variability of the target protein. The latter effect arises from a low-pass filtering effect introduced by the chain of maturation reactions. The results suggest that the maturation of fluorescent proteins should be taken into account when using such proteins as quantitative indicators of gene expression levels. This work will be discussed in Chapter 4.

1.4.4 Time series gene expression dynamics in E. coli batch culture

Extensive work on building synthetic gene systems inside cells and quantitative study of their gene expression has been done. However, no systematic studies have been done on the time dependent dynamics of gene expression and the gene expression noise. Here we systematically studied the time dependent gene expression dynamics of different simple unregulated and regulated synthetic gene systems using flow cytometry. Contrary to the usual assumption, we found that there is no steady state in the gene expression level for up to 12 hours. Both the mean gene expression level and the variability among a genetically identical population change as a function of time even in the exponential growth phase for both regulated and unregulated systems. We also showed that the time dependent behaviors of unregulated and regulated gene expression are different, and we cannot compare the noise of the regulated and unregulated plasmids without specifying which time point we are interested in. A simple mathematical model, where gene expression is coupled with the cell growth and division, was formulated which captures the qualitative features of time dependent gene expression dynamics. This work will be discussed in Chapter 5.

1.4.5 Effect of feedback on the noise in gene expression

Negative feedback, broadly present in natural systems, is usually thought to be able to stabilize the system and reduce the noise in that system, while positive feedback is often thought to make the system unstable and more noisy. However, when applying this rule, by modelling, to
the feedback circuits in gene expression, exceptions have been found: the negative feedback can actually increase the noise in the model. By comparing the noise of different species between models with and without feedback, we find that the increase of the noise in the model with feedback is due to the coupling of protein noise into its own input - the plasmid copy number, which in turn is propagated through the biochemical reaction steps and added to the noise in the protein number. On the other hand, negative feedback is found, also by modelling, to be able to suppress the external variation. Since the extrinsic noise is usually the dominant source of noise in a system, the suppression of external noise due to feedback, if larger than the increase of intrinsic noise, will result in a total reduction of noise. To verify the modelling result and our theory that feedback will be able to increase the intrinsic noise but decrease the extrinsic noise, I have designed an experiment using two colored proteins, enabling us to separate the intrinsic noise from the extrinsic noise, and compare directly the intrinsic and extrinsic noise from the model with and without feedback [24]. This experiment is currently being carried out by some other students in the lab. This work will be discussed in Chapter 6.

1.4.6 Macromolecular crowding effect on the dynamics in gene expression

Cells are crowded with macromolecules: in E. coli cells, around 30% of the volume is occupied by macromolecules. The crowding, shown by modelling, will slow down the diffusion of molecules inside the cell and make a large fraction of the volume inaccessible to big molecules, a phenomenon called the volume exclusion effect. Since, in crowded environments, the “well-mixed environment” assumption made when using the law of mass action will not hold, the law of mass action will not correctly describe the reaction dynamics in such environments. Based on the modelling results, many formulas, mostly representing the time-dependent reaction rates, have been used to describe and predict the time dependent biochemical reaction dynamics in crowded environments. Modelling of gene expression in crowded environments has also shown different behavior than the dynamics in well mixed environments. However, there is no experimental observation of what is the effect of macromolecular crowding on the dynamics of gene expression. What would be the appropriate formula for describing the gene expression dynamics in crowded environments in order to predict both dynamics and noise accurately and efficiently? To answer this question experimentally, I observe the expression of EGFP over time under different crowding conditions using in vitro kits. The results of my experiments and ODE models show that the crowding makes the transcription and translation rate slow down with time, consistent with previous modelling predictions. Further modelling work will be carried on by Prof. Raymond Kapral’s group, using detailed simulations that track the position of each molecule individually. This work will be discussed in Chapter 7.
Chapter 2

Simplification of stochastic chemical reaction models with fast and slow dynamics

The work in this chapter has been published in [55] and [56]. The original publication is available at www.springerlink.com.

2.1 Introduction

Formulating biochemical models by writing out reaction systems and translating them into deterministic sets of ordinary differential equations (ODEs) does not capture this biological variability, so stochastic modelling approaches are often applied to intracellular biological models [46]. By simulating the fluctuations in small-number regimes, such models can capture the intrinsic noise in biological systems, while extrinsic noise sources are not explicitly handled and must be added by other means. Stochastic models are computationally expensive, and cellular processes tend to be complicated, with detailed models quickly growing to include large numbers of interacting species and their associated chemical reactions. Beyond their computational expense, such models are difficult to understand and to analyze, and thus any reduction in model complexity is desirable, provided it can be achieved without substantially altering the system’s behaviour [57].

Three main methods have been developed for model reduction: lumping, sensitivity analysis, and time-scale analysis [58]. Lumping is useful when only limited measurements and information are available about specific reaction kinetics and detailed chemical composition [59]. Sensitivity analysis seeks to determine and eliminate insignificant reactions and species on the basis of their impact on designated important species [60]. As the number of important species increases,
sensitivity analysis is less likely to provide substantial model order reduction, and thus it is often used in conjunction with the quasi steady-state approximation. Time-scale analysis identifies the different time scales over which species react, and the fast-timescale reactions and species are assumed to be at steady state \[61,62\]. The slow variables remain in the reduced model, but their motion is constrained to a lower dimensional space on the basis of their relationship to the fast variables. Thus, order reduction is possible since the differential equations for the fast species are replaced by algebraic relations. This is the basis for the quasi-steady state approximation, mathematically formalized by singular perturbation theory. Timescale analysis requires manual manipulation of the model and significant \textit{a priori} knowledge of the behaviour of various species; in practice, this means that it is feasible to simplify only compact models. However, all three methods have been developed for deterministic modelling, whereas reduction approaches for stochastic models remain largely unexplored.

On the other hand, approaches have been developed to speed up stochastic simulations without an explicit reduction of the model, including the quasi-steady-state approximation approach \[63–65\], the \(\tau\)-leaping approach \[66\] and the R-leaping approach \[49\]. In the quasi-steady-state approximation approach, the intermediate species are first identified and the conditional probability of the intermediate species on the primary species is assumed to be time invariant, based on which, the complexity of the master equations is reduced and a modified Gillespie algorithm is used to simulate the reduced master equations. The \(\tau\)-leaping approach attempts to accelerate the stochastic simulations by using a special Poisson approximation to leap over sequences of noncritical reaction events, while the R-leaping approach attempts to simulate a predefined number of reactions instead of one reaction in each step. All those approaches are useful when our goal is to speed up the simulations, but none of them can give biological meaningful reduced models.

Here, we propose a systematic method for reducing the complexity of stochastic biochemical models while keeping their statistical properties unchanged based on time scale analysis \[55, 67\]. Signalling cascades and gene expressions are examples of this complexity, involving many participating species interacting in highly branched networks \[57\]. We apply the method to these two models, reducing them to substantially fewer reactions and species while maintaining the same overall behavior.

2.2 Methods

In this section, we address the algorithm used to find the simplified reaction system based on a reduction method designed for non-isothermal sytems \[67\]; we then extend this technique by adding a final step in which the system is returned to a set of biochemical reactions, suitable
for stochastic simulation by any desired method. The key problem of time scale analysis lies in
determining the lower dimensional space on which the nonlinear differential equations for the
slow species are constrained to evolve. This whole procedure is automated using Mathematica
(Wolfram Research; Mathematica script available online). The methodology is as follows:

Consider a spatially homogeneous biochemical reaction system with relatively fast and slow
reactions. The reaction dynamics can be written as differential equations:

\[
\frac{dx}{dt} = v(x) \cdot r(x)
\]  

(2.1)

where \( x \) denotes the species vector, \( v(x) \) is the stoichiometric matrix and \( r(x) \) is the reaction
rate vector.

**Example:** To illustrate the method, we will use a simplified model of gene transcription in
the bacterium \( E. coli \) [68] as a running example, showing for each step in the process how the
method is applied to this system. As shown in reactions (2.2) to (2.5) below, transcription
starts with the binding of RNA polymerase (A) to a region of the DNA (B). This RNA/wound-
DNA structure is referred to as the closed complex (C). Then the DNA is unwound and becomes
single-stranded ("open") in the vicinity of the binding site. This RNA/unwound-DNA structure
is called the open complex (D). The RNA polymerase will then move along DNA and transcribe
the information from DNA to form mRNA (E), which then degrades (vanishing from the model)
at a specified rate.

\[
A + B \xrightleftharpoons[k_{-1}]{k_1} C
\]  

(2.2)

\[
C \xrightarrow{k_2} D
\]  

(2.3)

\[
D \xrightarrow{k_3} A + B + E
\]  

(2.4)

\[
E \xrightarrow{k_4} \emptyset
\]  

(2.5)

where \((k_1, k_{-1}, k_2, k_3, k_4) = (4 \times 10^{-6}, 1, 1, 5.79, 5 \times 10^{-4})\)

The differential equations for this system can be written as:

\[
\begin{pmatrix}
\dot{A} \\
\dot{B} \\
\dot{C} \\
\dot{D} \\
\dot{E}
\end{pmatrix} =
\begin{pmatrix}
-1 & 1 & 0 & 1 & 0 \\
-1 & 1 & 0 & 1 & 0 \\
1 & -1 & -1 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 \\
0 & 0 & 0 & 1 & -1
\end{pmatrix}
\begin{pmatrix}
k_1AB \\
k_{-1}C \\
k_2C \\
k_3D \\
-k_4E
\end{pmatrix}
\]  

(2.6)

where \( A, B, C, D, \) and \( E \) represent the concentrations of each species, and the right-hand side
has been written as a column vector containing each of the five reaction rates in the system, pre-
multiplied by a stoichiometry matrix representing how each species is affected by each reaction.
For example, the entries in the first row of the stoichiometric matrix indicate that species A loses one molecule when the forward reaction (2) occurs, gains one when the reverse reaction (2) occurs, and gains one when reaction (4) occurs. In general, for a system with \( r \) reactions and \( s \) species, the stoichiometric matrix will be \( s \times r \); this matrix is post-multiplied by the \( r \times 1 \) column of reaction rates, yielding the desired \( s \times 1 \) vector of rates of change for the species. For our example, \( r = s = 5 \), since the forward and reverse directions in (2) are treated as distinct reactions.

**[Step 1]**. First we need to identify fast and slow reactions, which can be selected by examining the rate constants for the target system. By treating different sets of reactions as slow, the model may be reduced to varying degrees, resulting in reduced models with different final numbers of reactions.

After finding the fast and slow reactions, we can separate the vector \( r(x) \) into fast ones \( r_f(x) \) and slow ones \( r_s(x) \), and pick out the corresponding \( v_f(x) \) and \( v_s(x) \). Then we get differential equations:

\[
\frac{dx}{dt} = f(x) + v_f(x) \cdot r_f(x) \quad (2.7)
\]

where \( f(x) = v_s(x) \cdot r_s(x) \) is the dynamics for slow reactions. \( v_f(x) \) will be a \( s \times p \) matrix, where \( s \) is the number of species in the model and \( p \) is the number of fast reactions.

Note that the user is free to choose the set of fast reactions based on any desired criterion. The most common basis for selection would be either the rate constants associated with each reaction, or the net reaction rates. Reaction rates, taking into account both the rate constant and the abundance of the participating species, can vary over time, which may in some cases make it difficult to identify a single rate as a basis for selection. Empirically, we have found that better matches between the full and reduced models are obtained by choosing based on rate constants than by using the reaction rates. However, we have no rigorous proof that one method or the other is superior, and users may find that their model reduction works better by choosing fast reactions based on reaction rates, or through some other scheme of their own devising.

*Example (continued)*: Here, we select two reactions as slow and the remaining three as fast, since \( k_1 \) and \( k_4 \) are much smaller than all other rate constants. Picking out the columns corresponding to fast reactions from \( v(x) \) (indicated in bold on the left, below), we have:
Chapter 2. Simplification of stochastic chemical reaction models

\[
\begin{pmatrix}
-1 & 1 & 0 & 1 & 0 \\
-1 & 1 & 0 & 1 & 0 \\
1 & -1 & -1 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 \\
0 & 0 & 0 & 1 & -1
\end{pmatrix}
\Rightarrow
\begin{pmatrix}
1 & 0 & 1 \\
1 & 0 & 1 \\
1 & -1 & 0 \\
0 & 1 & -1 \\
0 & 0 & 1
\end{pmatrix}
\] (2.8)

Selecting the corresponding terms in \( r_f(x) \) and putting the remaining terms into \( f(x) \), we obtain:

\[
\begin{pmatrix}
\dot{A} \\
\dot{B} \\
\dot{C} \\
\dot{D} \\
\dot{E}
\end{pmatrix}
=
\begin{pmatrix}
-k_1 AB & -k_1 AB \\
-1 & 0 & 1 & -1 & 0 & 1 \\
-k_4 E & 0 & 1 & -1 & 0 & 1
\end{pmatrix}
\times
\begin{pmatrix}
k_{-1} C \\
k_2 C \\
k_3 D
\end{pmatrix}
\] (2.9)

[Step 2.] To find the slow manifold, we need to make the stoichiometric vectors of the fast reactions linearly independent, which is equivalent to ensuring that the generalized stoichiometric matrix for the fast dynamics, \( v_f(x) \), has full column rank. If \( v_f(x) \) has full column rank, set \( v'_f(x) = v_f(x) \) and \( r'_f(x) = r_f(x) \).

Otherwise we need select the independent columns of \( v_f(x) \) as \( v'_f(x) \), and let

\[
r'_f(x) = \left( [v_f(x)]^T [v'_f(x)] \right)^{-1} [v_f(x)]^T [v_f(x)] r_f(x)
\]

(2.10)

The differential equations will become:

\[
\frac{dx}{dt} = f(x) + v'_f(x) \cdot r'_f(x)
\]

(2.11)

Example (continued): Since \( v_f(x) \) has full column rank of 3, \( v'_f(x) \) will be identical to \( v_f(x) \) and \( r'_f(x) \) will be identical to \( r_f(x) \).

[Step 3.] Then we need to make the elements in \( r'_f(x) \) independent. To test whether the elements in \( r'_f(x) \) are independent, we must calculate the Jacobian matrix of the vector \( r'_f(x) \). The number of elements of \( r'_f(x) \) will be \( p' \), and let \( s \) be the number of species. We will have:

\[
J = 
\begin{pmatrix}
\frac{\partial r'_{f_1}(x)}{\partial x_1} & \cdots & \frac{\partial r'_{f_1}(x)}{\partial x_j} & \cdots & \frac{\partial r'_{f_1}(x)}{\partial x_s} \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
\frac{\partial r'_{f_k}(x)}{\partial x_1} & \cdots & \frac{\partial r'_{f_k}(x)}{\partial x_j} & \cdots & \frac{\partial r'_{f_k}(x)}{\partial x_s} \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
\frac{\partial r'_{f_{p'}}(x)}{\partial x_1} & \cdots & \frac{\partial r'_{f_{p'}}(x)}{\partial x_j} & \cdots & \frac{\partial r'_{f_{p'}}(x)}{\partial x_s}
\end{pmatrix}
\]

(2.12)
Define the row rank of the Jacobian matrix as \( p^* = \text{rank}(J) \). If \( p^* = p' \), let \( \mathbf{v}_r^*(x) = \mathbf{v}_r'(x) \) and \( \mathbf{r}_r^*(x) = \mathbf{r}_r'(x) \). Otherwise we can construct a nonsingular matrix \( \mathbf{E}(x) \) such that \( \mathbf{r}_r^*(x) = \mathbf{E}(x) \cdot \mathbf{r}_r'(x) \) has the first \( p^* \) rows with independent scalar functions, and the last \( p' - p^* \) rows identically equal to 0. The construction of \( \mathbf{E}(x) \) is also automated in our Mathematica script using the following method. First, find the independent rows from \( \mathbf{r}_r'(x) \) and construct the first \( p^* \) rows of \( \mathbf{E}(x) \) to pick out the independent scalar functions from \( \mathbf{r}_r'(x) \). Then to zero out the last \( p' - p^* \) rows of \( \mathbf{r}_r^*(x) \), the dependent scalar functions are decomposed in terms of the independent scalar functions in \( \mathbf{r}_r^*(x) \). By putting -1 in the last \( p' - p^* \) rows of \( \mathbf{E}(x) \) to pick out the dependent rows and putting the corresponding decomposition coefficients inside, we can make the last \( p' - p^* \) rows of \( \mathbf{E}(x) \) to be zero. From the \( \mathbf{E}(x) \) we construct, we get a transformed stoichiometric matrix \( \mathbf{v}_r^*(x) = \mathbf{v}_r'(x) \cdot \mathbf{E}(x)^{-1} \). Since the last \( p' - p^* \) rows of \( \mathbf{r}_r^* \) are zero, we can discard the last \( p' - p^* \) rows of \( \mathbf{r}_r^* \) and the last \( p' - p^* \) columns of \( \mathbf{v}_r^* \). The differential equation will become:

\[
\frac{dx}{dt} = f(x) + \mathbf{v}_r^*(x) \cdot \mathbf{r}_r^*(x).
\]

(2.13)

Example (continued): The Jacobian matrix of \( \mathbf{r}_r'(x) \) will be:

\[
J = \begin{pmatrix}
\frac{\partial r_1'(x)}{\partial A} & \frac{\partial r_1'(x)}{\partial B} & \frac{\partial r_1'(x)}{\partial C} & \frac{\partial r_1'(x)}{\partial D} & \frac{\partial r_1'(x)}{\partial E} \\
\frac{\partial r_2'(x)}{\partial A} & \frac{\partial r_2'(x)}{\partial B} & \frac{\partial r_2'(x)}{\partial C} & \frac{\partial r_2'(x)}{\partial D} & \frac{\partial r_2'(x)}{\partial E}
\end{pmatrix}
= \begin{pmatrix}
0 & 0 & k_{-1} & 0 & 0 \\
0 & 0 & k_2 & 0 & 0 \\
0 & 0 & 0 & k_3 & 0
\end{pmatrix}
\]

(2.14)

The row rank of the Jacobian matrix of the vector \( \mathbf{r}_r'(x) \) is \( p^* = 2 \). It is less than \( p' = 3 \), thus we need construct a nonsingular matrix \( \mathbf{E}(x) \) to make \( \mathbf{r}_r^*(x) = \mathbf{E}(x) \cdot \mathbf{r}_r'(x) \) have the first 2 rows with independent scalar functions and the last row of \( \mathbf{r}_r^*(x) \) be zero. Clearly the first and third rows of \( \mathbf{r}_r^*(x) \) are independent scalar functions, while the second row can be written as a function of the first. In this case, we can construct a matrix to pick out the first and third items from \( \mathbf{r}_r'(x) \) by putting \((1 0 0)\) as the first row of \( \mathbf{E}(x) \) and \((0 0 1)\) as the second row. To zero out the last row of \( \mathbf{r}^*(x) \), we multiply the second row by \( k_{-1}/k_2 \) and subtract it from the first row, creating a third row of \( \mathbf{E}(x) \) equal to \((1 \frac{k_{-1}}{k_2} 0)\), so that the desired matrix becomes

\[
\mathbf{E}(x) = \begin{pmatrix}
1 & 0 & 0 \\
0 & 0 & 1 \\
1 & -\frac{k_{-1}}{k_2} & 0
\end{pmatrix}
\]

(2.15)

In this case, \( \mathbf{E}(x) \) contains only constants, but in general it may be a function of the state variables \( x \). Applying \( \mathbf{E}(x) \) on \( \mathbf{v}_r'(x) \) and \( \mathbf{r}_r'(x) \), we will get:
\[ r_t^*(x) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & -\frac{k_{-1}}{k_2} & 0 \end{pmatrix} \times \begin{pmatrix} k_{-1}C \\ k_2C \\ k_3D \end{pmatrix} = \begin{pmatrix} k_{-1}C \\ k_2C \\ k_3D \end{pmatrix} \] (2.16)

\[ E^{-1}(x) = \begin{pmatrix} 1 & 0 & 0 \\ \frac{k_2}{k_{-1}} & 0 & -\frac{k_2}{k_{-1}} \\ 0 & 1 & 0 \end{pmatrix} \] (2.17)

\[ v_t^*(x) = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ -1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 \\ \frac{k_2}{k_{-1}} & 0 & -\frac{k_2}{k_{-1}} \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ -1 & -\frac{k_2}{k_{-1}} & 0 \\ \frac{k_2}{k_{-1}} & -1 & -\frac{k_2}{k_{-1}} \\ 0 & 1 & 0 \end{pmatrix} \] (2.18)

Discarding the zeros in \( r_t^*(x) \) and the corresponding columns in \( v_t^*(x) \), we will obtain the differential equations written as:

\[
\begin{pmatrix}
\dot{A} \\
\dot{B} \\
\dot{C} \\
\dot{D} \\
\dot{E}
\end{pmatrix} = \begin{pmatrix}
-k_1AB \\
-k_1AB \\
k_1AB \\
o \\
-k_4E
\end{pmatrix} = \begin{pmatrix}
1 & 1 & 0 \\
1 & 1 & 0 \\
-1 & -\frac{k_2}{k_{-1}} & 0 \\
\frac{k_2}{k_{-1}} & -1 & -\frac{k_2}{k_{-1}} \\
o & 1 & 0
\end{pmatrix} \times \begin{pmatrix}
k_{-1}C \\
k_3D
\end{pmatrix}
\] (2.19)

Note that to this point, no approximations have been made. Equation (2.19) represents precisely the same dynamics as the original equation (2.6), rearranged into a form suitable for the next step, where we begin to reduce the model, introducing approximations in the process.

**Step 4.** With \( r_t^*(x) \) now guaranteed to consist of linearly independent terms, we proceed to set \( r_t^*(x) = 0 \) to obtain the equilibrium manifold. The reduced differential equations will become [69]:

\[
\frac{dx}{dt} = f(x) - v_t^*(x) \cdot (L_{vt}r_t^*(x))^{-1}(L_tr_t^*(x))
\] (2.20)

where

\[
(L_tr_t^*(x))_i = \sum_{j=1}^{s} \left( \frac{\partial(r_t^*(x))_i}{\partial x_j} \right) \cdot (f(x))_j
\] (2.21)

for \( i = 1...p^* \), and
for \( i, j = 1 \ldots p^* \).

The matrix \( \mathbf{L}_{\mathbf{v}_f \mathbf{r}^*_f(x)} \) is assumed not to be singular, so that its inverse can be taken. In certain cases (not shown here), this matrix can become singular, in which case the method breaks down and the reduction will not be able to be completed. There do not appear to be any general results specifying the conditions under which \( \mathbf{L}_{\mathbf{v}_f \mathbf{r}^*_f(x)} \) is guaranteed to be non-singular, and we do not currently have any such results to present. This potential singularity represents a possible limitation in the utility of the algorithm. In practice, it is often possible to resolve the singularity issue by selecting a slightly different set of fast reactions and reapplying the steps.

**Example (continued):** Solving \( \mathbf{r}^*_f(x) = \mathbf{0} \), we find that \( C = 0 \) and \( D = 0 \). Calculating \( \mathbf{L}_{\mathbf{r}_f \mathbf{r}^*_f(x)} \) and \( \mathbf{L}_{\mathbf{v}_f \mathbf{r}^*_f(x)} \) according to equations (2.21-2.22), we find:

\[
\mathbf{L}_{\mathbf{r}_f \mathbf{r}^*_f(x)} = \begin{pmatrix} k_1k_{-1}AB \\ 0 \end{pmatrix} \quad (2.23)
\]

\[
\mathbf{L}_{\mathbf{v}_f \mathbf{r}^*_f(x)} = \begin{pmatrix} -(k_{-1} + k_2) & 0 \\ k_3 & -k_3 \end{pmatrix} \quad (2.24)
\]

\[
(\mathbf{L}_{\mathbf{v}_f \mathbf{r}^*_f(x)})^{-1} = \begin{pmatrix} 0 & 0 \\ -k_2 & -1 \end{pmatrix} \quad (2.25)
\]

\[
(\mathbf{L}_{\mathbf{v}_f \mathbf{r}^*_f(x)})^{-1}(\mathbf{L}_{\mathbf{r}_f \mathbf{r}^*_f(x)}) = \begin{pmatrix} \frac{k_1k_{-1}}{k_{-1} + k_2}AB \\ \frac{k_1k_2}{k_{-1} + k_2}AB \end{pmatrix} \quad (2.26)
\]

The reduced differential equations then become:

\[
\begin{pmatrix}
\dot{A} \\
\dot{B} \\
\dot{C} \\
\dot{D} \\
\dot{E}
\end{pmatrix} = \begin{pmatrix}
-k_1AB \\
-k_1AB \\
k_1AB \\
0 \\
- k_4E
\end{pmatrix} + \begin{pmatrix}
1 & 1 \\
1 & 1 \\
-1 - \frac{k_2}{k_{-1}} & 0 \\
\frac{k_2}{k_{-1}} & -1 \\
0 & 1
\end{pmatrix} \times \begin{pmatrix}
\frac{k_1k_{-1}}{k_{-1} + k_2}AB \\
\frac{k_1k_2}{k_{-1} + k_2}AB
\end{pmatrix} \quad (2.27)
\]

Resolving the matrix multiplication and simplifying, this becomes

\[
\begin{pmatrix}
\dot{A} \\
\dot{B} \\
\dot{C} \\
\dot{D} \\
\dot{E}
\end{pmatrix} = \begin{pmatrix}
0 \\
0 \\
0 \\
0 \\
\frac{k_1k_2}{k_{-1} + k_2}AB - k_4E
\end{pmatrix} \quad (2.28)
\]
Substituting the values of the rate constants, the single nonzero rate becomes

\[ \dot{E} = 2 \times 10^{-6} AB - 5 \times 10^{-4} E \]  

(2.29)

[Step 5.] We construct simplified reaction systems yielding the dynamics of the reduced set of differential equations. This process is difficult to formalize into an algorithm, but is generally straightforward by inspection of the equations, where first and second-order terms indicate the need for first- and second-order reactions.

Example (continued): From the single differential equation (2.29), we note that the product \( AB \) implies a second-order reaction, while the zero rates of change for species A and B indicates that these species must appear on both sides of this reaction. The negative term depending only on \( E \) implies a first-order decay of species \( E \), so the reduced set of chemical reactions is:

\[
\begin{align*}
A + B & \xrightarrow{k'_1} A + B + E \\
E & \xrightarrow{k'_2} \emptyset
\end{align*}
\]

(2.30)  

(2.31)

where \( k'_1 = 2 \times 10^{-6} \) and \( k'_2 = 5 \times 10^{-4} \).

Figure 2.1: Comparisons of the ODE and stochastic solutions for species \( E \) from the simple gene transcription system, reactions (2.2) to (2.5), used as a running example in the text. Curves are shown for the original model, equation (2.6) (solid line) and the reduced model, equation (2.29) (dashed line). (a) Time series for the deterministic versions of the original and reduced models. (b) Frequency histogram for species \( E \), for the original and reduced models. The shape of the histogram for the reduced model is faithfully reproduced, with a small shift (about 0.5%) in the mean, arising from the elimination of intermediate steps which, in the original model, slow down the transcription process and reduce the steady-state level of \( E \).
[Step 6.] In the reduced model, the fast dynamics are set to equilibrium instantaneously, whereas in fact equilibrium is expected to be reached quickly but not instantly. This approximation can lead to deviation between the reduced model and the original, full model. The degree of this deviation will depend on a number of factors, including the number of reactions treated as fast, and the details of relative speeds of the slow and fast time scales. The deviation in the reduced model can be significant when the reduction is substantial (that is, when many reactions are treated as fast), or when the difference between fast and slow rates is not large. To obtain a better fit between the original model and the reduced version, the rate constants in the reduced system can be adjusted, using the match between the models as a goodness-of-fit parameter in some form of optimization algorithm. For simple cases this may be possible using standard linear or nonlinear least-squares fitting, but in general there will be multiple rate constants to be adjusted, affecting multiple outputs. We compare the trajectories generated by the differential equations of the original deterministic model to the reduced deterministic model (neglecting stochastic effects in each case), and use simulated annealing [70] to adjust the rate constants. The adjusted rate constants usually represent the collective effect of a number of reactions in the original model, which have been lumped together into one reaction in the reduced model.

Example (continued): As shown in Figure 2.1a, in this case the reduced model gives a good match to the original model, without any fitting process. The more complex gene expression system considered in the Applications section, below, required fitting to obtain a similarly good match. In this example, the fitted rate constant for Reaction 2.30 in the reduced model represent the collective effect of Reaction 2.2 to 2.4 in the original model.

[Step 7.] The reduced reaction system obtained in Step 5 is simulated using a stochastic chemical kinetics algorithm. A number of methods exist for implementing such simulations [71]. Here, we use the Biochemical Network Stochastic simulator (BioNetS). Calculations required in the simplification process were carried out using Mathematica (Wolfram Research). Our reaction models are read directly into Mathematica from BioNetS files, simplifying the process of applying the method to a variety of biochemical systems.

Example (concluded): The simplified chemical reactions (2.30) and (2.31) were simulated in BioNetS, and the results compared to a stochastic simulation of the original reactions (2.2) to (2.5). Figure 2.1b compares the steady-state frequency histogram for species E for the two models, and shows that the reduced model reproduces the distribution of the original, with a slight shift in the mean introduced by the reduction process. In the reduction, the intermediate steps of forming species C and D have been eliminated completely, but in fact these reactions, while fast, do take some finite time to occur, slowing down the process of transcription slightly in the original model. This approximation leads to the roughly 0.5% increase in the mean seen
in the reduced model.

2.3 Applications

2.3.1 Signal cascade model

The signalling cascade of interest in this study begins with epidermal growth factor (EGF) and its corresponding epidermal growth factor receptor (EGFR); a schematic of the cascade is shown in Figure 2.2. The EGFR belongs to a family of receptors possessing an intrinsic tyrosine kinase activity, called receptor tyrosine kinases (RTKs) [72]. Receptor tyrosine kinases are characterized by containing an extracellular ligand (hormone) binding domain, a single transmembrane helix and a cytoplasmic protein tyrosine kinase domain [72]. The EGFR exists as a monomer in the plasma membrane, and the binding of EGF to the extracellular ligand binding domain of the EGFR induces a conformational change that allows two EGF/EGFR complexes to dimerize [73]. Dimerization then activates the cytoplasmic tyrosine kinase domain and the EGFR autophosphorylates itself at tyrosine residues in a trans fashion, whereby each subunit of the dimer phosphorylates the other [74]. These phosphorylated tyrosines now have the ability to serve as docking sites for downstream modular proteins with the appropriate SH2 or PTB domains since phosphotyrosine residues on the EGFR exist on the EGFR in both contexts [72]. The SchA protein binds to the phosphotyrosine residues on the EGFR through its PTB domain and it consequently becomes phosphorylated at three tyrosine residues [75]. These new phosphotyrosine residues, as well as specific phosphotyrosine residues on the EGFR, now serve as docking sites for the adaptor protein Grb2 [75]. Grb2 contains a central SH2 domain that is flanked by N-terminal and C-terminal SH3 domains, which serve to bind specific proline-rich regions [76]. As a result, via its SH2 domain Grb2 binds to phosphotyrosine residues and recruits SOS to the complex via its SH3 domains [72]. This brings SOS in close proximity to the membrane-bound Ras protein [72], a GTP-hydrolase that is inactive in its GDP-binding state and active in its GTP-binding state [77]. Once in its active GTP bound state, Ras interacts with Raf and consequently activates it through serine phosphorylation [72]. Activated Raf then catalyzes the phosphorylation of MEK at two serine residues [72]. The activated MEK then in turn activates ERK by catalyzing the phosphorylation of a threonine and a tyrosine residue [72]. Finally, activated ERK is readily translocated into the nucleus, where it phosphorylates and activates several transcription factors [72]. The modelling of this signal cascade will end at activated ERK and does not include the subsequent translocation to the nucleus followed by transcription factor activation. The complete list of names of species in the model is shown in Table 2.3.1. The chemical reactions involved are shown in Figure 2.3, as entered into the BioNetS user interface. The values of the rate constants are available in the BioNetS files, but
for completeness we list them below:

**Rate constants for the signal cascade model.** The same rate constants are used in both the full and reduced versions of the signalling cascade model, and their values are (all in units of $min^{-1}$): $k_1 = 2.1 \times 10^{-5}$, $k_2 = 7.8 \times 10^{-5}$, $k_3 = 163$, $k_4 = k_{20} = 2.1 \times 10^{-4}$, $k_5 = 446$, $k_6 = 8$, $k_7 = k_{24} = 7.1 \times 10^{-4}$, $k_8 = 1200$, $k_9 = k_{10} = k_{16} = k_{22} = 1.1 \times 10^{-5}$, $k_{11} = 0.066$, $k_{12} = 42$, $k_{13} = 3.6 \times 10^{-5}$, $k_{14} = 9$, $k_{15} = 0.012$, $k_{17} = 7.1 \times 10^{-5}$, $k_{18} = 0.17$, $k_{19} = 7.1 \times 10^{-6}$, $k_{21} = 9$, $k_{22} = 0.006$, $k_{23} = 291$, $k_{25} = 10^{-4}$, $k_{26} = k_{38} = k_{24} = 30$, $k_{27} = 7 \times 10^{-6}$, $k_{28} = 60$, $k_{29} = k_{31} = 8 \times 10^{-5}$, $k_{30} = 210$, $k_{32} = 174$, $k_{33} = k_{35} = 8 \times 10^{-7}$, $k_{34} = 960$, $k_{36} = 342$, $k_{37} = k_{18} = 0.011$, $k_{1} = k_{17} = k_{20} = 3.6$, $k_{2} = 49$, $k_{4} = 355$, $k_{7} = 60$, $k_{9} = 3$, $k_{10} = 0.18$, $k_{11} = 0.0036$, $k_{12} = 3.6 \times 10^{-6}$, $k_{13} = 5.7$, $k_{16} = 12$, $k_{19} = 6$, $k_{21} = 2.5 \times 10^{-4}$, $k_{23} = 4.6 \times 10^{-6}$, $k_{25} = k_{37} = 78$, $k_{27} = 0.32$, and $k_{29} = k_{31} = 1.1$. Note that the rate constants are obtained from publications [72–75]. Although the rate constants might subject to significant (but unknown) errors, the exact values aren’t important here, as this model is used here as a biologically plausible model to test our method.
Chapter 2. Simplification of stochastic chemical reaction models

<table>
<thead>
<tr>
<th>Name of Species</th>
<th>Description of Species</th>
<th>Initial Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>IR</td>
<td>inactive EGFR (epidermal growth factor receptor)</td>
<td>15000</td>
</tr>
<tr>
<td>L</td>
<td>ligand which in this case is EGF (epidermal growth factor)</td>
<td>100000</td>
</tr>
<tr>
<td>iRL</td>
<td>= epidermal growth factor bound to epidermal growth factor receptor</td>
<td>0</td>
</tr>
<tr>
<td>iRLRL</td>
<td>two epidermal growth factor receptors, bound together forming a dimer</td>
<td>0</td>
</tr>
<tr>
<td>RP</td>
<td>phosphorylated (now active) epidermal growth factor dimer</td>
<td>0</td>
</tr>
<tr>
<td>PT1</td>
<td>Tyrosine phosphotases 1</td>
<td>15000</td>
</tr>
<tr>
<td>PT2</td>
<td>Tyrosine phosphotases 2</td>
<td>30000</td>
</tr>
<tr>
<td>SH</td>
<td>Src homology and collagen domain protein A (ShcA)</td>
<td>0</td>
</tr>
<tr>
<td>SHP</td>
<td>Src homology and collagen domain protein A phosphorylated</td>
<td>30000</td>
</tr>
<tr>
<td>GR</td>
<td>growth factor receptor-binding protein 2 (Grb2)</td>
<td>17000</td>
</tr>
<tr>
<td>SO</td>
<td>Son of Sevenless homolog protein</td>
<td>20000</td>
</tr>
<tr>
<td>RD</td>
<td>Ras in its GDP (guanosine diphosphate) bound (inactive) state</td>
<td>20000</td>
</tr>
<tr>
<td>RasGTP</td>
<td>Ras in its GTP (guanosine triphosphate) bound (active) state</td>
<td>0</td>
</tr>
<tr>
<td>iRaf</td>
<td>Inactive inactive Raf</td>
<td>0</td>
</tr>
<tr>
<td>aRaf</td>
<td>Active active Raf</td>
<td>0</td>
</tr>
<tr>
<td>MEK</td>
<td>mitogen-activated protein kinase</td>
<td>360000</td>
</tr>
<tr>
<td>ERK</td>
<td>extracellular signal-regulated kinase</td>
<td>750000</td>
</tr>
</tbody>
</table>

All complexes not shown in this table have a initial value of zero.

The following standardized notation is used for all species:
- _ refers to a bound complex (A_B implies a complex where A is bound to B); and
- P - represents a phosphorylated tyrosine residue (EP and EPP represent species E phosphorylated at one and two residues, respectively).

Table 2.1: Brief Description of Species in Signal Cascade Model

The original model has 63 reactions and 41 species. To begin the reduction of the original model, we identify 19 reactions as fast, selecting based on the magnitude of their rate constants. Separating the fast and slow reactions, we will get \( \mathbf{v}_f(x) \) with 41 rows and 19 columns. The column rank of this matrix is 18, less than the number of columns 19. So independent items are picked out of the \( \mathbf{v}_f(x) \) to form \( \mathbf{v}'_f(x) \), and \( \mathbf{r}'_f(x) \) is calculated correspondingly. After this we calculate the Jacobian matrix of \( \mathbf{v}'_f(x) \), and the rank of this Jacobian matrix is 14, less than the number of columns 18, thus we need construct a nonsingular matrix \( \mathbf{E}(x) \) to make \( r''_f(x) = \mathbf{E}(x) \cdot r'_f(x) \) have the first 14 rows with independent scalar functions and the last 4 rows of \( r''_f(x) \) be zero. To construct \( \mathbf{E}(x) \), we need find the 14 independent rows from \( r'_f(x) \) first. then we can construct the first 14 rows of \( \mathbf{E}(x) \) to pick out the 14 independent scalar functions from \( r'_f(x) \). To zero out the last 4 rows of \( r''(x) \), the 4 dependent scalar functions are decomposed in terms of the 14 independent scalar functions in \( r'_f(x) \). Then, by putting -1 in the last four rows of \( \mathbf{E}(x) \) to pick out the dependent rows and put the corresponding decomposition coefficients inside, we can make the last 4 rows of \( \mathbf{E}(x) \) to be zero. From the \( \mathbf{E}(x) \) we construct, we get the final version of \( \mathbf{v}'_f(x) \) and \( r'_f(x) \). Setting \( r''_f(x) \) to zero and following the calculations in step 4, we get the final simplified differential equations for this system, from which we deduce the reduced reaction system as shown in Figure 2.4. For this system, no fitting of the rate
Figure 2.2: Schematic of the ShcA signalling cascade. Binding of epidermal growth factor (EGF) to a membrane-bound receptor triggers a chain of activations, starting with phosphorylation of the SchA protein and leading to extracellular signal-regulated kinase (ERK) and activated ERK (ERKPP). The actual cascade continues past this point, with nuclear translocations leading to transcription factor activation, but the model halts at ERKPP.
<table>
<thead>
<tr>
<th>Left</th>
<th>Right</th>
<th>Forward</th>
<th>Backward</th>
</tr>
</thead>
<tbody>
<tr>
<td>iR + L</td>
<td>iRL</td>
<td>k1</td>
<td>k_1</td>
</tr>
<tr>
<td>iRL + iRL</td>
<td>iRLRL</td>
<td>k2</td>
<td>k_2</td>
</tr>
<tr>
<td>iRLRL</td>
<td>RP</td>
<td>k3</td>
<td></td>
</tr>
<tr>
<td>RP + PT1</td>
<td>RP_PT1</td>
<td>k4</td>
<td>k_4</td>
</tr>
<tr>
<td>RP_PT1</td>
<td>iRLRL_PT1</td>
<td>k5</td>
<td></td>
</tr>
<tr>
<td>iRLRL_PT1</td>
<td>iRLRL + PT1</td>
<td>k6</td>
<td></td>
</tr>
<tr>
<td>RP + SH</td>
<td>RP_SH</td>
<td>k7</td>
<td>k_7</td>
</tr>
<tr>
<td>RP_SH</td>
<td>RP_SHP</td>
<td>k8</td>
<td></td>
</tr>
<tr>
<td>RP_SHP + GR</td>
<td>RP_SHP_GR</td>
<td>k9</td>
<td>k_9</td>
</tr>
<tr>
<td>RP_SHP_GR + SO</td>
<td>RP_SHP_GR_SO</td>
<td>k10</td>
<td>k_10</td>
</tr>
<tr>
<td>RP_SHP_GR_SO</td>
<td>RP + SHP_GR_SO</td>
<td>k11</td>
<td>k_11</td>
</tr>
<tr>
<td>RP_SHP</td>
<td>RP + SHP</td>
<td>k12</td>
<td>k_12</td>
</tr>
<tr>
<td>SHP + PT2</td>
<td>SHP_PT2</td>
<td>k13</td>
<td>k_13</td>
</tr>
<tr>
<td>SHP_PT2</td>
<td>SH + PT2</td>
<td>k14</td>
<td></td>
</tr>
<tr>
<td>SH_PT2</td>
<td>GR + SO</td>
<td>k15</td>
<td></td>
</tr>
<tr>
<td>RP + GR</td>
<td>RP_GR</td>
<td>k16</td>
<td>k_16</td>
</tr>
<tr>
<td>RP_GR + SO</td>
<td>RP_GR_SO</td>
<td>k17</td>
<td>k_17</td>
</tr>
<tr>
<td>RP_GR_SO</td>
<td>RP + GR_SO</td>
<td>k18</td>
<td>k_18</td>
</tr>
<tr>
<td>SHP + GR</td>
<td>SHP_GR</td>
<td>k19</td>
<td>k_19</td>
</tr>
<tr>
<td>SHPGR + SO</td>
<td>SHP_GR_SO</td>
<td>k20</td>
<td>k_20</td>
</tr>
<tr>
<td>SHP_GR_SO</td>
<td>SHP + GR_SO</td>
<td>k21</td>
<td>k_21</td>
</tr>
<tr>
<td>GR_SO</td>
<td>GR + SO</td>
<td>k22</td>
<td>k_22</td>
</tr>
<tr>
<td>RP_SHP_GR</td>
<td>RP + SHP_GR</td>
<td>k23</td>
<td>k_23</td>
</tr>
<tr>
<td>RP_SHP + GR_SO</td>
<td>RP_SHP_GR_SO</td>
<td>k24</td>
<td>k_24</td>
</tr>
<tr>
<td>RP_SHP_GR_SO + RD</td>
<td>RP_SHP_GR_SO_RD</td>
<td>k25</td>
<td>k_25</td>
</tr>
<tr>
<td>RP_SHP_GR_SO_RD</td>
<td>RP_SHP_GR_SO + RasGTP</td>
<td>k26</td>
<td></td>
</tr>
<tr>
<td>RasGTP + iRaf</td>
<td>Raf_RasGTP</td>
<td>k27</td>
<td>k_27</td>
</tr>
<tr>
<td>Raf_RasGTP</td>
<td>aRaf + RasGTP</td>
<td>k28</td>
<td></td>
</tr>
<tr>
<td>aRaf + MEK</td>
<td>aRaf_MEK</td>
<td>k29</td>
<td>k_29</td>
</tr>
<tr>
<td>aRaf_MEK</td>
<td>MEK + aRaf</td>
<td>k30</td>
<td></td>
</tr>
<tr>
<td>aRaf + MEKP</td>
<td>aRaf_MEKP</td>
<td>k31</td>
<td>k_31</td>
</tr>
<tr>
<td>aRaf_MEKP</td>
<td>MEKPP + aRaf</td>
<td>k32</td>
<td></td>
</tr>
<tr>
<td>MEKPP + ERK</td>
<td>MEKPP_ERK</td>
<td>k33</td>
<td>k_33</td>
</tr>
<tr>
<td>MEKPP_ERK</td>
<td>ERK + MEKPP</td>
<td>k34</td>
<td></td>
</tr>
<tr>
<td>MEKPP + ERKP</td>
<td>MEKPP_ERKP</td>
<td>k35</td>
<td>k_35</td>
</tr>
<tr>
<td>MEKPP_ERKP</td>
<td>ERK + MEKPP</td>
<td>k36</td>
<td></td>
</tr>
<tr>
<td>RP_GR_SO + RD</td>
<td>RP_GR_SO_RD</td>
<td>k37</td>
<td>k_37</td>
</tr>
<tr>
<td>RP_GR_SO_RD</td>
<td>RP_GR_SO + RasGTP</td>
<td>k38</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.3: Full signal cascade model, as entered into the BioNetS user interface [51]. Each line represents one reaction. Species shown on the "Left" column appear on the left side of a chemical reaction, while species shown on the "Right" column appear on the right. The "Forward" and "Backward" columns give the rate constants for the forward and reverse reactions.
The reduced model contains 27 species and 34 reactions with an overall reduction in computational time of 50% as compared to the original model. For an accurate comparison of the the fluctuations in the original and reduced models, both models are run 10000 times with varying random number generator seeds, with a simulated extracellular signal applied to the network at time zero. This yields 10000 realizations of the random process, and statistics (mean, standard deviation, skewness, and kurtosis) are calculated over this ensemble at each point in the time series, as shown in Figure 2.5. All random sample paths converge to a consistent final value of ERKPP after an initial transient, and the standard deviation drops to near zero after this transient. The distribution of ERKPP values over the ensemble converges to a Gaussian distri-
Figure 2.5: Comparisons of statistical parameters as a function of time for the ERKPP species, from the original (solid line) and reduced (dashed line) models. In each case, statistics were accumulated over 10000 runs. (a) Mean number of ERKPP. (b) Standard deviation. (c) Skewness. (d) Kurtosis.
Figure 2.6: Comparisons of statistical parameters as a function of time for the ERK species, from the original (solid line) and reduced (dashed line) models. In each case, statistics were accumulated over 10000 runs. (a) Mean number of ERK. (b) Standard deviation. (c) Skewness. (d) Kurtosis.
Figure 2.7: Comparisons of statistical parameters as a function of time for the ERKP species, from the original (solid line) and reduced (dashed line) models. In each case, statistics were accumulated over 10000 runs. (a) Mean number of ERKP. (b) Standard deviation. (c) Skewness. (d) Kurtosis.
Table 2.2: Brief Description of species in Gene Expression Model

<table>
<thead>
<tr>
<th>Name of Species</th>
<th>Description of Species</th>
<th>Initial Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rpoly</td>
<td>RNA polymerase</td>
<td>7600</td>
</tr>
<tr>
<td>Rpoly_operon_ns</td>
<td>RNA polymerase elongating an average mRNA transcript from a template operon</td>
<td>0</td>
</tr>
<tr>
<td>closed Rpoly_prom_ns</td>
<td>RNA polymerase in a closed-complex with an mRNA operon promoter</td>
<td>0</td>
</tr>
<tr>
<td>closed Rpoly_prom_s</td>
<td>RNA polymerase in a closed-complex with an mRNA operon promoter</td>
<td>0</td>
</tr>
<tr>
<td>mRNA</td>
<td>Average mRNA (gene length)</td>
<td>0</td>
</tr>
<tr>
<td>mRNA_small</td>
<td>Approximately 90% of the average mRNA</td>
<td>0</td>
</tr>
<tr>
<td>open Rpoly_prom_ns</td>
<td>RNA polymerase in an open-complex with an mRNA operon promoter</td>
<td>0</td>
</tr>
<tr>
<td>open Rpoly_prom_s</td>
<td>RNA polymerase in an open-complex with a stable RNA operon promoter</td>
<td>0</td>
</tr>
<tr>
<td>operon_ns</td>
<td>Average mRNA operon</td>
<td>1541</td>
</tr>
<tr>
<td>operon_s</td>
<td>Average stable RNA (full operon length)</td>
<td>104</td>
</tr>
<tr>
<td>stable_RNA</td>
<td>Average stable RNA</td>
<td>0</td>
</tr>
<tr>
<td>incom_mRNA</td>
<td>Nascent (incomplete) mRNA</td>
<td>0</td>
</tr>
<tr>
<td>incom_mRNA_small</td>
<td>Nascent small mRNA</td>
<td>0</td>
</tr>
<tr>
<td>open_ribo_incom_RBS</td>
<td>A ribosome bound to the RBS of a nascent mRNA</td>
<td>0</td>
</tr>
<tr>
<td>open_ribo_incom_RBS_small</td>
<td>A ribosome bound to the RBS of a small nascent mRNA</td>
<td>0</td>
</tr>
<tr>
<td>open_ribo_RBS</td>
<td>A ribosome bound to the RBS of an average mRNA</td>
<td>0</td>
</tr>
<tr>
<td>open_ribo_RBS_small</td>
<td>A ribosome bound to the RBS of a small mRNA</td>
<td>0</td>
</tr>
<tr>
<td>protein</td>
<td>Average polypeptide</td>
<td>0</td>
</tr>
<tr>
<td>protein_small</td>
<td>Polypeptide that is approximately 90% the size of an average polypeptide</td>
<td>0</td>
</tr>
<tr>
<td>ribo</td>
<td>Ribosome</td>
<td>48000</td>
</tr>
<tr>
<td>ribo_incom_mRNA</td>
<td>Ribosome currently elongating on a nascent mRNA</td>
<td>0</td>
</tr>
<tr>
<td>ribo_incom_mRNA_small</td>
<td>Ribosome currently elongating on a small nascent mRNA</td>
<td>0</td>
</tr>
<tr>
<td>ribo_mRNA</td>
<td>Ribosome currently elongating on an average mRNA</td>
<td>0</td>
</tr>
<tr>
<td>ribo_mRNA_small</td>
<td>Ribosome currently elongating on a small mRNA</td>
<td>0</td>
</tr>
<tr>
<td>v</td>
<td>Volume of the system</td>
<td>1000</td>
</tr>
</tbody>
</table>

bution for large time values, with a skewness of zero and a kurtosis of three. The reduced model gives an accurate representation for all statistical parameters in the species of interest. The comparison of ERK and ERKP is also shown in Figure 2.6 and Figure 2.7. Some species that have been removed in the model reduction existed only in numbers fewer than 10; interestingly, the statistical information present in the original model is satisfactorily preserved despite these excisions.

2.3.2 Detailed gene expression model

We have also applied the methodology in this paper to the process of gene expression (transcription and translation) in a dividing *E. coli* bacterial cell. Please see Iafolla and McMillen [68] for a more detailed discussion of the model; here we present an overview of the processes included. The whole model includes RNA polymerases (RNAPs) transcribing both stable RNA (sRNA) and messenger RNA (mRNA) from a dividing genome. Closed and open complexes between the promoter sequences and RNA polymerases are required prior to elongation events. The set of reactions is designed to allow multiple RNAPs binding to the same gene, mimicking events in a real cell. The nascent and complete mRNA are subject to translation by ribosomes. The model ensures that nascent transcripts are converted into their complete species, accounting for those species that are complexed with a ribosome. Analogous to RNA polymerases, the ribosomes form a complex with the ribosome-binding-site (RBS) prior to elongation and also offer the
Figure 2.8: (a) **Transcription**: 1. RNA polymerase (Rpoly) has the potential to bind and unbind to the promoter. 2. Upon binding to the promoter, Rpoly opens the DNA strands in a process called isomerization. 3. After opening the RNA, Rpoly clears the promoter site and begins transcribing the mRNA. Clearing the promoter site allows additional Rpoly to transcribe the same gene simultaneously (not shown). 4. After completing transcription, the Rpoly is ejected from the gene and releases the completed transcript.  

(b) **Translation**: 1. Ribosomes (ribo) both bind and unbind to the ribosome binding site (RBS) on a complete mRNA or nascent mRNA. 2. Upon binding to the RBS, the ribo clears the RBS and begins translation. Clearing the RBS allows additional ribos to translate the same transcript simultaneously (not shown). 4. After completing translation, the ribo is ejected from the transcript and releases the completed peptide.
possibility for multiple elongations from more than one ribosome bound to a single mRNA. A schematic view of the main processes is presented in Figures 2.8a and 2.8b, and the complete set of reactions is shown in Figure 2.9.

The original model has 71 reactions and 47 species, with a range of rate constants that makes the division into fast and slow reactions somewhat flexible. By treating different numbers of reactions as fast (29 and 59), we reduce the original model to different degree. For each case (29 fast reactions or 59 fast reactions), separating the fast and slow reactions, we will get $\mathbf{v}_f(x)$ with 29 columns and 59 columns respectively. The column rank of these matrices are 28 and 41 respectively, less than the number of columns (29 and 59). So independent terms must be picked out of the the $\mathbf{v}_f(x)$ to form $\mathbf{v}'_f(x)$, and $\mathbf{r}'_f(x)$ is calculated as described in the Methods section. After this we calculate the Jacobian matrices of $\mathbf{v}'_f(x)$, and the ranks of these matrices are 18 and 37, less than the number of columns (28 and 41). Thus we follow step 3 and use the same methods as used in example in the methods part to construct $\mathbf{E}(x)$, from which we get the final version of $\mathbf{v}'_f(x)$ and $\mathbf{r}'_f(x)$. Setting $\mathbf{r}'_f(x)$ to zero and following calculations on step 4, we get the final simplified differential equations for this system, which are then converted to the reduced reaction systems displayed in Figure 2.12 and Figure 2.13. The rate constants for both the full gene expression model and the reduced gene expression models are listed below:

**Rate constants for the full gene expression model.** Complete details of the model shown in Figure 2.9 have been reported elsewhere [68]. The rate constants are: $k_{\text{RBS-clearance}} = .5$, $k_{\text{cell-div}} = 4.8 \times 10^{-4}$, $k_{\text{isomerization}} = 1$, $k_{\text{off-Rpoly}} = 10$, $k_{\text{off-ribo}} = 2.25$, $k_{\text{on-Rpoly-prom ns}} = 3.9 \times 10^{-6}$, $k_{\text{on-Rpoly-prom s}} = 5.9 \times 10^{-4}$, $k_{\text{on-ribo-RBS}} = 5.4 \times 10^{-5}$, $k_{\text{prom-clearance}} = 1$, $k_{\text{rep-Rpoly}} = 5.2$, $k_{\text{rep-operon ns}} = 1.1$, $k_{\text{rep-operon s}} = .07$, $k_{\text{rep-ribo}} = 33.3$, $k_{\text{ribo-conversion}} = .05$, $k_{\text{transcription ns}} = .05$, $k_{\text{transcription s}} = .09$, and $k_{\text{translation}} = .06$. All zeroth and first order rate constants are given in units of $\text{sec}^{-1}$, while second order rate constants are given in $\mu\text{m}^3 \times \text{sec}^{-1}$.

**Rate constants for first reduction of the gene expression model.** The first reduction of the gene expression model (to 28 reactions and 29 species, as shown in Figure 2.12) used simulated annealing to adjust the rate constants to obtain a better fit to the original model. The fitted rate constants (with the values from the original model noted when a direct correspondence exists) are: $k_{\text{cell-div}} = 4.8 \times 10^{-4}$ (unchanged), $k_{\text{rep-Rpoly}} = 5.2778$ (vs. 5.2), $k_{\text{rep-operon ns}} = 1.0978$ (vs. 1.1), $k_{\text{rep-operon s}} = .0721$ (vs. 0.07), $k_{\text{rep-ribo}} = 33.3$ (unchanged), $k_{\text{ribo-on}} = .0095$, $k_{\text{transcription initial}} = 3 \times 10^{-4}$, $k_{\text{transcription ns}} = .071$ (vs. .05), $k_{\text{transcription stable}} = .0285$ (vs. .09), $k_{\text{translation}} = .051$ (vs. .06), and $k_{\text{translation small}} = .0094$. As above, all zeroth and first order rate constants are given in units of $\text{sec}^{-1}$, while second order rate constants are given in $\mu\text{m}^3 \times \text{sec}^{-1}$.

**Rate constants for second reduction of the gene expression model.** The second
Table 2.1: Reactions of the full gene expression model.

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Species/Compounds</th>
<th>Rate Constants</th>
</tr>
</thead>
<tbody>
<tr>
<td>operon,ns + Rpoly</td>
<td>closed_Rpoly_prom,ns</td>
<td>k_{on,Rpoly_prom,ns}/(v1000) k_{off,Rpoly}</td>
</tr>
<tr>
<td>closed_Rpoly_prom,ns</td>
<td>open_Rpoly_prom,ns</td>
<td>k_{isomerization}</td>
</tr>
<tr>
<td>open,Rpoly_prom,ns</td>
<td>operon,ns + Rpoly,operon,ns + incom_mRNA,1</td>
<td>k_{prom,clearance}</td>
</tr>
<tr>
<td>Rpoly,operon,ns,1</td>
<td>Rpoly,operon,ns,2 + mRNA + incom_mRNA,2</td>
<td>k_{transcription}</td>
</tr>
<tr>
<td>Rpoly,operon,ns,2</td>
<td>Rpoly,operon,ns,3 + mRNA + incom_mRNA,3</td>
<td>k_{transcription}</td>
</tr>
<tr>
<td>Rpoly,operon,ns,3</td>
<td>Rpoly,operon,ns,4 + mRNA + incom_mRNA,4</td>
<td>k_{transcription}</td>
</tr>
<tr>
<td>Rpoly,operon,ns,4</td>
<td>Rpoly,operon,ns,5 + mRNA + incom_mRNA,5</td>
<td>k_{transcription}</td>
</tr>
<tr>
<td>Rpoly,operon,ns,5</td>
<td>Rpoly,operon,ns,6 + mRNA + incom_mRNA,6</td>
<td>k_{transcription}</td>
</tr>
<tr>
<td>Rpoly,operon,ns,6</td>
<td>Rpoly,operon,ns,7 + mRNA + incom_mRNA_small</td>
<td>k_{transcription}</td>
</tr>
<tr>
<td>Rpoly,operon,ns,7</td>
<td>Rpoly + mRNA_small</td>
<td>k_{transcription}</td>
</tr>
<tr>
<td>operon,s + Rpoly</td>
<td>closed_Rpoly_prom,s</td>
<td>k_{on,Rpoly_prom,s}/(v1000) k_{off,Rpoly}</td>
</tr>
<tr>
<td>closed_Rpoly_prom,s</td>
<td>open,Rpoly_prom,s</td>
<td>k_{isomerization}</td>
</tr>
<tr>
<td>open,Rpoly_prom,s</td>
<td>operon,s + Rpoly,operon,s</td>
<td>k_{prom,clearance}</td>
</tr>
<tr>
<td>operon,s + Rpoly</td>
<td>Rpoly + stable RNA</td>
<td>k_{cell,div}</td>
</tr>
<tr>
<td>operon,ns</td>
<td>operon,s + Rpoly, operon,ns</td>
<td>k_{rep, operon,ns}</td>
</tr>
<tr>
<td>Rpoly,operon,ns</td>
<td>operon,ns</td>
<td>k_{rep, Rpoly}</td>
</tr>
<tr>
<td>ribo</td>
<td>incommRNA,1</td>
<td>k_{transcription}</td>
</tr>
<tr>
<td>incommRNA,2</td>
<td>k_{transcription}</td>
<td></td>
</tr>
<tr>
<td>incommRNA,3</td>
<td>k_{transcription}</td>
<td></td>
</tr>
<tr>
<td>incommRNA,4</td>
<td>k_{transcription}</td>
<td></td>
</tr>
<tr>
<td>incommRNA,5</td>
<td>k_{transcription}</td>
<td></td>
</tr>
<tr>
<td>incommRNA_small</td>
<td>k_{transcription}</td>
<td></td>
</tr>
<tr>
<td>ribo + incommRNA,1</td>
<td>open, ribo, incomm_RBS,1</td>
<td>k_{on, ribo_RBS}/(v1000) k_{off, ribo}</td>
</tr>
<tr>
<td>open, ribo, incomm_RBS,1</td>
<td>ribo, incommRNA,1 + incommRNA,1</td>
<td>k_{RBS, clearance}</td>
</tr>
<tr>
<td>ribo, incommRNA,1</td>
<td>ribo + protein</td>
<td>k_{RBS, clearance}</td>
</tr>
<tr>
<td>ribo + incommRNA,2</td>
<td>open, ribo, incomm_RBS,2</td>
<td>k_{on, ribo_RBS}/(v1000) k_{off, ribo}</td>
</tr>
<tr>
<td>open, ribo, incomm_RBS,2</td>
<td>ribo, incommRNA,2 + incommRNA,2</td>
<td>k_{RBS, clearance}</td>
</tr>
<tr>
<td>ribo, incommRNA,2</td>
<td>ribo + protein</td>
<td>k_{RBS, clearance}</td>
</tr>
<tr>
<td>ribo + incommRNA,3</td>
<td>open, ribo, incomm_RBS,3</td>
<td>k_{on, ribo_RBS}/(v1000) k_{off, ribo}</td>
</tr>
<tr>
<td>open, ribo, incomm_RBS,3</td>
<td>ribo, incommRNA,3 + incommRNA,3</td>
<td>k_{RBS, clearance}</td>
</tr>
<tr>
<td>ribo, incommRNA,3</td>
<td>ribo + protein</td>
<td>k_{RBS, clearance}</td>
</tr>
<tr>
<td>ribo + incommRNA,4</td>
<td>open, ribo, incomm_RBS,4</td>
<td>k_{on, ribo_RBS}/(v1000) k_{off, ribo}</td>
</tr>
<tr>
<td>open, ribo, incomm_RBS,4</td>
<td>ribo, incommRNA,4 + incommRNA,4</td>
<td>k_{RBS, clearance}</td>
</tr>
<tr>
<td>ribo, incommRNA,4</td>
<td>ribo + protein</td>
<td>k_{RBS, clearance}</td>
</tr>
<tr>
<td>ribo + incommRNA,5</td>
<td>open, ribo, incomm_RBS,5</td>
<td>k_{on, ribo_RBS}/(v1000) k_{off, ribo}</td>
</tr>
<tr>
<td>open, ribo, incomm_RBS,5</td>
<td>ribo, incommRNA,5 + incommRNA,5</td>
<td>k_{RBS, clearance}</td>
</tr>
<tr>
<td>ribo, incommRNA,5</td>
<td>ribo + protein</td>
<td>k_{RBS, clearance}</td>
</tr>
<tr>
<td>ribo + incommRNA,6</td>
<td>open, ribo, incomm_RBS,6</td>
<td>k_{on, ribo_RBS}/(v1000) k_{off, ribo}</td>
</tr>
<tr>
<td>open, ribo, incomm_RBS,6</td>
<td>ribo, incommRNA,6 + incommRNA,6</td>
<td>k_{RBS, clearance}</td>
</tr>
<tr>
<td>ribo, incommRNA,6</td>
<td>ribo + protein</td>
<td>k_{RBS, clearance}</td>
</tr>
<tr>
<td>ribo + incommRNA_small</td>
<td>open, ribo, incomm_RBS_small</td>
<td>k_{on, ribo_RBS}/(v1000) k_{off, ribo}</td>
</tr>
<tr>
<td>open, ribo, incomm_RBS_small</td>
<td>ribo, incommRNA_small + incommRNA_small</td>
<td>k_{RBS, clearance}</td>
</tr>
<tr>
<td>ribo, incommRNA_small</td>
<td>ribo + protein</td>
<td>k_{RBS, conversion}</td>
</tr>
<tr>
<td>ribo + mRNA</td>
<td>open, ribo_RBS</td>
<td>k_{on, ribo_RBS}/(v1000) k_{off, ribo}</td>
</tr>
<tr>
<td>open, ribo_RBS</td>
<td>ribo, mRNA + mRNA</td>
<td>k_{RBS, clearance}</td>
</tr>
<tr>
<td>ribo, mRNA</td>
<td>ribo + protein</td>
<td>k_{RBS, conversion}</td>
</tr>
<tr>
<td>ribo + mRNA_small</td>
<td>open, ribo_RBS_small</td>
<td>k_{on, ribo_RBS}/(v1000) k_{off, ribo}</td>
</tr>
<tr>
<td>open, ribo_RBS_small</td>
<td>ribo, mRNA_small + mRNA_small</td>
<td>k_{RBS, clearance}</td>
</tr>
<tr>
<td>ribo, mRNA_small</td>
<td>ribo + protein</td>
<td>k_{RBS, conversion}</td>
</tr>
</tbody>
</table>

Figure 2.9: Full gene expression model (71 reactions, 47 species), shown in the BioNetS user interface, with columns labelled as in Figure 2.3.
Figure 2.10: Comparisons of protein numbers in the gene expression system, from the original model with 71 reactions and 47 species (solid line), the reduced model with 28 reactions and 29 species (dashed line), and the reduced model with 10 reactions and 10 species (dotted line). (a) Before fitting, using rate constants from the original model directly in the reduced models. (b) After fitting, where the rate constants have been adjusted using simulated annealing.

reduction of the gene expression model (to 10 reactions and 10 species, as shown in Figure 2.13) again used simulated annealing to adjust the rate constants to improve the match between the ODE solutions of the reduced and full models. The fitted rate constants (with the values from the original model noted when a direct correspondence exists) are: $k_{cell\_div} = 4.8 \times 10^{-4}$ (unchanged), $k_{rep\_Rpoly} = 5.2778$ (vs. 5.2), $k_{rep\_operon\_ns} = 1.0978$ (vs. 1.1), $k_{rep\_operon\_s} = 0.0721$ (vs. 0.07), $k_{rep\_ribo} = 33.3$ (unchanged), $k_{transcription\_ns} = 0.00167$ (vs. 0.05), $k_{transcription\_s} = 0.074$ (vs. 0.09), $k_{transcription\_small} = 1.1 \times 10^{-4}$, $k_{translation} = 0.001$ (vs. 0.06), and $k_{translation\_small} = 0.001$. As above, all zeroth and first order rate constants are given in units of $sec^{-1}$, while second order rate constants are given in $\mu m^3 \times sec^{-1}$. Note that the rate constants are obtained from publication [68]. Although the rate constants might subject to significant (but unknown) errors, the exact values aren’t important here, as this model is used here as a biologically plausible model to test our method.

As shown in Figure 2.10a, the deterministic behavior of the reduced models shows significant deviations from the original model for both versions of the reduced model, but especially for the severely reduced 10 reaction, 10 species model. As in the illustrative Example in the Methods section, the reduction process tends to yield an over-expression of final products, by removing fast but not instantaneous intermediate steps that serve to slow down the net reaction rate in the original model. To obtain a better fit for the reduced models, we follow Step 6 from the
Figure 2.11: Comparisons of statistical parameters as a function of time for protein numbers in the gene expression system. Shown are the original model with 71 reactions and 47 species (solid line), the reduced model with 28 reactions and 29 species (dashed line), and the reduced model with 10 reactions and 10 species (dotted line). (a) Mean number of proteins. (b) Coefficient of Variation. (c) Skewness. (d) Kurtosis.
Figure 2.12: First reduction of the gene expression model (28 reactions, 29 species), shown in the BioNetS user interface, with columns labelled as in Figure 2.3.

Methods section: solve the original and reduced models deterministically, then use simulated annealing to find the optimized rate constants for each reduced model. Plugging the fitted reaction rates into the reduced model, we get a good match between the original model and two reduced models as seen in Figure 2.10b.

Comparing Figures 2.10 and 2.11, we note that the average behaviour of the stochastic simulations shows a significant difference from the deterministic behaviour: rather than maintaining the “sawtooth” pattern generated by the growth and division of the simulated cells, the stochastic mean shows a damped, decaying oscillation converging toward a flat steady state. This effect arises from the stochastic nature of the cell division itself in the simulations: the timing of the cell division events is itself a stochastic variable, with division triggered when a “dummy” species representing cellular volume crosses a designated threshold. Each realization of the stochastic process will trigger cell divisions at slightly different times, and over long enough times the
Figure 2.13: Second reduction of the gene expression model (10 reactions, 10 species), again given in the form of the BioNetS interface screen, with columns labelled as in Figure 2.3.

Phase information is erased, meaning that at a given instant each cell may be at any phase of the cell division cycle, and the oscillatory behaviour will not be seen in the ensemble average.

Using BioNetS, single simulation run of the original model takes 650 s on a 2.5 GHz G5 CPU, while the reduced model with 29 species takes 80 s and the reduced model with 10 species takes 22 s, representing an improvement in computational speed of between 8-fold and 30-fold, depending on the degree of reduction. Despite the simplicity of the reduced models, they still give a good representation in terms of the statistical parameters as seen in Figures 2.11 and 2.14. Figure 2.15 shows the steady-state histogram for the mRNA and protein species, and again the match between the original and reduced models is evident, though the mRNA results match better than the proteins. The statistical parameters for protein_small, mRNA_small, operon_ns, and operon_s also yield good matches between the original and reduced models (results not shown). One limitation, however, is that the reduced model with 10 species can only capture the steady state behaviour; during the initial transient before steady state is reached, this model displays large deviations from the original model. The reduced model with 29 species, on the other hand, can capture both the steady state and the transient behaviour. Depending on the goals of a particular modelling effort, different degrees of reduction may be appropriate; some investigations may consider only steady-state values, for example, while others may wish to capture transients.

To show how this method works in reaction networks with feedback (where product species interact with earlier steps in the reaction scheme), we add a simple feedback reaction to the original gene expression model:

\[
\text{protein\_small + operon\_ns} \leftrightarrow \text{protein\_small\_operon\_ns}
\]

This represents the small proteins produced in the model binding to operator sites, occupying them and preventing polymerase binding, and thus also preventing transcription; the result is a negative feedback, wherein the species protein\_small inhibits its own production. The
Figure 2.14: Comparisons of statistical parameters as a function of time for mRNA numbers in the gene expression system. Shown are the original model with 71 reactions and 47 species (solid line), the reduced model with 28 reactions and 29 species (dashed line), and the reduced model with 10 reactions and 10 species (dotted line). (a) Mean number of mRNA. (b) Coefficient of Variation (CV). (c) Skewness. (d) Kurtosis.
**Figure 2.15:** Comparisons of the steady-state frequency histogram for mRNA and protein in the gene expression system. Shown are the original model with 71 reactions and 47 species (solid line), the reduced model with 28 reactions and 29 species (dashed line), and the reduced model with 10 reactions and 10 species (dotted line). (a) Frequency histogram for mRNA, for the original and reduced models. (b) Frequency histogram for protein, for the original and reduced models.

**Figure 2.16:** Reduction of the gene expression model with negative feedback (12 reactions, 11 species), again given in the form of the BioNetS interface screen, with columns labelled as in Figure 2.3.
Figure 2.17: Comparisons of statistical parameters as a function of time for Protein numbers in the gene expression system. Shown are the original model with 73 reactions and 48 species (solid line), and the reduced model with 11 reactions and 12 species (dashed line). (a) Mean number of mRNA. (b) Coefficient of Variation (CV). (c) Skewness. (d) Kurtosis.
augmented model has 73 reactions and 48 species; the protein operon complex adds one species to the original model, and its binding and unbinding steps add two reactions. By making the binding constant of protein_small to operon_ns (0.0004) much larger than the binding constant of Rpoly to operon_ns (3.99 × 10^{-6}), most of the operon_ns will be occupied by protein_small, preventing Rpoly from binding to most of the operon_ns sites and thus greatly reducing the level of gene expression. Following the reduction method, we obtain a reduced model with 11 species and 12 reactions, consisting essentially of the 10 species/10 reactions reduced model, plus a reaction representing the feedback. A comparison of the reduced and full models is shown in Figure 2.17. After fitting the reduced model to the original model, the statistical parameters of the protein species show a good match to those in the original model. However, in a more extreme case with much stronger feedback (a binding constant of protein_small to operon_ns of 0.01), the reduced model is no longer be able to capture the exact stochastic behavior of the original model (results not shown). Further investigation will be required to elucidate the details of the conditions under which the method can yield satisfactory results for systems incorporating feedback.

**Rate constants for the reduction of the gene expression model with negative feedback.** The reduction of the gene expression model with negative feedback (to 12 reactions and 11 species, as shown in Figure 2.16) again used simulated annealing to adjust the rate constants to improve the match between the ODE solutions of the reduced and full models. The fitted rate constants (with the values from the original model noted when a direct correspondence exists) are: k\_cell\_div = 4.8 × 10^{-4} (unchanged), k\_rep\_Rpoly = 5.2778 (vs. 5.2), k\_rep\_operon\_ns = 1.0978 (vs. 1.1), k\_rep\_operon\_s = 0.0721 (vs. 0.07), k\_rep\_ribo = 33.3 (unchanged), k\_transcription\_ns = 0.0017 (vs. 0.05), k\_transcription\_s = 0.074 (vs. 0.09), k\_transcription\_small = 1.69 × 10^{-4}, k\_translation = 0.0109 (vs. 0.06), k\_translation\_small = 0.0182, k\_negative = 0.0004 (unchanged), and k\_off\_Negative = 0.001 (unchanged). As above, all zeroth and first order rate constants are given in units of sec^{-1}, while second order rate constants are given in μm³ × sec^{-1}.

### 2.4 Discussion

We have described a method of reducing the complexity of stochastic models of sets of biochemical reactions, and demonstrated that complex systems may be reduced significantly while retaining a good match with the statistical features of the original model. Such reductions offer not only faster computational times, but have the advantage of simplifying the system, making it easier to understand and manipulate. Depending on the requirements of the particular modelling effort, varying degrees of model reduction may be done, preserving the structure of
the original to varying degrees; for example, we have seen that a highly reduced model may preserve only steady-state behaviour rather than transients, while a less-reduced version of the same system can capture both aspects of the behaviour. Application of this approach to other biochemical networks is relatively straightforward, and offers the prospect of a systematic means of stripping away unnecessary detail when examining biological reaction systems.
Chapter 3

Increasing the efficiency of bacterial transcription simulations: when to exclude the genome without loss of accuracy

The work in this chapter has been published in [78]. This project is executed in collaboration with Marco Iafolla, with whom I share the first-authorship on the publication.

3.1 Introduction and background

The internal dynamics of cells are driven by the kinetics of a complex set of biochemical reactions: the state of the cell may be viewed as the numbers and binding states of all species of interest, and the time evolution of that state is defined by how those species react with one another. A central challenge in cellular modelling is to formulate correct biochemical reaction schemes to represent a process of interest, and then to populate the reaction system with appropriate rate constants [5-9]. Within this effort, two persistent difficulties arise: populating mathematical models based on incomplete experimental information [10, 11]; and the computational demands of simulating the resulting systems, which can grow large for even moderately complex processes: after we can figure out the major molecular reactions inside an Escherichia coli cell, simulating the major molecular events inside an Escherichia coli cell will lead to a very large number of reactions that compose its overall behaviour, and require quite a lot computational time.

However, what we usually are interested in is only a small part of the behavior of the whole cell - the gene expression from plasmid-borne promoter in our case. Having the rest
of the genome present in the system, even in our bulk-averaged way, added significantly to the computational demands of the simulations. Further investigation shows, however, that there are regimes in which the target system is not significantly affected by the presence of the remainder of the genome, and may thus well approximated by excluding the genome portion and simulating only the target system. The key quantity is the “on rate” of binding between RNA polymerase and the promoter of the target gene: for certain ranges of this parameter, the perturbation introduced by the presence of other genes (the rest of the genome in the cell) is small enough to be neglected, saving significant amounts of computational time. To identify this range, we compare the total number of transcripts produced from a plasmid vector generated as a function of this rate constant, for two versions of our gene expression model, one incorporating the host cell genome and one excluding it. By sweeping parameters over a four-dimensional parameter space: 24 min $\leq$ bacterial doubling time $\leq$ 100 min, $10 \leq$ plasmid copy number $\leq 1000$, $2 \text{ min} \leq \text{mRNA half-life} \leq 14 \text{ min}$, and $10 \text{ bp} \leq \text{gene length} \leq 10000 \text{ bp}$, we identify the $k_{\text{on}}$ range for which the difference between the genome and no-genome models drops below 5%. A simple MATLAB user interface generates an interpolated $k_{\text{on}}$ threshold for any point in this range; this rate can be compared to the ones used in other transcription studies to assess the need for including the genome.

Exclusion of the genome is shown to yield less than 5% difference in transcript numbers over wide ranges of values, and computational speed is improved by two to 24 times by excluding explicit representation of the genome. Although our results are obtained for our particular gene expression model, we anticipate that our promoter on-rates will apply, at least approximately, to other studies of transcription in bacteria, and thus offer guidance to others wishing to simplify their system by omitting the genomic influence.

3.2 Method

3.2.1 E. coli gene expression model

Our technique relies on the existence of experimental results [2] reporting bulk average assays of the amounts of each species present in the biological system of interest, as a function of growth rate; quantities such as average RNA polymerase per cell, average transcript content per cell, and so on, are much more readily obtained than specific rate constants for individual reactions. Using the bacterium Escherichia coli as a model organism, we have formulated a picture of the biochemical reactions underlying gene expression from an inserted plasmid carrying a promoter controlling the transcription of our target gene. A full list of the reactions included in the model and the nomenclature used for the species is provided in Tables 3.1, 3.2, and 3.3, and the reactions are shown schematically in Fig. 1.
Species Name | Species | Description
---|---|---
Rpoly | RNA polymerase | 
Rpoly_reporter | RNA polymerase elongating the reporter mRNA transcript from the reporter gene | 
closed_Rpoly_prom_reporter | RNA polymerase in a closed-complex with the reporter promoter | 
deg_mRNA_incom_reporter | Nascent reporter mRNA degradation product | 
deg_mRNA_reporter | Reporter mRNA degradation product | 
imcom_mRNA | Nascent reporter mRNA | 
mRNA_reporter | Reporter mRNA | 
open_Rpoly_prom_reporter | RNA polymerase in an open-complex with an mRNA reporter promoter | 
plas | Promoter on the plasmid | 
v | Counter (representing cell volume) | 
Rpoly_operon_ns | RNA polymerase elongating an average mRNA transcript from a template operon | 
Rpoly_operon_s | RNA polymerase elongating an average RNA transcript from a template operon | 
closed_Rpoly_prom_ns | RNA polymerase in a closed-complex with an mRNA operon promoter | 
closed_Rpoly_prom_s | RNA polymerase in a closed-complex with an mRNA operon promoter | 
mRNA | Average mRNA (gene length) | 
mRNA_small | Approximately 90% of the average mRNA | 
open_Rpoly_prom_ns | RNA polymerase in an open-complex with an mRNA operon promoter | 
open_Rpoly_prom_s | RNA polymerase in an open-complex with a stable RNA operon promoter | 
operon_ns | Average mRNA operon | 
operon_s | Average stable RNA operon | 
stable_RNA | Average stable RNA (full operon length) | 

Table 3.1: List of Species Names in the Gene Expression Model

<table>
<thead>
<tr>
<th>Left</th>
<th>Right</th>
<th>Forward Rate</th>
<th>Backward Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>operon_ns + Rpoly</td>
<td>closed_Rpoly_prom_ns</td>
<td>k_on_Rpoly_prom_ns(v/1000)</td>
<td>k_off_Rpoly</td>
</tr>
<tr>
<td>closed_Rpoly_prom_ns</td>
<td>open_Rpoly_prom_ns</td>
<td>k_isomerization</td>
<td></td>
</tr>
<tr>
<td>open_Rpoly_prom_ns</td>
<td>operon_ns + Rpoly_operon_ns_1</td>
<td>k_prom_degradation</td>
<td></td>
</tr>
<tr>
<td>Rpoly_operon_ns_1</td>
<td>Rpoly_operon_ns_2 + mRNA</td>
<td>k_transcription_ns</td>
<td></td>
</tr>
<tr>
<td>Rpoly_operon_ns_2</td>
<td>Rpoly_operon_ns_3 + mRNA</td>
<td>k_transcription_ns</td>
<td></td>
</tr>
<tr>
<td>Rpoly_operon_ns_3</td>
<td>Rpoly_operon_ns_4 + mRNA</td>
<td>k_transcription_ns</td>
<td></td>
</tr>
<tr>
<td>Rpoly_operon_ns_4</td>
<td>Rpoly_operon_ns_5 + mRNA</td>
<td>k_transcription_ns</td>
<td></td>
</tr>
<tr>
<td>Rpoly_operon_ns_5</td>
<td>Rpoly_operon_ns_6 + mRNA</td>
<td>k_transcription_ns</td>
<td></td>
</tr>
<tr>
<td>Rpoly_operon_ns_6</td>
<td>Rpoly_operon_ns_7 + mRNA</td>
<td>k_transcription_ns</td>
<td></td>
</tr>
<tr>
<td>Rpoly_operon_ns_7</td>
<td>Rpoly + mRNA_small</td>
<td>k_transcription_ns/0.871948719487194871</td>
<td></td>
</tr>
<tr>
<td>operon_s + Rpoly</td>
<td>closed_Rpoly_prom_s</td>
<td>k_on_Rpoly_prom_s(v/1000)</td>
<td>k_off_Rpoly</td>
</tr>
<tr>
<td>closed_Rpoly_prom_s</td>
<td>open_Rpoly_prom_s</td>
<td>k_isomerization</td>
<td></td>
</tr>
<tr>
<td>open_Rpoly_prom_s</td>
<td>operon_s + Rpoly_operon_s</td>
<td>k_prom_degradation</td>
<td></td>
</tr>
<tr>
<td>Rpoly_operon_s</td>
<td>Rpoly + stable RNA</td>
<td>k_transcription_s</td>
<td></td>
</tr>
<tr>
<td>v</td>
<td>2v</td>
<td>k_cell_div</td>
<td></td>
</tr>
<tr>
<td>v</td>
<td>operon_s</td>
<td>k_rep_operon_s</td>
<td></td>
</tr>
<tr>
<td>operon_s</td>
<td>k_rep_operon_s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>plas + Rpoly</td>
<td>closed_Rpoly_prom_reporter</td>
<td>k_on_Rpoly_prom_reporter(v/1000)</td>
<td>k_off_Rpoly</td>
</tr>
<tr>
<td>closed_Rpoly_prom_reporter</td>
<td>open_Rpoly_prom_reporter</td>
<td>k_isomerization</td>
<td></td>
</tr>
<tr>
<td>open_Rpoly_prom_reporter</td>
<td>plas + Rpoly_reporter + incom_mRNA_reporter</td>
<td>k_prom_degradation</td>
<td></td>
</tr>
<tr>
<td>Rpoly_reporter</td>
<td>Rpoly + mRNA_reporter</td>
<td>k_transcription_reporter</td>
<td></td>
</tr>
<tr>
<td>incom_mRNA_reporter</td>
<td>k_transcription_reporter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mRNA_reporter</td>
<td>k_deg_mRNA_reporter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>incom_mRNA_reporter</td>
<td>k_deg_mRNA_reporter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rpoly_reporter</td>
<td>Rpoly</td>
<td>incom_mRNA_reporter_k_deg_mRNA_reporter</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.2: List of Reactions in the Gene Expression Model with Gene Expression from Genome
Table 3.3: List of Reactions in the Gene Expression Model without Gene Expression from Genome

<table>
<thead>
<tr>
<th>Left</th>
<th>Right</th>
<th>Forward Rate</th>
<th>Backward Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>v</td>
<td>2v</td>
<td>k_cell_div</td>
<td></td>
</tr>
<tr>
<td>Rpoly</td>
<td>Rpoly</td>
<td>k_rap_Rpoly</td>
<td></td>
</tr>
<tr>
<td>plas + Rpoly</td>
<td>closed_Rpoly_prom_reporter</td>
<td>k_on_Rpoly_prom_reporter/(v/1000)</td>
<td>k_off_Rpoly</td>
</tr>
<tr>
<td>closed_Rpoly_prom_reporter</td>
<td>open_Rpoly_prom_reporter</td>
<td>k_isomerization</td>
<td></td>
</tr>
<tr>
<td>open_Rpoly_prom_reporter</td>
<td>plas + Rpoly_reporter + incom_mRNA_reporter</td>
<td>k_prom_clearance</td>
<td></td>
</tr>
<tr>
<td>Rpoly_reporter</td>
<td>Rpoly + mRNA_reporter</td>
<td>k_transcription_reporter</td>
<td></td>
</tr>
<tr>
<td>incom_mRNA_reporter</td>
<td>k_deg_mRNA_reporter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mRNA_reporter</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rpoly_reporter</td>
<td>Rpoly</td>
<td>incom_mRNA_reporter*k_deg_mRNA_report er/Rpoly_reporter</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.1: Schematic of the two versions of the model system. Our simulations compare two versions of a bacterial gene expression model. (A) In the first version, the genome is represented as an “average” open generating generic transcripts, rather than as the full set of individual genes. Bulk experimental measurements are used to generate the correct average number of transcripts from this generic operon in the genome. In this version of the model, the promoter residing on a plasmid of interest (plas) competes with the genomic operons for access to RNA polymerase (Rpoly) enzymes. (B) In this version, all references to the host cell genome are excluded from the model, leaving only the plasmid-borne promoter (plas) to be transcribed by RNA polymerase (Rpoly).
Below is a detailed explanation of the gene expression model, expanding on the information presented in the Methods section. A full list of kinetic parameters for each reaction is provided in Iafolla and McMillen [1].

**Cellular division**

To reflect the exponential growth of bacterial cells in a nutrient-rich liquid culture, we include cell growth and division, incorporated as a process that grows to a threshold volume and is then halved. At division, all species have their numbers cut approximately in half: for large numbers, a binomial distribution is used to calculate the new number, while small numbers (less than 100) have each molecule explicitly checked and randomly assigned to a daughter cell with equal probability [15]. The model follows only one cell as a representative of the full population, so the second daughter effectively vanishes after division. Tracking such a representative cell over long times yields the same statistics as tracking an ensemble of many cells over shorter times, if we make the reasonable assumption that the system is ergodic.

Cell volume is represented by a “counter” species, v, whose exponential growth is governed by the following reaction, with rate constants adjusted to produce various doubling times to match the experimental conditions being examined:

\[ \frac{d v}{dt} = (k_{\text{on}} \frac{v}{V_0}) - (k_{\text{off}} \frac{v}{V_0}) \]  

For a doubling time $\tau$, the rate constant is set to $k = \frac{\ln 2}{\tau}$. The reaction is initialized at $v = V_0$, and cellular division occurs when $v$ reaches $2V_0$. Our model treats all processes as stochastic, but the resulting degree of variability depends strongly on the number of molecules participating in the reaction. The range of cell division times can thus be tuned by the choice of $V_0$; here we set $V_0 = 1000$, which yields a very slight degree of variability in the cell division times. This variability arises from two sources: the stochastic rate of this reaction, and the random assortment of the counter $v$ between daughter cells at division: $v$ is cut only approximately in half at cell division, like all other species, and thus the initial volume after cell division lies in a small range around $V_0$.

**Enzyme binding, unbinding, isomerization and clearance**

Since the only enzymes used in this model are RNA polymerases only binding to promoters need consideration. The bimolecular reactions for RNA polymerase (Rpoly) binding to a promoter on a gene (plas or operon) are shown below:

\[ Rpoly + plas \leftrightarrow \text{close}_{Rpoly} \_prom \quad K = \frac{k_{\text{on}} \text{Rpoly}}{k_{\text{off}} \text{Rpoly}} \]  

(3.2)
RNA polymerase initially forms a closed-complex with the promoter region, which then undergoes isomerization into an open-complex. The rate constants are adjacent to the reactions; that for 3.2 is scaled to mimic dilution of cell cycle progression: as the cell grows, the increase in volume decreases the probability of the two species coming into contact and reacting, effecting reducing the rate constant [15]; this effect is incorporated by dividing the rate constants by \( v/V_0 \). Following binding, the enzyme clears the promoter at a particular rate. The elementary reactions for this process are shown below:

\[
\text{open}_\text{-}R\text{poly}_\text{prom} \rightarrow R\text{poly}_\text{operon} + \text{operon} + \text{incom}_\text{-}mRNA \quad (3.4)
\]

We create a nascent transcript (mRNA_incom) at this step to allow subsequent translation to proceed; this feature will prove very helpful in studying future simulated studies of protein synthesis. Reaction 3.4 also shows an important assumption: the regeneration of a binding site after clearance allows another enzyme to bind to the same gene, creating the multiple simultaneous elongation processes observed in actual bacterial cells.

**Elongation**

To avoid the computational complexity of accounting for all elongating intermediates (growing mRNA and peptides of every possible length), the following approximation has been employed: a single intermediate is converted to the final product at a rate corresponding to the average time taken by the complete polymerization process. Using average elongation rates for specific cell growth rates as specified by Bremer and Dennis [2], the elongating species produce only the enzyme and the polymerized product, not the template that is read. This is shown below:

\[
R\text{poly}_\text{operon} \rightarrow R\text{poly} + mRNA \quad (3.5)
\]

Compliment to this reaction is the disappearance of the nascent transcript made available during transcription: incom_mRNA \( \rightarrow \emptyset \), where \( \emptyset \) is a null placeholder. The elongation rate constant can be summarized as kelongation = \( \frac{\rho}{\lambda} \), where \( \rho \) and \( \lambda \) are the polymerization rate and length of template, respectively.

**Enzyme and genome production**

Many processes involved in molecular biology are either too complex to model or not characterized at present. In our model, we use simplified zeroth-order production rates for complicated species involved: although the assembly details of some species are not fully available, there is
considerable information on population size of these species. In E. coli, the average number of RNA polymerases and genome equivalents per cell are known at several cellular growth rates [2], and their production is represented by the elementary reactions below:

\[ \emptyset \rightarrow R\text{poly} \quad (3.6) \]

\[ \emptyset \rightarrow \text{plas} \quad (3.7) \]

\[ \emptyset \rightarrow \text{operon} \quad (3.8) \]

The operon species in 3.8 is representative of the genome, since our model employs RNA polymerase binding directly to the promoter sequence of the average operon. The rate constant for production can be summarized as 

\[ k_{\text{rep}} = \left( \frac{v}{1.5} \right) / \tau \]

where \( v \) and \( \tau \) are the average number per cell and cellular doubling time, respectively.

**mRNA degradation**

The presence of RNases in E. coli implies that mRNA possess a finite life-span. The following reactions are used to represent mRNA degradation:

\[ \text{mRNA}_{\text{reporter}} \rightarrow \emptyset \quad (3.9) \]

\[ \text{incom}_{\text{mRNA}}_{\text{reporter}} \rightarrow \emptyset \quad (3.10) \]

For a half-life \( h \), the rate constant for 3.9 and 3.10 is set to 

\[ k = \ln(2) / h. \]

We assume that RNases can degrade nascent transcripts. To account for degrading a transcript while it is being created we propose the following elementary reaction and rate constant:

\[ \text{Rpoly}_{\text{mRNA}}_{\text{reporter}} \rightarrow \text{Rpoly} \quad k_{R11} = \text{incom}_{\text{mRNA}}_{\text{reporter}} \cdot \frac{k_{\text{mRNA} \_ \text{degradation}}}{\text{Rpoly}_{\text{mRNA}}_{\text{reporter}}} \quad (3.11) \]

The reaction indicates that an RNA polymerase currently producing a transcript becomes an unscathed RNA polymerase and a degraded mRNA. Although this reaction implies that all RNA polymerases producing a transcript are subject to degradation, the proportionality to incomplete transcripts is specified in the rate constant. The Rpoly_mRNA species present in the denominator of the rate constant makes the reaction rate independent of the number of elongating RNA polymerases.
Modelling RNA production from operons

We assume that all genes in our relevant genome are clustered into operons. Our model creates a single transcript for the entire operon, mimicking the lac operon [16]. To make the elementary reactions simple and accurate for mRNA and subsequent peptide production, RNA polymerase binds once to the promoter and produces a transcript of average length under corresponding kinetics; the ejection of the mRNA occurs simultaneously with RNA polymerase transcribing the adjacent gene on the operon, or in the case of the last gene on the operon, being released. This is shown in the following reactions for a hypothetical three gene operon, where the binding (3.2), isomerization (3.3) and clearance steps (3.4) have been omitted:

\[ R_{poly\_operon1} \rightarrow R_{poly\_operon2} + mRNA k = k_{transcription} \]

\[ R_{poly\_operon2} \rightarrow R_{poly\_operon3} + mRNA k = k_{transcription} \]

\[ R_{poly\_operon3} \rightarrow R_{poly} + mRNA k = k_{transcription} \]

The numeric suffix on the Rpoly_operon species represents the gene number adjacent to the promoter. Notice that the rate constants for the above reactions are all equivalent. The release of the mRNA while the RNA polymerase is still elongating the operon allows ribosomes to bind and perform translation without requiring additional species; the act of transcription is conserved since RNA polymerase only binds once to the promoter. Evidently, the total time to transcribe all three genes is equivalent to the time for transcribing the whole operon. Contrast to mRNA production, stable RNA is easily produced. Since this RNA is not translated there is no need to include ribosomes translating complete transcripts before the operon is finished elongation. Hence, the length of stable RNA in the model is equivalent to the average stable RNA operon length.

With-genome and no-genome models

We have constructed two versions of the model, one containing a representation of the host cell genome and the reporter gene, the other neglecting the cellular genome and representing only the reporter gene on the plasmid. The with-genome model incorporates 26 reactions involving 27 species, while the no-genome version has 10 reactions involving 10 species; the two versions are shown schematically in Figures 3.1. The genome affects a plasmid-borne gene of interest by competing for RNA polymerase binding with the plasmid-borne promoter, while in the no-genome version of the model we omit the genomic promoter sites and thus this competition does
not occur. The goal, then, is to determine the parameter regimes in which this omission has an acceptably small influence on the behaviour of the system, and to determine how much more quickly the computational simulations will run as a result of the simplification.

3.2.2 Modelling approach and software

The chemical kinetics of this system were initially simulated using the Gillespie Monte Carlo algorithm [15-17], and these results were used to validate a deterministic, ordinary differential equation (ODE) version of the system, which was shown to yield identical average transcript numbers, allowing us to use the significantly faster ODE model to generate larger numbers of points in parameter space. Comparing the two models allowed us to determine the point at which the on-rate constant between the target promoter and RNA polymerase, $k_{on}$, crossed a threshold where the two models (with and without the host genome included) generated average transcript numbers differing by more than a certain percentage; here, we have chosen a five percent difference as an admittedly arbitrary significance threshold.

The original experimental measurements in the literature were carried out over a range of cellular growth rates, each of which yielded different average quantities of biomolecules per cell. Stochastic simulations of our system were carried out at each experimentally-examined growth rate (doubling times of 24, 30, 40, 60, and 100 minutes [2]) and sampled at discrete points in parameter space, as follows: plasmid copy numbers of 10, 100, and 1000; mRNA half-lives of 2, 6, 10, and 14 min; and gene lengths of 10, 100, 1000, and 10000 bp. The relationship between these independent variables and the point at which the promoter-RNAP on-rate begins to yield a significant difference between the genome and no-genome models is complex and highly nonlinear, and not amenable to reduction to a single equation. We have instead produced a MATLAB script (The MathWorks, Natick, MA) that generates an on-rate threshold given a user’s input of plasmid copy number, mRNA degradation rate, gene length and cellular doubling time: any promoter on-rate constant larger than this predicted value can exclude the computationally expensive genome from the simulations without creating more than a five-percent error, while any constant smaller than this should include the genome.

Deterministic chemical kinetics apply in the regime of large numbers of randomly interacting molecules. Inside cells, molecule numbers are often small enough to produce significant fluctuations [11, 12, 17-33], thus requiring a stochastic simulation of the reaction kinetics. The Gillespie algorithm [2] treats chemical reactions as Poisson processes, with event (reaction) rates given by microscopic rate constants and the current state of the system. For an elementary reaction of the form $A + B \rightarrow C$ with rate constant $k$, the Poisson rate of the forward reaction is $kab/V$, where $a$ and $b$ represent the numbers of molecules of species $A$ and $B$ present, and $V$ is the reaction volume (note that this volume is a changing parameter in a living bacterial cell). We
use the unit “n” to represent the number of molecules present in the system, rather than concentration units such as molarity. To advance the simulation, the timing of the next reaction event is randomly selected using the exponential distribution of inter-event times for the set of Poisson processes representing the reactions, and the probability of each reaction being the one that occurs at that instant is given by its fraction of the sum of all reaction rates [15-17].

Bacterial cells have often been approximated as well-stirred reactors: based on their small size, it is assumed that diffusion is sufficiently fast to yield a well-mixed system. Early experimental results showed protein mobility in vivo consistent with normal diffusion [34], and though the diffusion coefficients were substantially lower than for the same proteins in water, the diffusion was fast enough to spread the proteins over the volume of a bacterium on a time scale of seconds. Recent theoretical treatments [32, 35-38] have questioned the picture of bacterial cells as well-mixed systems, and recent experimental results [39] have reported subdiffusive behavior in the motion of individual RNA molecules, where each RNA is rendered visible through binding to multiple fluorescent protein labels. In this paper, we use the well-stirred reactor picture as a first approximation to gain insight, but it should be noted that this is a significant simplification, and that future refinements and extensions are possible. Approaches proposed to deal with crowded cellular environments include rate laws obeying fractal-like kinetics [38, 40, 41], and Monte Carlo simulations wherein two- or three-dimensional spatial information is retained for each molecule [32, 35, 38, 42].

The gene expression model was initially implemented using BioNetS (Biochemical Network Stochastic Simulator) [15], which provides a convenient interface for specifying reactants, products and kinetic data. The software generates C++ source code implementing the system using the Gillespie stochastic simulation algorithm (or an approximation, if desired), and this code is then compiled and executed with user-tunable parameters as inputs. Some species in the model exist in small numbers while others exist in large numbers; although continuum approximations and hybrid schemes are available through BioNetS [15], the Gillespie algorithm with no approximations yielded the best simulation speed. The data from the BioNetS-generated code was processed using DataTask (Visual Data Tools, Inc) and its run manager DataTask, which automated the process of sweeping parameter values and analyzing the results.

3.2.3 Additional deterministic simulations

The stochastic simulations are very computationally intensive, and thus we investigated methods of speeding up the calculations. The ordinary differential equations corresponding to the full reaction system for each model (genome and no genome) were derived using standard chemical kinetics and solved numerically using the solvers provided by MATLAB. To take cell growth and division into account, the ODEs were solved one cell cycle at a time, with the numbers of
molecules at the end of the cycle cut in half to simulate division, then used as the initial state for the next cell cycle. Within each set of parameter values, each ODE was run for ten cell cycles to allow the system to reach a steady state, then for more ten more cell cycles, during which state values were averaged to obtain the average mRNA numbers for the reporter gene. As shown in Figure 3.2, the average mRNA numbers from the stochastic simulations matched nearly perfectly with those generated by the ODEs, and on this basis we used the deterministic ODEs to increase the number of points in the parameter space that could be feasibly sampled. (This reduction to the deterministic model is possible because here we are considering only the mean values from the stochastic simulation; in cases where the fluctuations were the point of interest, fully stochastic simulations would of course be required.) Full-scale stochastic simulations were carried out for the experimentally available doubling times (24, 30, 40, 60, and 100 minutes [2]), varying the other parameters as follows: gene lengths of 10, 100, 1000, and 10000 base pairs (bp); mRNA half-lives of 2, 6, 10, and 14 minutes; and plasmid copy numbers of 10, 100, and 1000 per cell. These were supplemented by deterministic simulations for the same doubling times, at the following parameter values: gene lengths from 10 to 100 in steps of 10 bp, from 100 to 1000 in steps of 100 bp, and from 1000 to 10000 in steps of 1000 bp; mRNA half-lives from 2 to 14 minutes in steps of 1 minute; and plasmid copy numbers from 1 to 9 in steps of 1, from 10 to 100 in steps of 10, and from 100 to 1000 in steps of 100 copies per cell.

Similar to the parameter sweeping carried out for the stochastic simulations, we used the deterministic simulation results for each parameter set to calculate the RNA polymerase-promoter binding on rate, $k_{on}$, at which there will be a five percent difference between the models with and without a representation of the host cell genome; for the deterministic results, the 5% threshold was determined using the fzero function in MATLAB, which searches for a zero-crossing between two given points.

### 3.2.4 Parameter sweeping

The first step in deriving the on-rate constant that determines a 5% difference in transcript averages between models is to obtain steady-state values of all species in the simulations. Figure 3.3 shows the time series for one species in the model, the reporter mRNA. An initial run of 10 cell divisions in length is generated for each simulation, and the final state of this run is used as the initial state for the long-duration run in which statistics are accumulated to determine average species levels; this prevents the initial transient approach to steady state from distorting the averages.

Parameter sweeping begins by using on-rates that vary by a factor of 10 (Figure 3A). When the desired percent difference between models lies between two on-rate constants, another sweep is performed between these new limits incrementing the on-rate by a unit multiple of the smaller
Figure 3.2: Comparison of stochastic and deterministic simulation outputs. The stochastic simulations required too much computational time for it to be practical to sample the parameter space very densely. Since we have used only the mean values from the stochastic simulations, we explored the possibility of using deterministic simulations, which require a tiny fraction of the stochastic simulation time, to increase our sampling of the parameter space. The plot shows the average number of mRNA transcripts generated by the two methods, stochastic and deterministic. The straight diagonal line indicates a good match, and in fact the two methods differ by less than one percent in most cases.
Figure 3.3: Typical time series generated by the model. Plot A (left) shows a run with all intermediates and products initially set to zero, illustrating the initial transient. Plot B (right) shows a run initialized with the state obtained after 10 cell divisions in the left-hand run, thus removing the initial transient. Parameters for this example: doubling time = 24 min; on-rate constant = $10^{-3} \text{n}^{-1} \text{s}^{-1}$; plasmid copy number = 10; gene length = 104 bp; and mRNA half-life = 14 min.

The third parameter sweep uses a unit increment of the next significant digit between the new limits; this change in on-rate is small enough to approximate linearity (Figure 3.4). Only $R^2 \geq 0.90$ were accepted for interpolation; the range was narrowed until this level of linearity was achieved.

The duration of the stochastic simulations was varied to obtain linearity with $R^2 \geq 0.90$; this is achieved by using a minimum of 1000 cell divisions, although some simulations use more cell divisions to obtain the desired linearity. Since the doubling time of the cells is varied, the total duration in real time varies among the simulations; the number of cell divisions explored appears to be the key factor in obtaining well-converged statistics, rather than the absolute duration.

The minimum 1000 cell division duration was deduced by qualitative analysis of multiple simulations with the same seed but different durations; we examined the effect of duration on the mean values obtained from the reporter mRNA histograms. The on-rate constants used in the duration analysis was determined by comparing the histograms between models over a range of on-rate constants ($10^{-7} \text{n}^{-1} \text{s}^{-1}$ to $1 \text{n}^{-1} \text{s}^{-1}$); the range of on-rate constants that bound the percent difference in the above statistical parameters by 5% was investigated for duration
Figure 3.4: Parameter sweeping. Here, we compare two versions of the gene expression model, one incorporating the host cell genome and one excluding it. The RNA polymerase on-rate constant for binding to the promoter that produces the reporter mRNA is varied until the percentage difference between these models exceeds 5% (the value we have selected as our threshold for a significant difference between the two models, marked by a horizontal dashed line on each plot). The on-rate is first varied by a factor of 10 to determine the general location of the desired value (plot A, left), followed by a sweep on a finer scale to narrow in on an approximately linear region near the threshold crossing (plot B, right). The solid vertical line in Graph B shows the interpolated on-rate constant when the percent difference in transcript production between models crosses the 5% threshold. The parameters for this example are: doubling time = 24 min; plasmid copy number = 10; gene length = 10 bp; and mRNA half-life = 6 min.

analysis. Ultimately, longer-duration runs produced averages that were not statistically different from those obtained after 1000 divisions (see the Appendix for additional explanation), implying that longer durations only increase computational expense.

After interpolation, the validity of the on-rate was tested: using a different seed for 30 simulations – all employing steady-state initial conditions and the same duration, kinetics and interpolated on-rate – the sample mean difference between models of the 30 simulations was statistically compared to the population mean of 5%. The on-rate was accepted if the two means were not proven statistically different using a level of significance $\alpha = 0.95$. All simulations, either in parameter sweeping or verification, employ different nucleating random number generator seeds.
3.2.5 Interpolation of on-rate thresholds

The on-rate ($k_{on}$) threshold above which a 5% deviation between the genome and no-genome models occurred has been calculated explicitly only at the set of parameter values listed above (based on stochastic simulations supplemented by cross-validated deterministic simulations to increase the density of the sampling of parameter space). To allow the $k_{on}$ threshold to be calculated at values other than those explicitly simulated, we created a MATLAB script to carry out the necessary interpolation using a local minimization method. In local linear fitting, to find the unknown point at a desired parameter value, one draws a straight line connecting the known points on either side of the desired value, and takes the point on that straight line as the interpolated result at the desired parameter value. Note that this process minimizes the total distance between the interpolated point and the two known points, and we use this idea to perform our interpolation in our 5-dimensional space ($k_{on}$ as a function of four parameters: growth rate, gene length, mRNA half life, and plasmid copy number). For any single given 4-dimensional parameter set, the nearest available set of parameter values is determined by finding the two nearest parameter values in each direction on this 4-dimensional mesh; combining all four dimensions yields the 16 nearest points on the mesh. Since these 16 data points do not generally fit well to a linear function, we obtain the interpolated on-rate value for a given parameter set by searching for the $k_{on}$ value that minimizes the total distance in 5-dimensional space to those nearest 16 points, using the MATLAB fminsearch function to carry out the minimization operation.

The above interpolation has been implemented in MATLAB script that presents a simple user interface allowing the user to enter the desired parameter values (within the ranges spanned by the simulations), after which the script will carry out the interpolation for the given point and return the $k_{on}$ value above which a 5% difference arises between the genome and no-genome models: any promoter on-rate constant larger than this predicted value can exclude the computationally expensive genome from the simulations without creating more than a five-percent error, while any constant smaller than this should include the genome. The user interface is shown in Figure 5, and the MATLAB files required to implement it are available for download as part of the Additional Files associated with this paper.

3.3 Results and discussion

3.3.1 Percent difference of reporter transcript averages between models

As shown in Figure 3.4, using the stated parameters as a representative example, the percent difference of reporter transcripts between models changes as a function of binding constant be-
tween RNA polymerase and the target promoter (k_on). An excessively small binding constant 
\( \approx 10^{-10} n^{-1} s^{-1} \) to \( 10^{-7} n^{-1} s^{-1} \) prevents the RNA polymerase from binding to the promoter, thereby producing an insignificant number of transcripts, usually less than one per cell division, as shown in Figure 6. The constant can be so small that noise dominates the system, leading to essentially random results, including some in which more reporter transcripts are produced in simulations that use the genome, relative to the simulations that only use the plasmid-borne reporter genes. Eventually the binding constant becomes large enough to produce a considerable quantity of transcripts; at this point the genome’s presence competes with the reporter gene for access to RNA polymerase and reduces the transcription of the reporter gene, producing a significant percent difference between models. As the binding constant to the reporter promoter further increases, the RNA polymerase binding saturates and the promoter generates nearly the same number of transcripts with or without the presence of the competing genome; the difference between models trends towards zero as the binding constant approaches infinity.

Figure 3.4 shows there are two binding constant ranges for each set of parameters where there is less than a 5% difference in transcript production. We have not considered the lower range, here, because of the insignificant number of transcripts produced, usually an average of much less than one per cell division. In this regime, the two versions of the model are both matching simply because they are both yielding a result of “nearly zero.” For the case we wish to consider, that of observing the output of a target gene through the expression of a reporter, such low levels of transcription would be invisible to current detection techniques, requiring single-molecule resolution against the noisy background of the cytoplasm, and thus for the moment we consider it justified to exclude this near-zero range in our simulations. The higher k_on rate constant limit corresponds to transcript numbers on the order of \( 10^2 \) to \( 10^4 \), a magnitude that is much more amenable to experimental access and thus potentially more significant for use in other studies.

### 3.3.2 Accuracy of the interpolated on-rates

To test the accuracy of the interpolated on-rates, the on-rates were entered back into both versions of the model and run for 30 different simulations seeds for a duration of 30 cell divisions, after creating steady state values for all species within the model. The percent differences were assembled and statistically compared to the population mean of 5% using a level of significance \( \alpha = 0.95 \). This process was repeated for all 240 different kinetic situations generated using the stochastic simulations. There was no statistical difference between the population mean and the sample mean obtained from the simulations (data not shown), thereby ensuring that the interpolated values are the correct ones for producing a percent difference of 5%.
Figure 3.5: The effect of the genome on the reporter transcript output. At small enough binding constants neither model is able to produce a significant number of transcripts: the average time between transcriptions is much larger than the doubling time, leading to an average of much less than one transcript per cell division. As the binding constant increases, the reporter promoter starts to compete with the genomic promoters for RNA polymerase, ultimately producing a difference in the number of transcripts between models. The above example has been arbitrarily chosen; it uses the same parameters as in Figure 2 (doubling time = 24 min; plasmid copy number = 10; gene length = 10 bp; and mRNA half-life = 6 min). The error bars are a single standard deviation in the transcript number distributions generated by the stochastic simulations.
Figure 3.6: The ratio of simulation time between models with the genome to those excluding it, as a function of doubling time and plasmid copy number. Removing the genome from simulation studies can be 2 to 24 times more efficient compared to those that include it. The data was constructed by averaging the simulation times for all verification runs that employed the set doubling times and plasmid copy numbers, regardless of mRNA half-life and gene length. All computer simulation times were normalized with respect to the computer’s CPU strength. The trends suggest that the ratio will approach 1 for sufficiently long doubling times.
Figure 3.7: Number of reaction steps simulated. Fixing three of the input parameters (plasmid copy number is 10, mRNA half-life is 6 min, and gene length is 1000 bp), we plot the number of reaction steps taken in a stochastic run simulating 9000 seconds of time. (A) With-genome model. The total number of reaction steps, and the number of reactions dedicated to RNA polymerase binding/unbinding to the genomic operons, and to the plasmid carrying our gene of interest. (B) No-genome model. The total number of reaction steps, and the number of reactions dedicated to RNA polymerase binding/unbinding to the plasmid carrying our gene of interest.

3.3.3 Time reduction via genome exclusion

Excluding the genome from simulation studies does reduce CPU simulation time in the computationally intensive fully stochastic simulations. To illustrate this, the verification runs were used for comparison between models; these simulations employ the same kinetic parameters and duration, and offer a large population size (since each run was repeated multiple times with varying random seeds).

Dividing the average run time of the genome by those models excluding it produces a direct measure of the benefit of excluding the host cell genome in the simulations. As Figure 3.6 shows, computational time can be reduced by a factor ranging from two to 24-fold. Accurate analysis of the time saved between models requires standard CPU power. The verification simulations in this study have been spread out over many computers, most of which have different CPUs. To normalize the results, 10 replicates of a standard run with the same kinetic parameters, duration and random number seed was run (with minimal other processor load) on each type of CPU, for each version of the model. The simulation duration was set to take approximately 30 minutes of CPU time, to average away any aberrations caused by minor fluctuations in CPU availability over time. The run durations for these standard runs were then used to create a scaling factor.
for each CPU type, and the simulation times reported in Figure 3.6 were corrected by these factors.

The simulation spends most of its time on the RNA polymerase binding/binding reactions: the reactions \( \text{operon ns+Rpoly}, \text{operon s+Rpoly}, \) and \( \text{plas+Rpoly} \) in the with-genome model, and simply \( \text{plas+Rpoly} \) in the no-genome model. Figure 3.7 shows the number of reaction steps simulated in the with-genome and no-genome versions of the model (keeping plasmid copy number, mRNA half-life, and gene length fixed, while varying cell doubling time). As Figure 3.7 shows, the number of reaction steps dedicated to simulating the genomic RNA polymerase binding operations falls off more rapidly with growth rate than does the number of steps required to simulate the plasmid-to-RNA polymerase binding. Figure 3.7 shows that the number of reaction steps simulated in the no-genome version of the model falls off as a function of growth rate, but less rapidly than in the with-genome case; this is the cause of the reduction in the relative advantage of the no-genome version as the growth rate increases, seen in Figure 3.6. For large plasmid copy numbers, the RNA polymerase binding steps are more time-consuming in the no-genome version of the model, and the computational advantage of excluding the genome is correspondingly smaller; again, this is seen in Figure 3.6.

3.3.4 Relationship between the parameters

Figures 3.8, 3.9, and 3.10 show the dependence of the \( k_{\text{on}} \) value on gene length, plasmid number and mRNA half-life, while the doubling time is fixed at 30 minutes. These plots are 3D slices through the full 5D space of results (where the five dimensions are the four input parameters, mRNA half life, gene length, plasmid number, and doubling time, and the output promoter on-rate, \( k_{\text{on}} \)). The plots show some of the nonlinearity inherent in the relationship of \( k_{\text{on}} \) to the parameters, and help to indicate why it has not proven to be possible to reduce the parameter relationships to a single regression equation.

3.3.5 Potential extensions

Simulating the translation of mRNA to protein, downstream of the transcriptional events discussed here, requires a significantly more elaborate model [1] with correspondingly greater computational demands. One extension of this study would be to investigate the binding on-rates for ribosomes binding to the ribosome-binding-sites (RBS) of the mRNA binding sites, and once again compare the results when the presence of the genome is modelled to those when it is excluded; presumably there would be a similar possibility of excluding the representation of the genome under some parameter ranges (where the main parameters would remain the same: doubling time, gene length, mRNA half-life, and plasmid copy number). Since transla-
Figure 3.8: Dependence of promoter strength on gene length and plasmid number. The full set of simulations yields promoter strengths, $k_{on}$, as a function of four input parameters (gene length, plasmid number, mRNA half-life, and cell doubling time). Here, we fix the doubling time at 30 minutes and the mRNA half-life at 8 minutes, and plot $k_{on}$ as a function of the two remaining parameters: gene length and plasmid copy number.
Figure 3.9: Dependence of promoter strength on plasmid number and mRNA half-life. The full set of simulations yields promoter strengths, k\_on, as a function of four input parameters (gene length, plasmid number, mRNA half-life, and cell doubling time). Here, we fix the doubling time at 30 minutes and the gene length at 4000 base pairs, and plot k\_on as a function of the two remaining parameters: plasmid copy number and mRNA half-life.
Figure 3.10: Dependence of promoter strength on mRNA half-life and gene length. The full set of simulations yields promoter strengths, \( k_{\text{on}} \), as a function of four input parameters (gene length, plasmid number, mRNA half-life, and cell doubling time). Here, we fix the doubling time at 30 minutes and the plasmid copy number at 200, and plot \( k_{\text{on}} \) as a function of the two remaining parameters: mRNA half-life and gene length.
tion follows transcription in the gene expression process, the range of parameter values in which one can exclude the genome from studies of the translational output of a target gene should be smaller than the regions found in the current study of transcriptional output: the system will be subject to the constraints imposed by matching the transcriptional results, as well as additional constraints required to match the translational results.

The ability of RNA polymerase to produce an approximately equal amount of transcripts at large enough binding constants for both models raises an important question: are there enough RNA polymerases left when a large rate law exists for the reporter promoter to transcribe the necessary genomic genes for cell division? The presence of a large rate for the reporter transcript will produce metabolic strain on the cell [15-17], possibly leading to an increase in doubling time that is not captured within the current model. Further studies on modelling the effect of metabolic strain and its feedback with cellular doubling time will help to clarify this issue.

3.4 Conclusion

Efforts to create accurate, quantitative models of Escherichia coli genomic networks using chemical equations results in large reaction schemes, with reactions potentially proceeding at a wide range of rates. The large computational time required to simulate these reactions is a persistent problem for large-scale cellular simulation. To help address one aspect of this problem, we have investigated the necessity of simulating the presence of the E. coli genome when studying a target gene inserted on a plasmid. The presence of the genome, introduced using our “mean-field” approach, is felt by the target gene through the competition for free RNA polymerases available to bind to the target gene’s promoter and generate transcripts. However, there are ranges of the parameter space in which the presence of the genome yields a negligible difference in the number of reporter transcripts produced from the target gene, and in these cases is it possible to exclude any explicit representation of the genome and save the computations required to simulate the associated additional reactions. Stochastic simulations show speed increases of from two to 24 times, when the genome is excluded from our models. We have generated a set of fully stochastic simulations and found the promoter on-rate values for which the genome and no-genome models differ by less than 5%, and augmented these stochastic simulations with cross-validated deterministic runs to increase the number of sampled points in parameter space. Within the ranges of our four independent parameters (growth rate, gene length, mRNA degradation half-life, and plasmid copy number), we have produced a MATLAB user interface that will allow the user to input any set of parameters and obtain the promoter on-rate value (k_on) above which the effect of the genome will fall below our 5%-difference threshold. Given the
increasing computational demands of cellular simulations, we hope that this approach will aid in the efficiency of other studies, and suggest other methods in which portions of the full cellular system may be excluded without significantly affecting the final results.
Chapter 4

Effect of protein maturation on the noise in gene expression

The work in this chapter has been published in [79].

4.1 Introduction

Fluorescent proteins are frequently used as reporters for gene expression in living cells, either by being expressed in tandem with a protein of interest, or through the creation of fusion proteins. The data yielded by the fluorescence output is of considerable interest in efforts to formulate quantitative models of cellular behavior underway in fields such as systems biology and synthetic biology. Models formulated of these gene expression systems connect to experimental results by using the fluorescence intensity at a specified wavelength as an observable presumed to be proportional to the number of proteins being expressed in the cell. While there is a reasonable basis presuming that such a proportionality exists, it is not exact because of the internal dynamics of fluorescent proteins themselves, which can perturb the observed fluorescence away from being a true representation of the state of the proteins in the cell. In particular, the proteins must undergo a process of “maturation” before they become fluorescent; the process involves the folding of the protein, cyclization of a tripeptide motif, and oxidation of the cyclized motif [80–82]. These steps can take from a few minutes (for proteins such as the yellow-fluorescing Venus), to hours (GFP, EGFP), up to over a day (DsRed) [32,33,83]. Until the proteins have finished their maturation, they are invisible to fluorescent detection techniques, implying that in any given cell there will be a population of unobserved proteins in addition to those that are detected. This maturation process, while known and acknowledged, is not generally incorporated explicitly into gene expression models, and this has the potential to impact both estimates of the numbers of proteins present [31,84,85] and estimates of the variability in gene expres-
Chapter 4. Effect of protein maturation on gene expression noise

sion [15,19,22–24,28,29,34]. Here, we examine the effect of fluorescent protein maturation on the observed level and variability of gene expression, by comparing the number and variability estimates in simple models with and without the maturation steps included, the latter being equivalent to instant maturation. We find that the observed number and variability can in fact differ substantially from the true variability, when maturation is included. At low protein production rates, the effect is dominated by the size of fluctuations in the reaction rates, and the increased observed variability is attributable to the decreased rate of production of the matured proteins as compared to the true number of proteins being produced. At higher production rates, however, the maturation steps act as a low-pass filter, removing some of the noise in the protein production process and yielding observed variabilities that are lower than the true values.

We begin by formulating a model of the biochemical processes underlying gene expression, neglecting explicit representation of intermediate steps such as the elongation of transcript and peptide chains and the binding and open complex formation of enzyme initially leaving out of explicit consideration the process of cell growth and division. This simplified system is amenable to analytical treatment, and we present the results of solving the master equation for the stochastic processes representing the chemical reactions. We then examine the noise power spectra for this case, again obtaining analytic solutions that illustrate the low-pass filtering effect of the maturation steps. In summary, our results indicate that while fluorescent proteins are an invaluable tool for characterizing and quantifying gene expression, their output must be interpreted with caution, particular with regard to the observed levels of variability.

4.2 Gene expression model

4.2.1 Analysis of steady-state statistics

The molecular basis of gene expression is well established [86,87]; particularly accessible overviews have been presented by Ptashne and Gann [88,89]. Here, we consider a generalized gene expression model as shown in Fig. 4.1. The process begins with RNA polymerase (RNAP) binding to a DNA strand, forming an open complex, and initiating the process of transcription, wherein a messenger RNA (mRNA) strand is produced, carrying the genetic information. The mRNA is translated into a protein by binding with a ribosome, which decodes the triplet nucleotide codons of the gene into a sequence of amino acids. This model is highly simplified, but it can generate the same statistical behavior as a more elaborate model [55,68]. If the protein in question is a fluorescent reporter, it must undergo three more steps before it can generate fluorescence: folding, cyclization, and oxidation. Each species in chain is also subject to degradation, represented in Fig. 4.1 by processes terminating in the null set symbol.
The gene expression model in Fig. 4.1 may be approximated as a system of elementary biochemical reactions, as follows:

\[
\begin{align*}
RNAP + DNA & \xrightleftharpoons[c_1]{c_1} RNAP + DNA + mRNA \\
\text{mRNA} & \xrightarrow{c_2} \emptyset \\
mRNA + \text{ribo} & \xrightarrow{c_3} mRNA + \text{ribo} + \text{Protein} \\
\text{Protein} & \xrightarrow{c_4} \emptyset \\
\text{Protein-folded} & \xrightarrow{c_4} \emptyset \\
\text{Protein-cyclized} & \xrightarrow{c_4} \emptyset \\
\text{Protein-matured} & \xrightarrow{c_4} \emptyset
\end{align*}
\]

Using the common simplifying assumption that our biochemical reactions occur in a well stirred volume, and neglecting macromolecular crowding effects [90,91], the spatial information about individual molecules may be ignored. In this approximation, the dynamics of the system is describable using ordinary differential equations when the reaction rates are high, but in lower-rate (generally corresponding with lower-number) regimes we must consider fluctuations, treating each reaction as a Poisson process and describing the resulting stochastic process using the chemical master equation [19,50–52,92,93]. Making the further approximation that the numbers of RNAP, DNA, and ribosome (ribo) molecules are constant, the model becomes linear and the master equations may be solved analytically. The state of the system is represented by a five-dimensional vector giving the number of molecules of each species not being kept constant: \(\mathbf{n} = (n_1, n_2, n_3, n_4, n_5)\), where the elements are the numbers of molecules of species mRNA, Protein, Protein-folded, Protein-cyclized, and Protein-matured, respectively. Here, we adopt a symbol system employed by Gadgil [94] to represent the four categories of reactions occurring in the system: production from a source \(\emptyset \xrightarrow{c_i} n_i\); degradation \(n_i \xrightarrow{c_i^d} \emptyset\); conversion \(n_j \xrightarrow{c_{ij}^{con}} n_i\); and catalytic production from a source, \(n_j \xrightarrow{c_{ij}^{cat}} n_i + n_j\), where \(c_i\) or \(c_{ij}\) is the stochastic reaction constant [52]. Using this symbol system, the corresponding reaction-rate equations may be written as

\[
\frac{d\mathbf{n}}{dt} = C^s \times \mathbf{1} - C^d \times \mathbf{n} + C^{cat} \times \mathbf{n} + C^{con} \times \mathbf{n}
\]

where \(\mathbf{1} = (1, 1, 1, 1, 1)^T\), \(C^s = \text{diag}\{c_i^s\}\), \(C^d = \text{diag}\{c_i^d\}\), \(C^{cat} = c_{ij}^{cat}\), and...
Figure 4.1: Schematic depiction of the simplified gene expression model considered here. The enzyme RNA polymerase (RNAP) binds to a DNA strand and produces a messenger RNA strand (mRNA), which is then translated by a ribosome into a protein. Fluorescent proteins must then fold, cyclize, and oxidize before they become “mature” and thus able to fluoresce and be detected experimentally. This schematic is converted into the set of biochemical reactions (4.1) through (4.10).
Figure 4.2: Time series of one realization of the stochastic process defined by biochemical reactions (4.1) through (4.10), generated by a Monte Carlo method [51,52], at transcription rate: 0.46s\(^{-1}\), translation rate: 0.0047s\(^{-1}\), and maturation time: 28 minutes. From top (greatest number of molecules) to bottom (least number), the traces represent the following species: Protein, Protein-folded, Protein-cyclized, and Protein-matured (the fluorescing species visible experimentally); mRNA numbers are not shown. The horizontal lines represent the mean and ±3 standard deviations for each species, calculated analytically by solving Eqs (4.19) and (4.20).
Chapter 4. Effect of protein maturation on gene expression noise

\[ C_{\text{con}}^{\text{ij}} = \begin{cases} c_{\text{con}}^{\text{ij}} : i \neq j \\ - \sum_k c_{\text{con}}^{kj} : i = j \end{cases} \]  

(4.12)

For our model, \( C^{\text{s'}} \), \( C^{\text{d}} \), \( C^{\text{cat}} \), and \( C^{\text{con}} \) are

\[
C^{\text{s'}} = \begin{pmatrix}
  c_1 n_{\text{RNAP}} n_{\text{DNA}} & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0
\end{pmatrix} \equiv \begin{pmatrix}
  c_1 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0
\end{pmatrix}  
\]  

(4.13)

\[
C^{\text{d}} = \begin{pmatrix}
  c_2 & 0 & 0 & 0 & 0 \\
  0 & c_4 & 0 & 0 & 0 \\
  0 & 0 & c_4 & 0 & 0 \\
  0 & 0 & 0 & c_4 & 0 \\
  0 & 0 & 0 & 0 & c_4
\end{pmatrix}  
\]  

(4.14)

\[
C^{\text{cat}} = \begin{pmatrix}
  0 & 0 & 0 & 0 & 0 \\
  c_3 n_{\text{rib}} & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0
\end{pmatrix} \equiv \begin{pmatrix}
  0 & 0 & 0 & 0 & 0 \\
  c_3 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0
\end{pmatrix}  
\]  

(4.15)

and

\[
C^{\text{con}} = \begin{pmatrix}
  0 & 0 & 0 & 0 & 0 \\
  0 & -c_5 & 0 & 0 & 0 \\
  0 & c_5 & -c_6 & 0 & 0 \\
  0 & 0 & c_6 & -c_7 & 0 \\
  0 & 0 & 0 & c_7 & 0
\end{pmatrix}  
\]  

(4.16)

where \( n_{\text{DNA}} \), \( n_{\text{RNAP}} \) and \( n_{\text{rib}} \) represent the number of DNA, RNAP, and ribosomes respectively, and the rate constants are chosen to reflect reasonable values for gene expression in the bacterium \textit{Escherichia coli} [68, 95]. The reaction rates used for the parameter sweep are listed below:
Transcription rate: \( c_1 = c_1^n_{RNAP}n_{DNA} = (4^{1-m_1} \times 2 \times 10^{-19} L^{-1} s^{-1})/v \times 7600 \times 154 = 4^{1-m_1} \times 117 \ s^{-1} \), where \( m_1 = 1...7 \), we take the number of RNAP and DNA to be 7600 and 154 respectively (modelling a medium-copy plasmid bearing many copies of the fluorescent protein of interest) [68,95], and we assume a typical \( E. \) coli cell volume of \( v = 2 \times 10^{-15} \ L \) [96];

Degradation rate of mRNA: \( c_2 = 4.2 \times 10^{-3} \ s^{-1} \), corresponding to a typical bacterial mRNA half-life of 4.8 minutes [87]; Translation rate: \( c_3 = c_3^n_{ribo} = (4^{1-m_2} \times 2 \times 10^{-19} L^{-1} s^{-1})/v \times 48000 = 4^{1-m_2} \times 4.8 \ s^{-1} \), where \( m_2 = 1...10 \), and we take the number of ribosomes to be 48000 [68,95];

Degradation rate of protein: \( c_4 = \ln 2/(1440 \ s) = 4.8 \times 10^{-4} \ s^{-1} \), representing the effective degradation of proteins through dilution by cell growth during a 1440 second (24 minute) cell division cycle; Folding rate: \( c_5 = 4^{2-m_3} \ s^{-1}, m_3 = 1...20 \); Cyclization rate: \( c_6 = 4^{2-m_3} \ s^{-1}, m_3 = 1...20 \); Oxidation rate: \( c_7 = 4^{1-m_3} \ s^{-1}, m_3 = 1...20 \) [80–82]. Note that the connection between the stochastic reaction constants \( c \) and the deterministic reaction-rate constants \( k \) [52] is: \( c_1' = k_1/v; c_2 = k_2, c_3' = k_3/v, c_4 = k_4, c_5 = k_5, c_6 = k_6, \) and \( c_7 = k_7 \), where \( v \) is the volume of the cell.

Throughout the following, we use maturation time rather than rate to characterize the maturation steps, where the maturation time is the average time required to mature a fraction \((1-1/e)\) of the protein. Fluorescent reporter proteins often have long half-lives, on the order of tens of hours or more [32,97,98], and we thus make the further simplifying assumption that the degradation of the proteins considered here is dominated by dilution due to cell growth, in which the protein number is halved after each cell division (a 24-minute cycle in our model \( E. \) coli): this implies that the degradation constant \( c_4 \) is the same for all forms of the protein, and we also take it to be unchanged when we alter the maturation times to represent different fluorescent reporter proteins. It should be noted that rapid degradation of the proteins by means other than dilution through cell growth could place our model into a different regime, and one that we have not addressed, here.

The maturation time is calculated for the case where the protein is subject to no processes other than maturation; that is, we neglect degradation and consider the following three reactions: \( \text{Protein} \rightarrow \text{Protein-folded}; \) \( \text{Protein-folded} \rightarrow \text{Protein-cyclized}; \) \( \text{Protein-cyclized} \rightarrow \text{Protein-matured} \). The average Protein-matured number as a function of time, given an initial protein number \( p_0 \), is

\[
\text{Protein-matured}(t) = \frac{c_5 p_0}{c_5 - c_6} \left( \frac{c_7}{c_7 - c_5} (e^{-c_5 t} - 1) + \frac{c_7}{c_6 - c_7} (e^{-c_6 t} - 1) \right) + \frac{c_5 p_0}{c_5 - c_6} \left( \frac{c_6 (c_6 - c_7)}{c_7 - c_5} (e^{-c_7 t} - 1) \right).
\]

We solve the above equation numerically for the time at which the number of the species Protein-matured will be \((1 - 1/e)p_0\), or equivalently when the immature Protein number will
be $p_0/e$.

The deterministic reaction-rate equations do not capture the stochastic behavior of the system. For this, we represent the probability density associated with each state of the number vector as $P(\mathbf{n})$, and write the master equations for the time evolution of this probability density:

$$
\frac{dP(\mathbf{n},t)}{dt} = \sum_{i=1}^{5} \left[ \mathbf{C}_i^s P(\ldots, n_i - 1, \ldots, t) + \sum_{j=1}^{5} \left( \mathbf{C}_{ij}^{\text{con}} (n_j + 1) P(\ldots, n_i - 1, \ldots, n_j + 1, \ldots, t) \right. \right.
\left. + \mathbf{C}_{ij}^{\text{cat}} (n_j P(\ldots, n_i - 1, \ldots, t) - n_j P(\mathbf{n},t)) \right. \right.
\left. - \mathbf{C}_{ij}^{\text{d}} (n_j P(\ldots, n_i + 1, \ldots, t) - n_j P(\mathbf{n},t)) \right]\right].
$$

Eq. (4.17) is not directly solvable. Exact solutions for the chemical master equation have been found for the case of monomolecular reactions [50], but the presence of the catalytic reaction (4.3) means that our system does not fall into this category. We obtain a partial solution of Eq. (4.17) using moment-generating functions. In this approach, we multiply both sides of the master equations by dummy variables $z_{n_1}^1, \ldots, z_{n_5}^5$ and sum over all equations to obtain the generating function $G(z,t) = \sum_{n_i=0}^{\infty} z_{n_i}^1 \ldots z_{n_5}^5 P(\mathbf{n})$, where $\mathbf{z} = (z_1, z_2, z_3, z_4, z_5)$ [93]. The time evolution of this function is given by

$$
\frac{\partial G(z,t)}{\partial t} = \sum_{i=1}^{5} (z_i - 1) \left( \mathbf{C}_i^s \times G(z,t) + \sum_{i=1}^{5} \left( \mathbf{C}_{ij}^{\text{con}} + \mathbf{C}_{ij}^{\text{cat}} \times z_j - \mathbf{C}_{ij}^{\text{d}} \right) \times \frac{\partial G(z,t)}{\partial z_j} \right) .
$$

(4.18)

Here we focus on the first and second moments of the distribution, taking the first and second derivatives of the generating function,

$$
\mathbf{M}_k(t) = \frac{\partial G(z,t)}{\partial z_j} \bigg|_{z=1} = E[n_k(t)]
$$

(4.19)

and

$$
\mathbf{V}_{ik}(t) = \frac{\partial G(z,t)}{\partial z_i z_k} \bigg|_{z=1} = \begin{cases} E[n_i(t)n_k(t)], & i \neq k \\ E[n_k^2(t)] - E[n_k(t)], & i = k \end{cases}
$$

(4.20)

Eqs (4.19) and (4.20) allow us to determine the coefficient of variation ($\eta$) for each species, defined as the standard deviation divided by the mean. The coefficient of variation of the $i$-th species is

$$
\eta_i(t) = \sqrt{\frac{\mathbf{V}_{ii}(t) + \mathbf{M}_i(t) - \mathbf{M}_i^2(t)}{\mathbf{M}_i(t)}} .
$$

(4.21)
Taking first and second order derivatives of Eq. (4.18) and using the quantities defined by Eqs (4.19) and (4.20), we find equations for the time evolution of the first and second moments of the system,

\[ \frac{dM(t)}{dt} = C \times M(t) + C^s \]  \hspace{1cm} (4.22)

and

\[ \frac{dV(t)}{dt} = C \times V(t) + (C \times V(t))^T + \Psi(t) + \Psi^T(t). \]  \hspace{1cm} (4.23)

where \( C \equiv C^{con} + C^{cat} - C^d \), \( M(t) = (E[n_1(t)], ..., E[n_5(t)]) \), and \( \Psi_{ij}(t) \equiv (C^{cat}_{ii} + C^{s}_{ii})M_{j}(t) \).

Solving for the steady-state solution of the above differential equations yields exact solutions for the first and second moments of the distribution of each species. We first consider an “instant maturation” case in which the protein matures infinitely quickly, so that species Protein and Protein-matured become identical. In this case, the system can be described by reactions (4.1) to (4.4), and the state of the system is fully characterized by the number of mRNA and Protein molecules, whose mean \( m \) and coefficient of variation \( \eta \) are

\[ <n_{\text{mRNA}} > \equiv m_{\text{mRNA}} = \frac{c_1}{c_2} \]  \hspace{1cm} (4.24)

\[ \eta_{\text{mRNA}} = \sqrt{\frac{c_2}{c_1}} \]  \hspace{1cm} (4.25)

\[ <n_{\text{protein}} > \equiv m_{\text{protein}} = \frac{c_1 c_3}{c_2 c_4} \]  \hspace{1cm} (4.26)

and

\[ \eta_{\text{protein}} = \sqrt{\left( \frac{1}{m_{\text{protein}}} \right) \left( 1 + \frac{c_3}{c_2 + c_4} \right)}. \]  \hspace{1cm} (4.27)

As expected, the mean number of proteins expressed varies directly with the transcription \( (c_1) \) and translation \( (c_3) \) stochastic reaction constants, and varies inversely with the mRNA degradation \( (c_2) \) and protein degradation \( (c_4) \) constants.

Analytic results also exist for models with finite maturation steps (non instant maturation), including the subsequent species in the chain of maturation steps: Protein-folded, Protein-oxidized, and Protein-matured. However, the number of terms in the expressions grows rapidly for these species, and they are too cumbersome to reproduce here. The full solutions were obtained using the Maple symbolic manipulation package (Wolfram Research, Champaign IL). To verify the analytical results, we have carried out Monte Carlo realizations of the stochastic
process corresponding to the set of biochemical reactions [52], using BioNetS [51], a piece of software specialized for that purpose.

We are mainly interested in the mean and variability of the fully matured protein species, Protein-matured, since this is the form whose fluorescence is observable in biological experiments. We also wish to examine the impact of neglecting the maturation effects on estimates of the variability of the protein species, and thus for each case we compare the full model with the instant-maturation model described above. In the following, we vary three sets of stochastic reaction constants: transcription rate \( c_1 \); translation rate \( c_3 \); and the maturation rate (the reaction constants for the folding, cyclization, and oxidation processes, \( c_5 \), \( c_6 \), and \( c_7 \)). To alter the total maturation rate, all three maturation constants are varied by the same factor, and as noted above, the maturation time is given as the average time to mature a fraction \((1 - 1/e)\) of the protein.

Fig. 4.3 shows the average number of matured proteins as a function of maturation time (set by \( c_5 \), \( c_6 \), and \( c_7 \)) and translation rate \( (c_3) \), keeping the transcription rate \( (c_1) \) fixed. As expected, the average number of mature (and thus experimentally observable) proteins decreases with maturation time and increases with translation rate. Fig. 4.4 shows the coefficient of variation of the matured protein species, again as a function of translation rate \( (c_3) \) and with varying maturation times. The coefficient of variation decreases as the translation rate increases, and over much of the range the observed variability \( \eta_{\text{protein-matured}} \) is increased by longer maturation times. The increase in coefficient of variation from the instant-maturation to the finite-maturation cases varies strongly with maturation time: at the lowest translation rate shown in Fig. 4.3, the difference is 8% for a maturation time of 7 minutes; 160% for a 107 minute maturation time; a factor of 9 times for a seven-hour maturation; and approximately a factor of 50 times for a 28-hour maturation time. Note that the difference becomes less pronounced as the translation rate increases, until in the high translation rate regime the behavior is reversed and proteins that mature quickly (or instantly) show less variability than those that mature slowly. The trends seen in Fig. 4.4 are the result of a combination of two effects induced by maturation steps: the smaller number of matured proteins than the total number produced, leading to an increase in observed variability; and a noise filtering effect arising from the maturation steps, leading to a decrease in observed variability (described in section B, below). Which effect dominates depends on the regime we consider.

To see the effect of changing the total number of matured proteins on the coefficient of variation at a given maturation time, we vary the transcription or translation constants, equivalent to changing the promoter strength or the translational efficiency (e.g. by altering ribosome binding sites), respectively. Fig. 4.5 plots the coefficient of variation against mean number of matured proteins when the translation constant \( c_3 \) is fixed, and we vary the transcription constant \( c_1 \),
Figure 4.3: Mean number of matured proteins versus protein maturation time. The translation rate (constant $c_3$) is varied as shown, for a fixed transcription rate $c_1 = 0.46$ s$^{-1}$. The fastest maturation time has $c_5 = c_6 = 4$ s$^{-1}$, $c_7 = 4^{-1}$ s$^{-1}$, and longer maturation times (slower maturation rates) are achieved by successively reducing all three constants by the same factor. The lines represent analytical results from solving the master equations, while the points represent computational simulations run for approximately 5 billion reaction update steps.
Figure 4.4: Coefficient of variation ($\eta$) of the fully matured protein species versus translation rate ($c_{3}$), for varying fluorescent protein maturation times, at a fixed transcription rate $c_1 = 0.46 \text{ s}^{-1}$.

The lines represent analytical solutions from the master equations, and the points are from computational simulations run for approximately 5 billion reaction update steps. The “Instant maturation” case shows the result when the proteins are taken to fluoresce instantly upon being produced. Note the high-rate regime wherein we see a qualitative change in the behavior, and the instantly-maturing protein is more variable than the versions with finite maturation times.

Stochastic reaction constants for the maturation processes are of the form $c_5 = c_6 = 4^{2-k} \text{ s}^{-1}$, $c_7 = 4^{1-k} \text{ s}^{-1}$, with $k$ taking on the following values: 7 min maturation time, $k = 5$; 107 min, $k = 7$; 7 h, $k = 8$; and 28 h, $k = 9$. 
Figure 4.5: Coefficient of variation ($\eta$) of the fully matured protein species versus mean protein number, for varying fluorescent protein maturation times, at a fixed translation rate. Mean protein numbers are varied by altering the transcription rate ($c_1$), keeping the translation rate fixed at $c_3 = 4.8 \text{ s}^{-1}$. The dashed line shows $\eta = 1/\sqrt{m_{\text{protein-matured}}}$. Reaction constants for the varying maturation times are as given in Fig. 4.4.
Figure 4.6: Coefficient of variation ($\eta$) of the fully matured protein species versus mean protein number, for varying fluorescent protein maturation times, at a fixed transcription rate. For each protein maturation time, the translation rate ($c_3$) is varied while keeping the transcription rate fixed at $c_1 = 0.46 \text{ s}^{-1}$. The dashed line shows $\eta = 1 / \sqrt{m_{\text{protein-matured}}}$. Note that in the high-number regime, the coefficient of variation deviates from $\eta = 1 / \sqrt{m_{\text{protein-matured}}}$, and the instantly-maturing version shows less variability than the finite maturation time versions. Reaction constants for the varying maturation times are as given in Fig. 4.4.
for several different maturation times. Fig. 4.6 shows the coefficient of variation as a function of mean number of matured proteins present, when we fix the transcription constant $c_1$ and vary the translation constant $c_3$, for the same set of maturation times.

The behavior of the curve of the “instant maturation” case in Figs 4.5 and 4.6 may be understood by considering Eqs (4.26) and (4.27), giving the mean and coefficient of variation for the instant-maturation case. Taking logarithms and combining Eqs (4.26) and (4.27), we find that a log-log plot of coefficient of variation against mean protein number, in the instant maturation case, has the form

$$\log \eta_{\text{protein}} = -\frac{1}{2} \log m_{\text{protein}} + f(c_2, c_3, c_4)$$

(4.28)

where $f(c_2, c_3, c_4) = \frac{1}{2} \log [1 + c_3/(c_2 + c_4)]$. Thus our log-log plot of the coefficient of variation for the instantly maturing protein versus its mean has a constant slope of negative one-half, offset by a constant depending only on the translation rate ($c_3$) and the mRNA and protein degradation rates ($c_2$ and $c_4$); note that this slope is the same as that from the classical result that the coefficient of variation varies inversely with the square root of the mean for a Poisson process [93].

In Fig. 4.5, Eq. (4.28) yields a perfectly straight line on the plot, since we are varying only the transcription rate $c_1$ and thus the offset $f(c_2, c_3, c_4)$ is fixed across the plot. The analytic solution for the finite maturation time case does not reduce to a simple form like that of Eq. (4.27), but as Fig. 4.5 illustrates, the effect of the additional maturation steps in the large translation rate case shown is to reduce the variability of slow-maturing proteins compared to fast-maturing ones at all different transcription rates, with the lower limit being $\eta = 1/\sqrt{m_{\text{protein-matured}}}$ (appearing as a dashed line in Figs 4.5 and 4.6).

In Fig. 4.6, we are varying the translation rate $c_3$, and thus the offset $f(c_2, c_3, c_4)$ changes across the plot. For small translation rates, corresponding to low mean numbers of matured proteins, $f(c_2, c_3, c_4) \to \log 1 = 0$, and thus the coefficient of variation approaches the $\eta = 1/\sqrt{m_{\text{protein-matured}}}$ limit (shown as a dashed line on the plot). To address the behavior at large translation rates (and thus large mean protein numbers), we note that Eq. (4.27) may be rearranged into the form shown in Eq. (4.29), as has been previously derived [99]:

$$\eta_{\text{protein}} = \sqrt{\frac{1}{m_{\text{protein}}} + \eta_{\text{mRNA}}^2 \frac{c_4}{c_2 + c_4}}.$$  

(4.29)

As $c_3$ increases in Eq. (4.29), the $1/m_{\text{protein}}$ term vanishes, yielding $\eta_{\text{protein}} \approx \eta_{\text{mRNA}} \sqrt{c_4/(c_2 + c_4)}$, a function only of $c_1$, $c_2$, and $c_4$: for high translation rates, the variability of the protein is dominated by that of the mRNA. Since we are only varying $c_3$ in Fig. 4.6, this term will be a constant, yielding in the flat line seen in the high number regime for the instant-maturation case. The
analytic solution for the finite maturation case once again does not have a compact form, but as seen in Fig. 4.6, the filtering effect of the maturation steps is observed only in the high-number (high translation rate) regime, where the maturation process reduces the coefficient of variation of proteins with slow maturation compared to those with fast or instant maturation. The difference in behavior in the high-number regime in this case arises from a low-pass filtering effect created by the maturation steps, as discussed in section B, below.

The results in this section clearly indicate that neglecting protein maturation effects will lead to significant underestimates of the number of proteins of interest being produced in the cell (Fig. 4.3), since the number of observable (matured and fluorescing) proteins decreases as the maturation time increases. The observed fluorescence also provides an inaccurate measure of the variability in the total production of the target protein (Figs 4.5 and 4.6). When the translation rate is small, the small-number effects dominate and the increased variability introduced by the process of maturation closely matches that expected from the reduced mean numbers of matured proteins present, whereas at higher translation rates the trend reversed, with maturation acting to reduce the observed variability. This latter effect arises from a low-pass filtering introduced by the maturation steps, as we will show by considering noise power spectra, below.

4.2.2 Noise power spectra

To further investigate how the fluctuations in species numbers are affected by the addition of the maturation steps, we calculate noise power spectra using a method developed by Warren et al. [100]. The noise power spectrum is the Fourier transform of the autocorrelation function, and is defined as

\[ P_i(\omega) = 2 \times \int_0^{\infty} \cos(\omega t)R_{ii}(t). \]  

(4.30)

The correlation function is defined as

\[ R_{ii}(t) = \langle \Delta n_i(0)\Delta n_i(t) \rangle \]  

(4.31)

where \( \Delta n_i(t) \equiv n_i(t) - E[n_i(t)] \).

The noise power spectra follow the sum rule,

\[ \pi^{-1} \int_0^{\infty} P_i(\omega)d\omega = \sigma_i^2 \]  

(4.32)

where \( \sigma_i^2 \) is the variance of the \( i \)-th species. In the following analysis, we redefine the noise power spectrum to be \( P_i(\omega)/<n_i>^2 \), so that the integral of the noise power spectrum will simply equal \( \pi \eta_i^2 \), a convenient form given our use of the coefficient of variation as an indicator of the degree of variability in a species.
Figure 4.7: Noise power spectra for different species in the model, calculated as described in the text, at transcription rate $c_1 = 0.46 \text{ s}^{-1}$, translation rate $c_3 = 4.8 \text{ s}^{-1}$, and a maturation time of 107 min ($c_5 = c_6 = 4^{-5} \text{ s}^{-1}$, $c_7 = 4^{-6} \text{ s}^{-1}$). Parameter settings have been chosen to yield lower noise in the matured protein species than in the instant-maturation case (not shown). The curves shown are analytic solutions, with points added only to distinguish between the lines.

The equations for the noise power spectra can be obtained by taking the Laplace transform of a set of differential equations giving the time evolution of the correlation coefficients [100]. Solving those equations in Maple, we get the analytical results for the noise power spectra; again, the detailed results are too unwieldy to reproduce here.

Fig. 4.7 shows the noise power spectra for different species for a parameter set where the noise is suppressed by the maturation steps; that is, where the coefficient of variation of the matured species is lower for finite maturation times than when the maturation is taken to happen instantly (see the high-number regime in Fig. 4.6 and the whole range of Fig. 4.5). The noise power spectrum for a given species is obtained by neglecting all reactions (except degradation) downstream of the species of interest.
Figure 4.8: Noise power spectra for different species in the model, at transcription rate $c_1 = 0.46 \text{ s}^{-1}$, translation rate $c_3 = 0.0047 \text{ s}^{-1}$, and maturation time 107 min ($c_5 = c_6 = 4^{-5} \text{ s}^{-1}$, $c_7 = 4^{-6} \text{ s}^{-1}$). Parameter settings have been chosen to yield higher noise in the matured protein species than in the instant-maturation case (not shown). The curves shown are analytic solutions, with points added only to distinguish between the lines.
Figure 4.9: Noise power spectra for species Protein-cyclized, Protein-matured, and Protein-matured minus the intrinsic noise power spectrum of the conversion reaction, which gives the level of upstream extrinsic noise power (from Protein-cyclized) still present in species Protein-matured, at transcription rate $c_1 = 0.46 \text{ s}^{-1}$, translation rate $c_3 = 4.8 \text{ s}^{-1}$, and maturation time 107 min ($c_5 = c_6 = 4^{-5} \text{ s}^{-1}$, $c_7 = 4^{-6} \text{ s}^{-1}$). The curves shown are analytic solutions, with points added only to distinguish between the lines.
Eqs (4.25) and (4.27) demonstrate that for certain parameter values (when \( c_3 > c_4(1 + c_4/c_2) \)), the coefficient of variation of the protein can be lower than that of the mRNA. This may also be seen in the noise power spectra shown in Fig. 4.7. Except in the very low frequency range, the noise power of the Protein species is smaller than that of the mRNA species. The fluctuations in the mRNA level are partly suppressed by the translation reaction, which acts like a low-pass filter.

Fig. 4.7 also shows the noise power spectra for the species Protein-folded, Protein-cyclized, and Protein-matured. Each of the three maturation steps decreases the noise in the middle frequency range, increases the noise at high frequencies, and very slightly increases the noise at the low frequency end of the spectrum. Samoilov et al. [101] conclude using deterministic analysis that linear feedforward networks of chemical reactions act as low-pass filters, while Warren et al. [100] examine stochastic reaction behavior, noting that such reactions combine low-pass filtering with an injection of noise from their own noisy reaction rates, creating a noisy low-pass filter. In Fig. 4.7 the model has been placed in a regime wherein the low-pass filtering effects dominate, and the coefficient of variation of the final matured protein species is lower than in the instant-maturation case. Fig. 4.8 shows a different parameter set, where the coefficient of variation of the matured protein is higher than for the instant-maturation case. Here, the low-pass filtering behavior is still visible as a reduction in noise power in the higher frequency ranges, but this reduction in noise power is dominated by an increase in noise power in the low-frequency range, leading to an overall increase in variability as a result of the maturation steps.

Consider just the final step in the maturation process, the oxidation of Protein-cyclized that yields Protein-matured. The noise in Protein-matured is affected by three factors: the intrinsic noise introduced by fluctuations in the rate of the oxidation reaction itself; the extrinsic noise from the previous step in the process (that is, the noise in Protein-cyclized); and the suppression of the extrinsic noise by the action of the oxidation reaction. The presence of the oxidation step has two effects: it acts as a low-pass filter on the noise from upstream (that is, the noise in the species Protein-cyclized); and it adds the noise from its own fluctuations [100]. The intrinsic noise introduced by the oxidation step may be calculated by fixing the number of Protein-cyclized at its mean value, and generating the noise power spectrum for species Protein-matured under these conditions. To examine the filtering effect, we subtract this intrinsic noise spectrum from the full noise power spectrum of Protein-matured, and denote the remaining noise power spectrum as the “extracted extrinsic noise” of the matured species. This noise spectrum, along with the original spectra for the cyclized and matured species, is plotted in Fig. 4.9; note the large degree of suppression of the higher frequencies evident in the extracted extrinsic noise. We then compute the “passing ratio” for the process by dividing the extracted
Figure 4.10: Passing ratio of the upstream extrinsic noise through the final maturation reaction step (oxidation), for different maturation times, at transcription rate $c_1 = 0.46 \text{ s}^{-1}$ and translation rate $c_3 = 4.8 \text{ s}^{-1}$. The curves shown are analytic solutions, with points added only to distinguish between the lines.

Extrinsic noise in Protein-matured by the total noise present in Protein-cyclized. The result is plotted in Fig. 4.10, which clearly shows the noisy low-pass filter behavior of this single-step maturation process. Increasing the maturation time has the effect of increasing the range of frequencies over which the low-pass filtering effect operates, and thus for equal mean numbers of matured protein, longer maturation times will yield lower observed coefficient of variation, as seen in Fig. 4.6. Each step in the full maturation process has the same structure as the final step considered here, so the noise suppression is repeated and augmented through the folding, cyclization, and oxidation steps leading to the fully matured protein.

Fig. 4.11 shows the noise power spectra for different maturation times at a fixed transcription and translation rate. At the same mean level of protein expression, longer maturation times lead to greater noise suppression (see Fig. 4.6), but with fixed transcription and translation
Figure 4.11: Noise power spectrum for different maturation times while transcription rate and translation rate are kept constant, at transcription rate $c_1 = 0.46 \, s^{-1}$ and translation rate $c_3 = 4.8 \, s^{-1}$. The curves shown are analytic solutions, with points added only to distinguish between the lines.

rates, longer maturation times yield lower numbers of mature proteins. Fig. 4.11 thus illustrates the competition between the injection of extra noise from the additional reaction steps and the low-pass filtering effect of these reactions: at the parameters shown, for short maturation times the filtering effect dominates, reducing the noise power by filtering away higher-frequency components; at longer maturation times, this effect is swamped by the injection of fluctuations caused by lowering the molecule numbers, and the 27-hour maturation time yields higher noise power at all frequencies, with a corresponding increase in the observed coefficient of variation.

### 4.3 Conclusion

We have examined a model of the production of fluorescent reporter proteins incorporating significant simplifications: we have omitted a number of intermediate steps in gene expression;
neglected nonlinear effects arising from processes such as regulatory feedback; and we assumed that the fluorescent proteins are degraded mainly through dilution by cell growth rather than rapid proteolysis or other active processes (often the case, but not universally so). The parameter settings have been selected to reflect typical values for *Escherichia coli* and thus may not be appropriate for all organisms.

Keeping these caveats in mind, we may use the model to extract information about the effects of protein maturation on what is seen experimentally when using fluorescent proteins as reporters. As we have seen, the observed level and variability of gene expression can be significantly affected by the maturation steps that must be completed before a fluorescent protein can generate fluorescence output. Interestingly, the effect is not uniformly in the direction of increasing the observed variability: in some regimes, the observed variability may in fact decrease compared to the actual variability of the total protein production. The maturation process has two effects: it introduces additional intrinsic noise from fluctuations in the rates of the maturation reactions; and it acts as a low-pass filter, removing some of the noise from earlier steps in the gene expression process. Longer maturation times lead to a smaller pass band in the low-pass filter, which increases the frequency range of upstream noise filtered out; at the same time, longer maturation times introduce larger intrinsic noise. Combining these two effects for a given translation rate, there is a range of maturation times where the noise in the output (matured) protein is suppressed rather than enhanced, with the suppression manifesting itself as a reduction in noise power in the intermediate frequency ranges. The low-pass filtering acts to smooth out bursts of protein production from the translation process. Thus the noise-reduction effect occurs most strongly at large translation rates, which will have more high-frequency components in the immature protein numbers, and at longer maturation times, which will have smaller passing bands for the low-pass filter effect. In general, we observe that the larger the translation rate, the larger will be the set of maturation times able to suppress the variability in the matured protein; and the longer the maturation time, the stronger the noise-suppression effect will be. Though we have analyzed the case of maturation of a fluorescent protein, the model can represent any linear chain of transformation reactions, and thus our results might be of use in other cases as well.

Based on the model, the process of fluorescent protein maturation affects the observed number of proteins present, when fluorescent reporters are used as reporters in living cells; as expected, the longer the maturation time, the fewer matured proteins will be visible relative to the total protein population. It is perhaps less intuitively clear that this maturation process also affects the observed variability in gene expression, and that the effect can either increase or decrease the observable coefficients of variation relative to the true variability of the protein population. In either case, it is clear that the maturation process can potentially introduce a
significant perturbation on the observed data, and should be taken into account in quantitative
studies of gene expression. Our model suggests a method of correcting experimental data to
yield more accurate measurements of levels of protein expression and variability, and future
work could extend these results to more complex systems.
Chapter 5

Time dependent gene expression dynamics in *E. coli* Batch culture

Experiments are carried out with Sangram Bagh. We will share the first authorship on the publication (in preparation).

5.1 Introduction

Both systems biology and synthetic biology require the quantitative characterization of the gene expression dynamics and noise of a given plasmid. Currently, at large, the behavior of gene expression systems was analyzed at different time points, 3 hours [30, 102], or 5 hours early in the log phase growth or different OD points [28, 102, 103]. One of the reasons for analyzing the gene expression data in the exponential phase of growth in a batch culture is the assumption that gene expression is in the steady state. Is there any “true” steady state in gene expression dynamics in a batch culture? How do the gene expression level and noise vary with time? Is the time dependent behavior the same for all regulated and unregulated systems? How do the time dependent gene expression dynamics depend on biochemical parameters? We address those questions in our study experimentally and computationally. In this study we construct different unregulated and regulated synthetic gene systems in high and low copy plasmids and insert in *E coli*. Gene expression is studied as a function of time by means of fluorescence of our reporter enhanced green fluorescence protein (EGFP) detected by flow cytometry. We show in this study that both the gene expression level and gene expression variability among a genetically identical population of cells vary strongly with time. This time dependent variation is different for different types of unregulated and regulated gene systems. This difference in behavior is a direct manifestation of biochemical parameters associated with the gene systems. As a result we found there is no global steady state in a batch culture.
5.2 Method

5.2.1 Plasmid construction

In this work, both gene expression from regulated and unregulated promoter systems are studied. EGFP are expressed from plasmids in \textit{E. coli} cells, and used to monitor the gene expression level and noise. All plasmids used in this study are constructed using standard molecular biology techniques \cite{Reference104}. The genes and promoters are amplified by PCR using Pfu Turbo hot start master mix in a thermocycler (MJ research, South San Francisco, USA) using manufacturers protocol. Plasmid DNA is purified using QIAPrep Spin Miniprep kits (Qiagen Canada, Mississauga, ON), sequenced (The Centre for Applied Genomics, Toronto, ON), and transformed by electroporation (ECM 399, BTX, San Diego, CA) into \textit{E. coli} strain DH5a (Invitrogen Canada, Burlington, ON).

5.2.2 Cell culture

Cells from single colonies are cultured for 16 hrs in Luria-Bertani (LB) medium (BioShop Canada, Burlington, ON) plus 100 g/ml ampicillin (BioShop) at 37\(^o\)C at 300 RPM. Overnight cultures are diluted 100 times in fresh medium to a final volume of 400 mL and grown under the same conditions.

5.2.3 Flow cytometry

Time series flow cytometry is performed using an Epics Altra Cell sorter (Beckman Coulter, Mississauga, ON, Canada). Cells are excited at 488 nm at 300 mW with an Innova 70 laser (Coherent, Santa Clara, CA). The cell growth is monitored simultaneously with a spectrophotometer by measuring optical density (OD) at 600nm OD600 (Ultrospec 100 pro spectrophotometer, Biochrom, Cambridge, England). Every 15-20 minutes, 2 mL of cell culture is taken out for measuring the OD600, diluted 10-500 times, and run through flow cytometry.

5.2.4 Microscopy

Fluorescence level is also measured using a TE2000U microscope (Nikon Canada, Mississauga, ON), to confirm the results from flow cytometry. The microscope is equipped with a Piston GFP filter cube (HQ470/40x, Q495LP, HQ515/30m; Chroma Technology, Rockingham, VT). Images are collected with a charge-coupled device camera (Cascade 650, Photometrics, Tucson, AZ) with fixed on-chip multiplier gain, with a 100 ms integration time to reduce photobleaching. Image analysis is performed with METAMORPH v6.2r6 (Molecular Devices, Sunnyvale, CA). Cell sizes are determined using a fixed intensity threshold to define cell boundaries, and cells are
distinguished from debris using METAMORPH image analysis tools to define regions of total size, shape factor, and total and average fluorescence intensity that constituted probable cells rather than noncells. In each case, the software-identified cells are double checked manually; phase contrast images of the cells are used to gain a sense of the range of cell sizes present in the population, and this is used to allow the human operator to make final decisions about including or excluding borderline objects.

5.2.5 Protein number quantification

Fluorescent proteins allow the measurement of a signal whose intensity is proportional to the number of proteins being expressed in an individual cell. Here we use a straightforward fluorescence intensity calibration method to obtain the average number of EGFP expressed per cell. The method is described elsewhere [30]. In brief, we generate a standard calibration curve between EGFP numbers and fluorescence counts in fluorimetry (Quanta-Master; Photon Technology, Birmingham, NJ) using standard EGFP solutions. The protein number of EGFP expressing pTLEGFP in DH5α cell in 3 hr culture (diluted from overnight, then grown) is determined by comparing to the standard calibration curve. Now taking these cells as standard (the protein number is constant for the 3 hr culture found in different independent experiments throughout the year), the numbers of EGFP for other cells are approximated comparing the fluorescence at flow cytometer.

5.2.6 Data analysis

All data is analyzed in MATLAB. The raw data from flow cytometry is thresholded in two steps. First, background from running sheath fluid is analyzed to define the lower limit of the forward and side scatter outputs of cells, and control samples of cells with no plasmids are analyzed to define the lower limit of fluorescence output for cells with plasmids. Second, the cells falling out of the region of 5 standard deviation along forward scatter, and side scatter are excluded to cut off the long tails in histograms: this region likely includes only dead cells or instrument noise.

5.3 Result

The descriptions of the plasmids used in this work are listed in Table 5.1. All those plasmids are inserted in DH5α cells either individually (pLAEGFP, pZ12EGFP, pCIts-EGFP) or in combination (pZ12EGFP+pLac). The cells carrying plasmids pLAEGFP and pZ12EGFP have unregulated EGFP expression whereas others have regulated expression. CIts-EGFP is an auto-regulated system where at low concentration of CIts, the $P_{RM}$ promoter works as positive regulator, and as concentration of CIts increases, it starts behaving as a negative regulator. As
Table 5.1: Description of plasmids used in this project.

<table>
<thead>
<tr>
<th>Plasmids</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pLAEGFP</td>
<td>EGFP gene is expressed from synthetic promoter pLlacAra1 in an unregulated system, in a plasmid backbone with PMB ori (plasmid copy number: 50-80) and Amp resistance marker gene in PBR322 backbone</td>
</tr>
<tr>
<td>pZ12EGFP</td>
<td>EGFP gene is expressed from synthetic promoter pLlacO-1 in an unregulated systems, in a plasmid backbone with ColE1 type ori (plasmid copy number: 50-70) and Amp resistance marker gene in pZE12-luc backbone</td>
</tr>
<tr>
<td>pLac</td>
<td>Lac I (repressor) protein is expressed from pLtetO-1 promoter. in a plasmid backbone with ColE1 type ori and Cm resistance marker gene</td>
</tr>
<tr>
<td>pC1ts-EGFP</td>
<td>Temperature sensitive mutant of lambda repressor gene (CIts) is expressed from lambda PRM promoter with all three operating sites (OR1, OR2, OR3), EGFP is attached bi-cistronically with CIts, in a plasmid backbone with ColE1 type ori (plasmid copy number: 50-70) and Amp resistance marker gene.</td>
</tr>
</tbody>
</table>

the CIts in connected downstream of the $P_{RM}$ promoter, it autoregulates its own production. On the other hand in cells with dual plasmids (pZ12EGFP+pLac), the LacI expressed from pLac represses the production of EGFP from pLlacO-1.

After cells containing different plasmids are grown over night for 16 hours, the overnight culture is transferred into a flask with new fresh medium, grown under the same conditions. Every 15 to 60 minutes, 2 ml of the cell culture is taken out, measured using a spectrophotometer, and run through flow cytometry, where the forward scattering, side scattering and fluorescence signal from individual cells are recorded. The forward scattering (FS) signal is thought to be proportional to the cell size, and the fluorescent intensity is proportional to the number of proteins. The absolute protein number is obtained by converting the fluorescence signal to protein number using the calibration method described above. Microscopy is also used to measure the time evolution of cells containing the pLac plasmid: the result from microscopy shows good match with the flow cytometry result. For each plasmid setup, three colonies are grown and analyzed separately. The statistics of the measured data are calculated in MATLAB and plotted together with the growth curve in Figure 5.1.

Comparing the results for different plasmids, we can conclude that they all show qualitatively similar behavior in terms of OD, cell size, protein number and CV. The OD curves of all those systems follow three stages: lag phase, exponential growth phase, and stationary phase. In the first stage, cells are adapting to the new environment and grow very slowly. During the
Figure 5.1: Results for 4 different plasmid systems. For each system, we show the time series of the mean protein number, the coefficient of variation of protein, the mean cell size and optical density from three different runs, denoted by different points on the graph.
Figure 5.2: Cell growth at different growth rates and division rates. For each growth rate and division rate combination, we show the cell size time series for 20 hours, the average cell size, plotted as the horizontal line in center, and the cell division size, plotted as the horizontal line on top.

second stage, cells have adapted to the new environment and start to grow with the maximum speed. In the third stage, the nutrition inside the medium are almost used up, cells are forced to grow slower. The exponential phases in our cells range from about 2 hours to around 7 hours. Although similar, the OD curves for different systems are not exactly the same, which is thought to be caused by the effect of plasmids on the cell growth [30].

The cell sizes for different plasmid systems also show similar pattern: the cell sizes become larger from 0 to 2 hours, become smaller from 2 to 6 or 8 hours, and then stay approximately constant. The average cell size is determined by two factors, the cell growth rate, and the cell division rate, as shown in Figure 5.2. The faster the cells grow, the bigger the cell size will be under the same cell division rate, while the more frequently the cells divide, the smaller the cell size will be. The division size, the size when the cell divides, and the average cell size at certain
time will both be determined by the growth rate and division frequency at that time. If the
decrease in the cell growth rate is smaller compared to the decrease of the cell division rate,
the average size when the cell will divide will become larger and the average cell size will also
increase, which is what happened in the first two hours in the cell culture. After 2 hour, the cell
starts to grow exponentially, the growth speed increases; however, the cell division frequency
increases even more, resulting in a smaller average cell size. After 4 or 6 hours, the cell division
rate and the cell growth rate reach some equilibrium with each other, maintaining a nearly
constant cell size.

For all the different cases, the mean number of proteins increases with time for the first 2
hours after the cells are diluted in new fresh medium from over-night culture, decreases with
time from 2 to 4 hours period, and increases again after 4 hours. There is no steady state during
our 12 hours’ observation - the difference between the maximum fluorescence level at 2 hours
and the minimum fluorescence level at around 4 hours can be as large as 3 times.

The similarities in the protein number curves imply that the shape of the time dependent
curves are probably mainly the result of some external factor other than from the gene expression
itself, which will affect the EGFP expression from different plasmids in a similar way. One of the
extrinsic factors should be the change of cell division frequency over time. Let us first consider
the case when we have the EGFP expressed at a constant rate over time. In the first stage from
0 to about 2 hours, the cells seldom divide, while the fluorescent proteins are still expressed as a
constant speed, accumulating more and more proteins inside each cell. After 2 hours, the cells
divide more frequently, making the average number of protein decreasing. After 4 hours, the
cell division is slowed down again, resulting in an increase in protein number. A quantitative
explanation of this behavior is shown in the modelling section.

The coefficient of variation of proteins also varied with time, and the time-dependent behav-
or is somewhat different for different runs. Generally, the noise increases for the first several
hours and then decreases. There is no steady state in the coefficient of variation curve, and
the difference at different time points can be more than 30%. There is no steady state during
exponential growth phase for all those different plasmids in terms of the mean number of protein
and the noise in protein numbers. When making a measurement, we will have to specify the
time point when we are interested in, and the measurements at different time points should not
be compared directly.

Since the time dependent behavior of different systems are divergent, if we compare the
fluorescence level and noise at different time points, even at different time points during the
exponential phase, we will get significantly divergent results. Take the coefficient of variation
of the pZ12EGFP run and the pZ12EGFP+pLac run as an example as shown in Figure 5.3: If
we compare the CV at 3 hours as is often done [30,102], we will conclude that the CV from the
Figure 5.3: Comparison of CV of regulated and unregulated systems.
5.4 Modelling

5.4.1 Modelling by tracking all cells

When modelling the cells in E. coli batch culture, after the cell divides, usually only one of the sibling cells is tracked, while in reality, both the sibling cells exist in the cell culture. Here we compare the model that tracks only one sibling cell with the model that tracks both sibling cells. The two models are run with 1000 initial cells sampled randomly from the initial cell size and protein number distribution measured experimentally. In the tracking-one-cell model, we will have only 1000 cells over the 12 hour simulation, while in the tracking-all-cell model the number
Chapter 5. Time series gene expression dynamics

Figure 5.6: Time series of protein number from model that tracks all cells and model that only tracks one sibling cell. In each case, the simulations are run from 1000 initial cells and, the plotted curve shows the average cell size from all simulations.

of cells increase exponentially (about 110000 cells after the 12 hour simulation). The results from the simulations are plotted in figure 5.6 and 5.7, which shows that the tracking-one-cell model will give very similar result compared to the tracking-all-cell model. Since the tracking-all-cell model takes much less computer power to simulate, we only use the tracking-one-cell model in the following work.

5.4.2 Modelling by tracking only one random daughter cell

To understand the time dependent gene expression dynamics measured experimentally, we formulate a simple gene expression model as shown in Eq. 5.1 and Eq. 5.2, and use this model to simulate the time dependent behavior of the system. In the model, each cell is represented by two parameters, the cell volume and the number of EGFP inside the cell. Only three cellular
Figure 5.7: Time series of protein number from model that tracks all cells and model that only tracks one sibling cell. In each case, the simulations are run from 1000 initial cells and, the plotted curve shows the average protein number from all simulations.
processes are considered in the model: cell growth, cell division and the expression of EGFP.

\[
v \xrightarrow{k_1} 2v \\
\emptyset \xrightarrow{k_2} EGFP
\]  

(5.1)  
(5.2)

The first and second reaction represent the cell growth and gene expression respectively, while the cell division is simulated by dividing the cell volume and protein number using binomial distribution when the cell size reaches a certain limit [51].

Eq. 5.1 describes the growth of individual cells. If the growth rate of each cell is assumed to be the same as the growth rate of the biomass of the whole culture - \( m'(t)/m(t) = v'(t)/v(t) \), where \( m(t) \) represent the biomass of the whole culture at time \( t \), the growth of the biomass of the whole culture can also be described by reaction 5.1. If the cell growth rate \( k_1 \) in this model is kept constant over time, the biomass of the whole culture will grow exponentially, which is obviously not true compared to the experimental OD curve, especially in the longer time end of the curve. To capture the pattern shown in the experimental curve, we will need to make the cell growth rate a time dependent variable. A function, shown below, is chosen to represent the time dependent cell growth rate [105], used to calculate the growth of biomass of the whole culture, and fitted to the experimental OD curve using the fminsearch approach in MATLAB. The fitting result is shown in Figure 5.8: the fitted function gives a good match to the experimental OD curve.

\[
k_1(t) = v'(t)/v(t) = m'(t)/m(t) = T e^{\gamma(1-m(t)/m_{max})} - 1 \frac{m^n(t)}{e^\gamma - 1 - m^n(t) + d^n}
\]

(5.3)

Where \( T, \gamma, m_{max}, \) and \( n \) are constants, calculated by fitting the model to the experimental OD curve.

With certain initial conditions, the average size of a population of cells is determined by the cell growth rate with the cell division frequency, or the cell division volume – the volume when the cell will divide. Under the same growth rate and initial conditions, the average cell size will becomes larger if the cell division volume become larger, or smaller if the cell division volume becomes smaller, or stay constant if the cell division volume is constant. As the experimental cell size is changing with time, the cell division volume in the model must also be time dependent to match the experimental cell size curve. The function used to represent the time dependent cell division volume is shown below:

\[
v_{max}(t) = k e^{\gamma(1-m(t)/m_{max})} - 1 \left( b + \frac{m^n(t)}{m^n(t) + c^n} \right) + v_0
\]

(5.4)
Figure 5.8: Fitting of cell growth using Eq. 5.3 to experimental OD curve.
Where the $v_{\text{max}}(t)$ represents the time dependent cell division volume, $m(t)$ stands for the biomass of the whole cell culture, and $k$, $\gamma$, $m_{\text{max}}$, $b$, $n$, $c$, and $v_0$ are constants, calculated by fitting the model to the experimental cell size curve.

To calculate the average cell size, the initial volume of 10000 in silico cells are chosen randomly from the experimental cell size distribution (FS signal). Allowing those 10000 in silico cells to grow independently and divide when the volume reaches $v_{\text{max}}(t)$, we obtain the time dependent average cell size by averaging over those 10000 cells. By changing the parameters in Eq. 5.4 and fitting the modelling result to the experimental cell size curve, we get a good match between the model and the experiment, as shown in Figure 5.9.

Using the fitted time dependent cell growth rate and cell division volume, we calculate how the average protein number changes with time when the protein production rate is assumed to be constant. Again, the initial conditions of 10000 in silico cells are sampled randomly from the initial experimental cell size and protein number distribution. The result of the simulation is plotted in Figure 5.10. The modeled protein number time series shows similar qualitative
behavior compared to the experimental fluorescence time series. The average protein number increases in the first 2 hours, decreases from 2 to 4 hours, and continuously increases after 4 hours. This qualitative match implies that the rise and fall behavior of the protein number is partly due to the changing of the cell growth rate and the cell division volume with time. The quantitative difference between the modelling and experimental curves implies that there will be other factors not represented in the model that affect the gene expression rate over time, such as the depletion of the nutrition in the batch culture, which is expected to slow down the gene expression rate over time.

5.5 Conclusion

Through the discussion, we have shown that the gene expression level of many synthetic plasmid systems in *E. coli* batch culture changes with time for at least 12 hours. There is absolutely no
steady state in terms of the average number of protein, no matter whether the gene expression is regulated or not. Further, the average cell size and the coefficient of variation of the protein level are also time dependent. We will have to specify the time point when we make a measurement of the system. And we will not be able to compare the measurement of the system at different time directly.

The change of coefficient of variation for different plasmid systems does not follow the same pattern. If we want to compare the coefficient of variation for two different plasmids, we will have to specify the time point when we want to compare the two cases, because comparisons made at different time points will give different qualitative conclusions.

By using a quantitative model, we have shown that the time dependent behavior of the gene expression level is partly due to the changing of the cell growth rate and the cell division volume over time. However, the quantitative difference implies that there are other factors that affect the gene expression rate over time. One possible candidate of this factor is the nutrition depletion. Further study will be carried out to look into this question.
Chapter 6

Effect of feedback on the noise in gene expression

6.1 Introduction

Negative feedback, broadly present in natural systems, is usually thought to be able to stabilize a system and reduce the noise in that system, while positive feedback is often thought to make the system unstable and more noisy [21, 106]. However, when applying this rule to gene expression networks, exceptions have been found: the negative feedback can actually increase the noise under some situations. It has been shown that negative feedback will increase the noise in a simple gene expression model by solving the langevin equation of that model [107]. When modelling the noise in a detailed gene expression model, I have also found that the negative feedback actually increases the noise in the protein number in that system.

However, some experimental work has shown that negative feedback can suppress the noise in certain systems [106]. Why does the modelling give results contrary to experiments? One possible explanation is that the models we use to describe the gene expression networks do not take into consideration the extrinsic noise: only the intrinsic noise is captured in typical stochastic gene expression models. While the feedback increases the intrinsic noise, it might decrease the extrinsic noise in gene expression networks: if the decreasing of extrinsic noise due to feedback is more dominant than the increasing of intrinsic noise, the total noise in that system will be reduced. In living cells, it has been shown that the extrinsic noise is the dominant source of noise in many cases. In those systems, probably, the decreasing in the extrinsic noise is larger than the increasing in the intrinsic noise, making the total noise smaller.

To test this assumption, I have simulated models with and without feedback to compare the intrinsic and extrinsic noise in these models. The modelling result demonstrates that negative feedback will increase the intrinsic noise and decrease the extrinsic noise. To further test this
Chapter 6. Effect of feedback on the noise in gene expression

6.2 Modelling

6.2.1 Detailed gene expression model with feedback

The detailed gene expression models with and without feedback as shown in Section 2.3.2 are used as the feedback and non-feedback models. The only difference between these two models is this reaction: $protein\_small + operon\_ns \leftrightarrow protein\_small\_operon\_ns$: the binding of protein\_small to operon will make some of the operon unavailable for the transcription reaction and thus serve as negative feedback. The simulations are run using BioNetS and the results analyzed in MATLAB.

Figure 6.1 and 6.2 show the protein number time series, from one of the simulation runs, for the model with and without feedback. The mean protein number and the CV of protein for the model without feedback are $2.9 \times 10^5$ and 0.26 respectively, while for the model with feedback, the mean protein number and the CV of protein are $1.4 \times 10^5$ and 0.3 respectively. Although the mean protein number in the model with feedback is twice as much as in the model without feedback, which will result in a smaller CV if the numbers of protein follow a Poisson distribution, the model with feedback shows a much larger variation, noise or stochasticity, than the model without feedback. Note that the model only takes into consideration the intrinsic noise, while no extrinsic noise, such as the plasmid copy number variation, the cell division size variation, or the concentration variation of molecules like amino acids or expression-related enzymes, is taken into consideration. It is quite probable that the extrinsic noise is the dominant source of noise and the feedback is capable of reducing the extrinsic noise substantially and thus...
How does the feedback increase the intrinsic noise of the model? Examining the model in further detail, we find a possible explanation: In the model without feedback, the number of plasmid is kept at constant, while in the model with feedback, the proteins will bind to some of the plasmids, making some of the plasmids unavailable for transcription, and since the number of proteins fluctuates with time, the number of available plasmids will also fluctuate with time, the noise in which will in turn be propagated into protein number, making the fluctuation in protein number much larger than the non-feedback, constant plasmid copy number case. This behavior is similar to the amplification of noise - the noise in the proteins is fed into its input, propagated through gene expression steps, and amplifies itself.

6.2.2 Simple gene expression model with feedback

To prove our assumption that negative feedback increases the intrinsic noise and at the same time decreases the extrinsic noise, we formulate a simple gene expression model as shown in figure 6.3. The gene expression process starts with the production of the plasmid with certain
Chapter 6. Effect of feedback on the noise in gene expression

Figure 6.4: Noise vs Feedback strength for different binding affinities.

rate constants, to which we can add extrinsic noise, and the degradation of the plasmid at a constant rate. Next, the plasmid will be transcribed, producing mRNA at a constant rate. The mRNA will then be translated into proteins. The protein itself will have the ability to bind to plasmids (the rate for binding is called "binding constant" in later description), forming a Plasmid-Protein complex. The transcription from this complex will be at a rate different from the original transcription rate with a coefficient $\alpha$ (feedback strength): if $\alpha$ is greater than one, the binding of protein to plasmid will facilitate the transcription, making it a positive feedback loop; if $\alpha$ is less than one, the binding of protein to plasmid will reduce the efficiency of transcription, thus making it a negative feedback loop.

To investigate what is the effect of negative and positive feedback on the intrinsic noise in gene expression, the model is simulated without the extrinsic noise added, using BioNetS [51]. Two different parameters, the binding constant and the feedback strength, are varied over wide ranges. The result of the simulation is plotted in Figure 6.4. When the noise strength is equal to one, or the binding constant is 0, there will be no feedback inside the model, which is shown as a single crossing point and the horizontal line in the plot. To the left side of this crossing point, the points represent negative feedback, and to the right side of this crossing point, the points represent positive feedback. The results show that the negative feedback will increase
the noise in the model, compared to the model without feedback, while the positive feedback will decrease the noise in the model. Comparing the curves for different binding constants, we can conclude that the stronger the binding, the more the noise is increased or decreased. Figure 6.5 plots some of the simulation results for models without feedback, with positive feedback and with negative feedback respectively. The number of proteins is varied by varying the translation rates. And the dashed line represents $y = 1/\sqrt{x}$. In this plot, when the protein number is large, the model with negative feedback shows larger noise than the model without feedback, which implies that the increasing of noise by negative feedback does not solely come from the smaller protein number resulting from negative feedback.

What would be the effect of the negative and positive feedback on the extrinsic variation?
Chapter 6. Effect of feedback on the noise in gene expression

Figure 6.6: Frequency response to extrinsic noise for models without feedback, with positive or with negative feedback.

To answer this question, I have investigated the frequency response of the positive, negative, and non-feedback model to the time variant plasmid production rate. The plasmid, instead of being produced at a constant rate, is produced at some form of time dependent rate, with the mean level the same as before and a sine function added to it. Varying the frequency of the sine function, we will be able to calculate the frequency response of those models to upstream variations. The result is plotted in Figure 6.6. All three models are able to suppress the variation in the higher frequency range, which is probably due to the slowness of biochemical reactions and inability to respond to high frequency changes from upstream. However, in the lower frequency range, the negative feedback suppresses the variation to a larger degree, compared to the model without feedback, while the model with positive feedback increases variation. This clearly shows how the negative feedback decreases the extrinsic variation and how the positive feedback increases the extrinsic variation, compared to the non-feedback case.

6.3 Experimental verification

To verify the modelling result, I have designed an experiment to measure the intrinsic and extrinsic noise of gene expression in living cells, which is being carried out by other members
Figure 6.7: Plasmid expressing EGFP and EYFP. The expression of EGFP and EYFP is controlled by two separate PLtetO-1 promoters on the opposite side of a plasmid. As the environmental factors in a single cell will affect the expression of EGFP and EYFP in the same way, the difference between the expression levels of EGFP and EYFP comes from the intrinsic noise of gene expression.

Figure 6.8: Plasmid expressing tetR-EGFP and tetR-EYFP. The tetR sequence are linked to EGFP or EYFP sequence on the opposite side of the plasmid. The expression of tetR-EGFP and tetR-EYFP is controlled by PLtetO-1 promoters, while tetR will bind to the promoter and prevent further transcription. By adding different amount of anhydrous tetracycline (aTc), which will bind to the tetR protein and prevent it from repressing the promoter, the amount of transcription can be fine tuned.
in the lab right now. When measuring the number of certain fluorescent protein in living cells directly, we will only be able to get the statistics in the number of that protein, and will not be able to separate the extrinsic noise from the intrinsic noise. To overcome this difficulty, we will use reporters with two different colors, EGFP and EYFP in our case [24]. The two colored proteins will be inserted on the same plasmid and controlled by the same promoters on the opposite side of a plasmid. They will be produced at the same average rate independently. Any global noise factor, such as plasmid copy number variation, will affect the gene expression from both proteins in a similar way, while the difference between the expression level of this two proteins will be attributed to the intrinsic noise in the gene expression processes of these two proteins. Using this trick for plasmids with and without feedback, we will be able to compare the intrinsic noise and extrinsic noise separately from plasmids with and without feedback, and test our assumption that feedback will increase the intrinsic noise in gene circuits and decrease the extrinsic noise from outside. The details of the plasmids are shown in Figure 6.7 and Figure 6.8.

6.4 Conclusion

In gene expression networks, negative feedback does not always reduce the noise. The effect of negative feedback is two-fold: on one side, it will suppress the extrinsic noise; on the other side, it will increase the intrinsic noise in the model. The average effect will depend on which effect is dominant. But in general, since the extrinsic noise is the dominant source of noise, resulting in a larger reduction of the extrinsic noise than the increase of the intrinsic noise, the negative feedback will usually suppress the total noise, as shown by some experimental observations. After the experimental part is finished, we hope we can prove this theory experimentally.
Chapter 7

Macromolecular crowding effect on the dynamics in gene expression

7.1 Introduction

Cells are crowded with macromolecules (In *E. coli*, around 30% in volume [108]), which slows down the diffusion of molecules and makes a large fraction of the volume inaccessible to big molecules, named the volume exclusion effect, as shown in Figure 7.1 [90,109]. Suppose the environment is crowded with big molecules, if another big molecule is entering this volume, the volume in light gray color will not be accessible to the big molecule, while, at the same time, most of the total volume will still be available to small molecules. The effective volume a molecule can diffuse into in total volume is reduced.

When using the law of mass action to describe the reaction dynamics of chemical reactions, we have to assume the reaction volume is a well-mixed environment. Due to the slow diffusion and the volume exclusion effect in crowded environments, the environment crowded with big molecules is more likely not a well-mixed environment. Thus the assumption made when using the law of mass action will not be valid anymore. What would be the reaction dynamics in this kind of crowded environment?

Many computational studies, usually tracking the positions of molecules individually, have been done to look at the effect of crowding obstacles on the dynamics and even noise in biochemical reactions [91,110–112]. It is observed, by modelling, that the reaction rates in crowded environments dependent on time and initial conditions, while the noise is increased in the crowding environments compared to non crowded environments. In well-mixed environments, to describe Markov processes such as biochemical reactions, master equations can be used, with the transition probabilities represented by the reaction rate calculated using the law of mass action. Since the law of mass action is not valid in crowded environments, the master equations
approach we used before will not be able to represent the stochastic behavior of such systems faithfully.

Currently, most experimental work on macromolecular crowding is focused on the effect of macromolecular crowding on the diffusion of certain molecules or particles, or the effect of macromolecular crowding on the folding processes of proteins. Experimental results have shown that macromolecular crowding will slow down the diffusion [113], while other researchers have found the acceleration of protein maturation due to crowding [114]. However, no experiment has been done to observe the effect of macromolecular crowding on the dynamics of gene expression. Since living cells are usually crowded with big molecules, such information is valuable when trying to model the gene regulation network quantitatively and accurately.

To investigate and understand the effect of macromolecular crowding on gene expression dynamics, I observe the time series expression of EGFP under different crowding conditions, using in vitro kits, which enable us to control the crowding conditions explicitly.

### 7.2 Method

EasyXpress Protein Synthesis kits are obtained from QIAGEN (Qiagen, Mississauga, Canada). CoverWell incubation chamber gaskets (Invitrogen, Carlsbad, USA), 13mm in diameter and 2 mm deep, are used as the container for in vitro gene expression. Plasmids expressing EGFP
are designed particularly for expressing protein *in vitro*, and constructed using the EasyXpress Linear Template Kit (QIAGEN, Mississauga, Canada). 200ug of plasmid is extracted from *E. coli* cells containing the plasmid, using Plasmid Maxi kit (QIAGEN, Mississauga, Canada). To heat the sample to $37^\circ$ C, a thermal plate (Tokai Hit Co. Ltd, Japan) is placed on top of the incubation chamber. The fluorescence level of the sample inside the incubation chamber is tracked using the Nikon TE2000U microscope (Nikon Canada, Mississauga, Canada). Since the sample need to be measured using fluorescence microscope for many hours, the stability of the power will be highly required: we use a 400mW LED light source (Thorlabs Inc, New Jersey, USA) as our excitation light source, which is a much more stable excitation light source than traditional white light bulbs.

To ensure the comparability of the time series gene expression results from different days, experiments are carried on using the same procedure every day. The detailed steps are listed below. One hour before starting the experiment, the thermal plate and the camera are turned on for warming up to ensure they have reached a steady state before the experiment is started. Reaction solutions are thawed and mixed on ice by mixing $17.5\mu l$ *E. coli* extracts, $0.2 \mu g (10 \mu l)$ plasmid, $20\mu l$ reaction buffer, and $20\mu l$ EB buffer with or without bovine serum albumin (BSA). The mixed reaction solution is centrifuged for 10 seconds to ensure all reaction solution is collected at the bottom of the small tube, vortexed gently for 20 seconds and then sonicated for 5 minutes in ice water to ensure a well-mixed reaction solution. 20$\mu l$ mixed reaction solution is then transferred into the incubation chamber and sealed with a thin cover glass. The sample is put inside an ice bucket and ready for the time series gene expression experiment. When waiting for the thawing of the sample or sonicating, the microscope is calibrated by measuring fluorescence intensity of two standard fluorescent samples - two highly concentrated quantum dot solutions (Vive Nano, Toronto, Canada). After the sample is transported onto the microscopy stage, the cover glass, used to seal the incubation chamber, is marked using a marking pen, aiding focusing on the surface of the cover glass, after which the fine tune focusing wheel is turned by one full counter-clockwise turn to bring the focal point 0.1mm deep inside the reaction solution. The incubation chamber is then covered with the thermal plate, pre-heated to $37^\circ$ C. The whole microscope is covered with a soft black sheet to prevent any background light from entering into the objective and being treated as a fluorescence signal. METAMORPH is configured to open the illumination shutter and take a fluorescence image of the reaction solution every 10 minutes for many hours. The data is first analyzed in METAMORPH to gather the fluorescence level of the solution at each time point and then transferred to MATLAB for further analysis.
7.3 Result

The results for the in vitro expression of EGFP when there is no crowding agent, with 20µl EB buffer, and when there is crowding agent, with 20µl 200µg/µl BSA solution - diluted in EB buffer, making 13.5% volume (calculations shown below) occupied by macromolecules BSA, is plotted in Figures 7.2 and 7.3. To ensure the pattern we see does not result from the random day-to-day variation, 4 runs are used to calculate the average time series of EGFP expression, including the error bar.

The size of a BSA molecule has been reported to be 40x40x140 cubic angstroms = 2.24 × 10^{-25} m^3 [115]. The number of BSA molecules in our reaction solution is 20\( \mu l \times 200 \mu g/\mu l / 1.10 \times 10^{-22} kg = 3.636 \times 10^{16} \), where 1.10 \times 10^{-22} kg is the weight of a BSA molecule. The percentage of volume occupied by the BSA molecules can be calculated to be 3.636 \times 10^{16} \times 2.24 \times 10^{-25} m^3 / 60.5 \mu l = 13.5\% . Note that the in vitro kits also contain an unknown amount of macromolecules. This will make the total fraction of the volume occupied by macromolecules larger than 13.5\%.

Figure 7.2 shows the absolute fluorescence intensity, represented by the average count on CCD at each time point, over time. The EGFP expression in no-crowding-agent case not only has a faster EGFP production speed in the first couple of hours, but also reaches a higher maximum fluorescence level at about 1.5 hour, compared to the EGFP expression in the with-crowding-agent environment. Figure 7.3 shows the relative fluorescence intensity over time, with the maximum fluorescence level set to be 1 and initial fluorescence level set to be 0. In this figure, it is more clear that the EGFP expression in the more crowded environment is slower than in the less crowded case. The slowing-down of EGFP expression in more crowded environments can be attributed to the slowing-down of diffusion in such environments, making the reaction rates of second order reactions decreasing. The difference between the absolute fluorescence level in the less-crowded and more-crowded cases can be attributed to the short life time of mRNA (around 5 minutes): Suppose the life time of mRNA is the same for both cases. If the crowding makes the translation of protein from mRNA slower, in the life time of a single mRNA, it will be translated less times. Further, if the total number of mRNA that can be produced, limited by the resources in the reaction solution, is the same for both cases, the total number of proteins will be smaller in the more crowded case.

If we focus on the behavior of the gene expression after around 2 hours, interesting behavior appears. Both fluorescence levels of the less crowded and more crowded cases decrease with time, probably representing protein degradation. But the decreasing of fluorescence intensity in the less crowded case is much faster than the more crowded case. We believe that this faster degradation phenomenon is due to the fact that EGFP becomes more stable in more crowded
Figure 7.2: Time series expression of EGFP *in vitro* with and without crowding agents. Fluorescence intensity is characterized by the average count per pixel on CCD. Each line shown on the plot is generated by averaging over 4 individual runs, and the error bar shows one standard error from the mean value.
Figure 7.3: Time series expression of EGFP *in vitro* with and without crowding agents. Fluorescence intensity is normalized such that the maxim fluorescence level is equal to 1. Each line shown on the plot is generated by averaging over 4 individual runs, and the error bar shows one standard error from the mean value.

environments, which has been shown by other researchers [116].

### 7.4 Modelling of gene expression in crowded environments

To further understand how macromolecular crowding changes the gene expression dynamics of the system, a simple model is formulated to simulate the experimental observation, as shown in Eq. 7.1 to Eq. 7.6.
List of variable names

<table>
<thead>
<tr>
<th>Name of Variable</th>
<th>Description of Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_m$</td>
<td>Total amount of mRNA produced from time 0 to time t</td>
</tr>
<tr>
<td>$T_p$</td>
<td>Total amount of protein produced from time 0 to time t</td>
</tr>
<tr>
<td>$m$</td>
<td>Amount of mRNA at time t</td>
</tr>
<tr>
<td>$p$</td>
<td>Amount of protein at time t</td>
</tr>
<tr>
<td>$p_m$</td>
<td>Amount of matured protein at time t</td>
</tr>
</tbody>
</table>

Table 7.1: List of names of variable

\[
\emptyset \overset{k_1}{\rightarrow} \text{mRNA} \\
\text{mRNA} \overset{k_1'}{\rightarrow} \emptyset \\
\text{mRNA} \overset{k_2}{\rightarrow} \text{mRNA + Protein} \\
\text{Protein} \overset{k_2'}{\rightarrow} \emptyset \\
\text{Protein}_{\text{matured}} \overset{k_3}{\rightarrow} \text{Protein}_{\text{matured}} \\
\text{Protein}_{\text{matured}} \overset{k_2'}{\rightarrow} \emptyset
\]  

(7.1) (7.2) (7.3) (7.4) (7.5) (7.6)

The rate constants in the above reactions can not be constant, because the resources inside the reaction solution is limited, and will be consumed over time, which will affect the reaction rates: transcription rate $k_1$ and translation rate $k_2$. To incorporate the effect of limited resources, I multiply the transcription and translation rate by this formula: $(\frac{c^n}{c^n + T^n})^n$, where $c$ is an arbitrary number, $n$ is an integer and $T$ represents the total number of mRNA or protein being produced. When the total number $T$ is smaller than $c$, this formula will approximately equal to 1, meaning that the reaction solution has enough resources for the transcription and translation reaction to occur at maximum speed, while, as $T$ becomes closer to and then larger than $c$, the value becomes nearly zero and the transcription and translation rates will be substantially reduced, representing the using up of the resources. Let us first consider the case when the reaction environment is a well-mixed environment: the differential equations describing the time evolution of the gene expression dynamics in such environments can be written as Eq. 7.7 to Eq. 7.11. The complete list of names of variable in the differential equations is shown in Table 7.1. $n_p, n_m, c_m, c_p$, and reaction rate constants are obtained by fitting the model to experimental data.
\[
\frac{dT_m}{dt} = \left( \frac{c_{nm}^m}{c_{nm}^m + T_{nm}^m} \right)^n m k_1 (t + \tau_m)^h_m \tag{7.12}
\]
\[
\frac{dT_p}{dt} = \left( \frac{c_{np}^p}{c_{np}^p + T_{np}^p} \right)^n p k_2 (t + \tau_p)^h_p \tag{7.13}
\]
\[
\frac{dm}{dt} = \left( \frac{c_{nm}^m}{c_{nm}^m + T_{nm}^m} \right)^n m k_1 (t + \tau_m)^h_m - k'_1 m \tag{7.14}
\]
\[
\frac{dp}{dt} = \left( \frac{c_{np}^p}{c_{np}^p + T_{np}^p} \right)^n p k_2 (t + \tau_p)^h_p m - k'_2 p - k_3 p \tag{7.15}
\]
\[
\frac{dp_m}{dt} = k_3 p - k'_2 p_m \tag{7.16}
\]

In crowded environments, despite the possible slowing down of reactions due to the consumption of the resources, the reduced degree of mixing with time will also slow down the transcription and transcription reaction. To represent this effect in the model, an additional term \( \frac{1}{(\tau + t)^\nu} \), will be multiplied to the rate constant [91]. The resulting differential equations are shown from Eq. 7.12 to 7.16.

The fitting of Eq. 7.7 to Eq. 7.11 to the experimental curve of gene expression without crowding agent is shown in Figure 7.4. Keeping all the fitted parameters except protein degradation rate, Eq. 7.12 to 7.16 is used to fit to the experimental curve of gene expression with crowding agent. The re-fitted protein degradation rate for the crowding case is 0.0015, compared to the non-crowding degradation rate 0.0043, we can see that the macromolecular crowding makes the protein more stable, which is consistent with previous observations [116]. Moreover, the fitted \( \tau_m, \tau_p, h_m, \) and \( h_p \) are 1.723, 1.032, 0.151, and 0.414 respectively. Those parameters will make the reaction rates decrease with time, which is consistent with the assumption that macromolecular crowding will slow down the reactions.
Figure 7.4: Fitting of experimental data to proposed models.
7.5 Conclusion

Our experimental results clearly show that the macromolecular crowding will change the reaction dynamics of gene expression. The gene expression of our reporter protein EGFP is slowed down in more crowded environments, compared to the less crowded case. Also the maximum level of protein being produced in the more crowded environment is also less, probably due to the limited life time of mRNA.

Our modelling results, using simple ODE models, show that the behavior of the system in the non-crowded case can be captured by adding an extra formula representing the consumption of the resources to the common ODE models for transcription and translation, while the effect of crowding can be captured by incorporating time dependent reaction rates, which is in agreement with other simulation results [91].
Chapter 8

Conclusion

Quantitative understanding of the gene expression dynamics and noise is one of the key problems in quantitative biology. During my study, I have combined experimental and modelling approaches to answer some unresolved questions related to the source of noise in *E. coli* gene expression and the time dependent gene expression dynamics.

Comparison of models with and without protein maturation process shows that the maturation process, necessary for widely used fluorescent proteins to give out fluorescence, has a non-negligible effect on the mean level and the noise in gene expression models. The maturation process does not necessarily make the system more noisy: in some parameter regimes, the maturation process suppresses the noise in the system, while it increases the noise in some other parameter regimes. This phenomenon comes from the fact that the maturation steps (and more broadly speaking, any conversion reactions) can act as a low pass filter - filtering out noise in low frequency range, in addition to adding its own intrinsic noise to the system. The total effect of protein maturation steps on the gene expression noise will depend on which effect is dominant: if the amount of noise that is filtered out is more than the amount of noise induced by the maturation steps, the total noise in the system will be decreased; otherwise, the noise will be increased. From our observations, generally speaking, when the translation rate is high, the noise introduced by the maturation steps will be less than the noise filtered out by the maturation steps, resulting in a reduction of total noise. No matter what is the effect of the protein maturation steps on the noise in the model, the protein maturation steps will bring large changes to the model in terms of the mean number of protein and the noise in protein numbers. The consideration of the protein maturation process is necessary to model the quantitative gene expression process accurately. Possible future extensions of this work include extending the result to nonlinear models, investigating what would be the effect of protein maturation steps on the gene expression noise with external noises present, and experimental verification of the modelling results.
Comparison of models with and without negative feedback shows that negative feedback increases the intrinsic noise of a gene expression model, and decreases the extrinsic noise. The total effect of negative feedback on the gene expression noise depends on which effect is dominant. In living cells, extrinsic noise is usually the dominant source of noise, thus the reduction of extrinsic noise is more apt to surpass the increase of intrinsic noise, resulting in a total reduction of noise. Experimental work is currently being carried out by other students to verify the modelling predictions.

During our 12 hours observation of the gene expression dynamics in *E. coli* batch culture, the cell size, the gene expression level and the noise keep changing with time. Moreover, the time dependent behaviors for different plasmid systems are different, and if we compare the noise between different systems at different time points, we might reach divergent conclusions. Thus if we want to measure or compare the properties of populations of cells in *E. coli* batch culture, we will have to specify the time point when we are measuring. Our modelling result of gene expression dynamics over time implies that the time dependent pattern of protein number mainly comes from the time dependent cell growth rate and cell division volume. While the model captures part of the qualitative behavior of the system, a more detailed model is necessary to obtain more thorough and deep explanation of the experimental observation. The detailed model might be constructed by combining published metabolism models with all different kinds of experimental data. The hope, after a more detailed model is formulated, is to predict the time dependent cell growth, cell division and gene expression processes in *E. coli* batch culture. Another interesting addition is to observe the time varying behavior of individual cells under microscope, which will give much more information than measuring the change of distributions over time. To observe individual cells, they can be confined in a microfluidic device. Further, the nutrition condition in the microfluidic devices can be varied to observe the growth, division and gene expression of individual cells under different nutrition conditions. With all those information, a better model could be formulated to better explain and understand the gene expression noise and dynamics in living cells.

The experimental observation of the macromolecular crowding effect on the gene expression dynamics shows that macromolecular crowding slows down the gene expression rate with time, and brings down the gene expression level. Using simple ODE models, we have shown that the nutrition consumption *in vitro* can be modelled using a Hill function, while the macromolecular crowding effect can be modelled using time dependent transcription and translation rates. The volume of the reaction solution in my experiment is around 30 µl, which is considerably larger than the size of a single cell. If the reaction volume can be reduced to the femtoliter range using liposomes or microfluidic devices, the effect of macromolecular crowding on the gene expression noise could be observed. Moreover, a detailed model keeping track of individual cells might
provide a better explanation of what is happening in living cells. Another possible extension is using molecular beacons or other techniques to monitor the level of mRNA over time. Note that the whole study is carried out in vitro: by working in vitro, we gain the ability to control the content of the reaction solution directly, but the dynamics in vitro might be different from the gene expression dynamics in vivo. Thus, the repeating of this work in vivo might provide some new and valuable information.

Stochastic simulations are usually time intensive. One way to improve the simulation speed is to reduce the complexity of the biochemical reaction models. To accomplish this, I have developed a method to simplify biochemical reaction models with fast and slow dynamics. The simplified models takes much less simulation time than the original model, while still predicting very similar results compared to the original model. Another way to speed up the simulation is to exclude gene expression from the genome when modelling the gene expression from plasmids. By modelling, we have found that when simulating the gene expression from plasmid, we can ignore the gene expression from genome when certain criteria is satisfied - the binding rate of RNA Polymerase to the promoter region on the plasmid falls within certain range. By ignoring the background gene expression from the genome, the simulations can be sped up substantially. However, modelling the gene expression behavior using detailed biochemical reaction models require the detailed rate constants for interactions between molecules, which are largely unavailable and difficult to measure experimentally. Thus, constraint-based approaches, or control theory might be better ways to model the gene expression dynamics observed experimentally.

In summary, the quantitative understanding of the gene expression dynamics and noise requires detailed observations of the gene expression processes in living cells, and requires realistic models to explain and predict the gene expression dynamics and noise. To better observe the gene expression process in living cells, high throughput methods to observe the cell growth, the cell division, and the number of plasmid, mRNA and protein of individual cells, or to monitor single gene expression events in living cells are highly valuable. To better model the gene expression process, constraint-based approaches, control theory and detailed biochemical reaction models might be combined to give a better description of gene expression dynamics and noise.
Bibliography


Bibliography


