Computational design and characterisation of synthetic genetic switches

Miriam Leon

A thesis submitted in partial fulfilment of the requirements for the degree of:

Doctor of Philosophy of
University College London

2016

Primary supervisor:
Dr. Chris P Barnes

Secondary supervisor:
Prof. Geraint MH Thomas

I, Miriam Leon, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

Genetic toggle switches consist of two mutually repressing transcription factors. The switch motif forms the basis of epigenetic memory and is found in natural decision making systems, such as cell fate determination in developmental pathways. A synthetic genetic switch can be used for a variety of applications, like recording the presence of different environmental signals, for changing phenotype using synthetic inputs and as building blocks for higher-level sequential logic circuits.

In this thesis, the genetic toggle switch was studied computationally and experimentally. Bayesian model selection methods were used to compare competing model designs of the genetic toggle switch. It was found that the addition of positive feedback loops to the genetic toggle switch increases the parametric robustness of the system.

A computational tool based on Bayesian statistics was developed, that can identify regions of parameter space capable of producing multistable behaviour while handling parameter and initial conditions uncertainty. A collection of models of genetic switches were examined, ranging from the deterministic simplified toggle switch to stochastic models containing different positive feedback connections. The design principles behind making a bistable switch were uncovered, as well as those necessary to make a tristable or quadristable switch.

Flow Cytometry was used to characterise a known toggle switch plasmid. A computational tool was developed which uses Bayesian statistics to infer model parameter values from flow cytometry data. This tool was used to characterise the toggle switch plasmid and fit a stochastic computational model to experimental data.

The work presented here suggests ways in which the construction of genetic switches can be enhanced. The algorithms developed were shown to be useful in synthetic system design as well as parameter inference. The tools developed here can enhance our understanding of biological systems and constitute an important addition to the systems approach to synthetic biology engineering.

Acknowledgements

I would like to thank the people whose help and support made the work presented in this thesis possible.

First and foremost I would like to thank my supervisor, Dr. Chris Barnes. The hard work and dedication you have shown your students have always been inspiring and appreciated. None of this would have been possible without your push, encouragement and patience which were sometimes the only things that kept me going. I was lucky to be your first student, a badge I will always wear with pride! Thank you for trusting me, a young biologist, with this project and for always having the time and will to help me through this journey.

I would also like to thank Professor Geraint Thomas. Your cheerful attitude to science has never failed to lift my spirits. Your ability to put problems into perspective has always allowed me to keep pushing forward.

I am grateful for the funding I received through the UCL Impact Award scheme.

Thank you to Professor David Hall for his help with plasmid cloning and flow cytometry. I would also like to thank Dr Gyorgy Szabadkai's lab for their help with flow cytometry, Dr. Ayad Eddaoudi in the flow cytometry core facility at the UCL lnstitute of Child Health and Dr Malte Paulsen at the Flow Cytometry Core Facility at the National Heart & Lung Institute (NHLI), Imperial College London.

To all the members of the Computational Systems and Synthetic Biology (CSSB) group, especially Gerold Baier, Alex Fedorec, Riana Gaifulina, David Gonzales, Aaran Lewis, Phil Lewis, Helina Marshall, Tanel Ozdemir, Lourdes Sriraja, Gordon Walsh, Marc Williams, Bethan Wolfenden, and Mae Woods. Thank you all for your constant support and friendship. A special thanks to Alex Fedorec for proofreading this thesis and to Tanel Ozdemir for spending hours teaching me how to work in the lab, as well as Lewis Brayshaw for giving me cloning advice. A special thanks also to Dr. Mae Woods for always helping me and being supportive with any maths and coding questions, for your kind words and for proofreading this thesis. I would also like to thank Dr. Robert Stanley for kindly providing the technical help in the

writing of this thesis. And to all, thank you for being such a great team.

To my friends that kept me sane throughout this journey, I thank all of you for your constant encouragement. To Effie and Eva for setting an example of what courage and hard work can achieve. To Daisy and Dafni for always believing in me and listening to my woes. To Daniel and Sinan for setting the bar high while encouraging me to reach it. To Adam, Jon, Meg, and Zosia for taking me on adventures when they were most needed.

To my family who have always been my cheerleaders. Mum, Dad, Evie thank you for your unconditional love and support. I have been fortunate to have you on my team.

Finally, to Takao for your support and love. From going over algorithms to endless emotional support, your help in every aspect of this endeavour has made it all possible.

Contents

List of Figures

List of Tables

Abbreviations

ABC Approximate Bayesian Computation.

aTc anhydrotetracycline.

BSA Bovine Serum Albumin.

CS-LU Lu classic switch.

CS-MA Mass action classic switch.

DNA Deoxyribonucleic acid.

dNTPs Deoxynucleotide.

DP-LU Lu double positive switch.

DP-MA Mass action double positive switch.

GFP green fluorescent protein.

GPU Graphical Processing Units.

IPTG Isopropyl-beta-D-thiogalactopyranoside.

KS Kolmogorov-Smirnov.

LB Lysogeny broth.

MCMC Markov Chain Monte Carlo.

MJP Markov jump process.

ODE Ordinary differential equation.

PCA Principal component analysis.

PCR Polymerase Chain Reaction.

QSSA quasi-steady state approximation.

SBML Systems Biology Markup Language.

SDE Stochastic differential equation.

SMC Sequential Monte Carlo.

SP-LU Lu single positive switch.

1 Introduction

1.1 Introduction to synthetic biology

Synthetic biology is centered around the rational design and construction of biological parts, devices, and systems in order to engineer organisms to perform new tasks (Lu, Khalil, & Collins 2009; Andrianantoandro et al. 2006). A part is a basic unit, like a promoter or a ribosome binding site that when combined with other parts will make a functional unit, a device (Heinemann & Panke 2006). A device processes inputs performs functions and produces outputs (Andrianantoandro et al. 2006). A system comprises of a collection of devices.

Emphasis is put on the use of engineering principles such as modularity, standardisation, use of predictive models and the separation of design and construction (Agapakis & Silver 2009; Heinemann & Panke 2006). An analogy can be drawn with the hierarchy used in computer science, with cells, pathways and biochemical reactions acting as computers, modules and gates respectively (Andrianantoandro et al. 2006).

Numerous applications of synthetic biology have emerged, from altering existing metabolisms (Wang & Yu 2007) to producing synthetic drugs (Ro et al. 2006) or creating new synthetic life forms (Hutchison et al. 2016). Despite the successes there is still a lack of predictive power due to the stochasticity and lack of complete knowledge of the cellular environment (Andrianantoandro et al. 2006).

1.2 Quantitative modelling in synthetic biology

Synthetic biology draws from the multidisciplinary work of biologists, mathematicians, computer scientists, physicists and chemists (Vinson & Pennisi 2011) in order to engineer biology. The randomness of the biological environment, the plethora of unknowns in the engineered system as well as its surrounding environment

and their interaction therein makes this task extremely challenging. The field has made great advancements in recent years, and a collection of simple synthetic circuits have been built such as toggle switches (Gardner, Cantor, & Collins 2000; Kramer et al. 2004; Isaacs et al. 2003; Ham et al. 2008; Deans, Cantor, & Collins 2007; Friedland et al. 2009), oscillators (Stricker et al. 2008; Fung et al. 2005; Tigges et al. 2009) and pulse generators (Basu et al. 2004).

These synthetic circuits have been built to imitate controllers from electrical engineering, like logic gates, switches, and oscillators, but the inherent complexity and stochasticity of biology and the cellular environment make their predictability and application challenging. This has highlighted the importance of using more advanced computational tools to aid in the design, and ultimately the construction, of novel synthetic biological devices.

This has led to systems and synthetic biology increasingly being merged together in an effort to understand the inherent complexity of engineering biological systems (Gramelsberger 2013). Quantitative modelling has been used to aid and improve the systems under consideration. Successful examples include that of Stricker et al. (2008) and Entus, Aufderheide, & Sauro (2007). Stricker et al. (2008) designed a genetic oscillator and mathematical modelling of the system allowed them to identify the parameters of their system that give rise to oscillations. Entus, Aufderheide, & Sauro (2007) used modelling to design and construct incoherent feed-forward loops in *E.coli*.

The design of genetic circuits has an additional challenge compared to other areas of genetic engineering. The components of the circuits have to be finely tuned to work together towards the desired behaviour of the system. This is in contrast to engineering a cell to produce a single protein where its production has to be maximized (Nielsen, Segall-Shapiro, & Voigt 2013). The need to orchestrate a number of genetic components toward a common goal has made the integration of systems and synthetic biology all the more important.

In this work I use quantitative modelling to understand a synthetic system. I look at the problem from two different but related perspectives, design and inference. I aim to improve on the design of a synthetic biological system and understand the principles dictating the behaviour of new designs. I also aim to quantitatively study an existing system and infer the underlying principles that govern its behaviour.

1.3 Thesis Outline

This thesis will focus on the biochemical modelling and analysis of the genetic toggle switch. The thesis is organised as follows:

Chapter 2 provides an introduction to biochemical modelling. It contains an overview of the mathematical methods that formed the basis of the methods used throughout this thesis. It also contains a literature review on the current understanding on the dynamics of the genetic toggle switch. I provide material that is necessary for the understanding of the rest of this thesis.

In **Chapter 3** I explore the effect that adding feedback loops has on the stability and parametric robustness of the toggle switch. I develop more realistic biochemical models of the genetic toggle switch and study their ability to behave like a switch.

In **Chapter 4** I develop a parameter estimation algorithm for multistable switches, called StabilityFinder. I benchmark this algorithm using a toggle switch model with known results. I then apply it to extensions of the simplified toggle switch as well as more realistic models of the toggle switch developed in Chapter 3 in order to study the design principles that make a multistable switch. Finally, I develop an algorithm for estimating the robustness of a system using the results from StabilityFinder and use it to study the effect of feedback loops on the robustness of the switch.

In **Chapter 5** I develop an algorithm based on Bayesian statistics for parameter estimation of flow cytometry data, called ABC-Flow. I also characterise the genetic toggle switch experimentally and provide an overview of the methods used. Finally, I apply ABC-Flow to the experimental data collected and infer the parameters that give rise to the data.

In **Chapter 6** contains a detailed experimental plan to be carried out as the next step to this work. I outline the steps needed to design and construct genetic toggle switches with added autoregulating feedback loops experimentally.

Chapter 7 concludes this thesis with an overview of the work presented here and a discussion of future directions.

Work carried out throughout my candidature has been published in the following article:

- Leon, M., Woods, M., Fedorec, J. A., & Barnes, C. P. (2016). ‘A computational method for the investigation of multistable systems and its application to ge-

netic switches'. [Submitted].

- Leon, M. & Barnes, C. P. (2016). 'Characterising a genetic toggle switch using Approximate Bayesian Computation'. [In preparation].

2 Background

2.1 Current understanding of the genetic toggle switch

One of the first successfully constructed synthetic devices is the genetic toggle switch. A toggle switch consists of a set of transcription factors that mutually repress each other (Gardner, Cantor, & Collins 2000). Genetic switches play a major role in binary cell fate decisions like stem cell differentiation, as they are capable of exhibiting bistable behaviour. Bistability of a system is defined by the existence of two distinct phenotypic states but no intermediate state. Bistability is a property that is important in nature and a valuable resource to exploit in synthetic biology. It allows cells to alter their response to environmental cues and increases the overall population fitness by 'hedge-betting' the response of the population (Veening, Smits, & Kuipers 2008).

2.1.1 The genetic toggle switch in natural systems

In developmental processes, bistability ensures that the differentiating cell will follow one pathway, or the other, with no possible intermediate phenotypes. This is vital for the correct development of a cell in a specific pathway. One example is the trophectoderm differentiation pathway, in which a mutually inhibitory toggle switch exists between Oct3/4 and Cdx2. This determines whether an embryonic stem cell will differentiate into a trophectoderm cell, if Cdx2 dominates the system, or an inner cell mass cell if Oct3/4 dominates (Niwa et al. 2005). Bistability is critical in this system as a cell must differentiate into either a trophectoderm cell or an inner cell mass cell and there should not be any intermediate signals.

In the case of the GATA1 and PU.1 toggle switch, the transcription factor pair controls the fate of the common myeloid progenitors, and the two possible differentiation paths are erythroid and myeloid blood cells (Liew et al. 2006). The double-negative feedback loop created by the mutually repressive pair of transcrip-

tion factors sustains the system in balance until an external stimulus causes one of the two transcription factors to increase in concentration. The increased concentration of one transcription factor causes the increased repression of the production of the antagonistic transcription factor, tipping the balance towards the dominance of the first transcription factor. The double negative feedback loop reinforces this dynamic and the system remains in the same state until an external stimulus disturbs it (Ferrell 2002).

2.1.2 Uses in synthetic biology

Despite their simplicity, toggle switches can be powerful building blocks with which to create complex responses in a synthetic network (Lu, Khalil, & Collins 2009). They can be used in isolation and have the potential to be used tandem to create complex networks and signalling cascades (Lu, Khalil, & Collins 2009). The toggle switch has been used for the regulation of mammalian gene expression (Deans, Cantor, & Collins 2007; Kramer et al. 2004). Other synthetic applications of the toggle switch include the construction of a synthetic genetic clock (Atkinson et al. 2003), of a predictable genetic timer (Ellis, Wang, & Collins 2009), and the formation of biofilms in response to engineered stimuli (Kobayashi et al. 2004).

These applications are modifications of the classical toggle switch (Gardner, Cantor, & Collins 2000), but an application made of a cascade or collection of the switch would be more challenging. This would make more complex applications possible and could be used to solve real-life problems. For example, an analog-to-digital converter to translate external stimuli like the concentration of an inducer into an internal digital response, or programmable bacteria to move from point to point up different chemical gradients (Lu, Khalil, & Collins 2009). For a review on current circuits see (Khalil & Collins 2010) and for possible future applications see (Lu, Khalil, & Collins 2009). This leap will be difficult to achieve before first being able to build robust and well characterised individual switches.

2.1.3 Modelling the genetic toggle switch

The toggle switch motif has been studied extensively and there are numerous studies based on a number of different methods of modelling and analysis of the dynamics, including both deterministic and stochastic approaches. The conclusions drawn about the stability and robustness of the toggle switch vary between the different modelling approaches. Numerous studies have concluded that cooperativity is a

necessary condition for bistability to arise (Gardner, Cantor, & Collins 2000; Walczak, Onuchic, & Wolynes 2005; Warren & ten Wolde 2004; Warren & ten Wolde 2005; Cherry & Adler 2000). However, Lipshtat et al. (2006) found that stochastic effects can give rise to bistability even without cooperativity in three kinds of switch; the exclusive switch, in which there can only be one repressor bound at any one time, a switch in which there is degradation of bound repressors, and the switch in which free repressor proteins can form a complex, which renders them inactive as transcription factors (Lipshtat et al. 2006).

In another study, Ma et al. (2012) found that the stochastic fluctuations in a system involving such a small number of molecules, like the toggle switch, uncovers effects that cannot be predicted by the fully deterministic case (Ma et al. 2012). In their system, the toggle switch was found to be tristable, as small number effects render the third unstable steady state stable. Biancalani & Assaf (2015) identified multiplicative noise as the source of bistability in the stochastic case (Biancalani & Assaf 2015). Warren & ten Wolde (2005) concluded that the exclusive switch is always more robust than the general switch since the free energy barrier is higher (Warren & ten Wolde 2005). A summary of the toggle switch models is shown in Table 2.1. As is clear from above, there is yet to exist a consensus on the stability a switch is capable of, and the most appropriate method of modelling it. Different methods arrive at different conclusions, creating confusion on which behaviour to be expected by the experimentalist for even a simple system like the toggle switch, consisting of just two genes. The toggle switch cannot be used as a building block of larger, more complex systems until its behaviour can be predicted accurately. Until then, designing systems with predictable behaviour will be near impossible.

Table 2.1 Summary of stabilities for the classical switch and the switch with double positive feedback found via different modelling approaches

	Standard toggle switch		Double positive autoregulation
Stability	Deterministic	Stochastic	Deterministic
Monostable	(Loinger & Biham 2009)	(Loinger & Biham 2009)	(Guantes & Poyatos 2008)
	(Gardner, Cantor, & Collins 2000)	(Lu, Onuchic, & Ben-Jacob 2014),	
Bistable	(Loinger & Biham 2009)	(Lipshtat et al. 2006),	(Guantes & Poyatos 2008)
		(Biancalani & Assaf 2015),	
		(Loinger & Biham 2009)	
Tristable		(Loinger & Biham 2009),	(Guantes & Poyatos 2008),
		(Ma et al. 2012)	(Lu, Onuchic, & Ben-Jacob 2014)
Quadristable			(Guantes & Poyatos 2008)

2.2 Methods in biochemical modelling

Modelling attempts to describe the elements and dynamics of the biochemical system of interest. It is a tool used for integrating knowledge and experimental data as well as for predicting the behaviour of the system (Wilkinson 2006).

2.2.1 Representation of transcription networks

A transcription network can be represented in a number of ways. A network can be described by using a diagram supplemented by verbal explanations or by a set of differential equations. A diagram with a lengthy verbal explanation risks the not providing sufficient clarity whereas a set of differential equations cannot easily be separated from the underlying assumptions made on the kinetics of the network. A convenient way of describing sufficient information about a system while avoiding the addition of the particular interpretation of the underlying kinetics is the use of coupled chemical reactions (Wilkinson 2006).

2.2.1.1 Coupled chemical reactions and the law of mass action

Coupled chemical reactions are often used to describe transcription networks in systems biology. They have the advantage of describing a system concisely while they can be used subsequently for a variety of different simulation methods, each with their associated interpretation of chemical kinetics (Wilkinson 2006). Coupled chemical reactions take the form

$$R_a + R_b \xrightarrow{k} P, \tag{2.1}$$

where R_a represents reactant A, R_b reactant B and P a product. Each reaction has an associated rate constant k. A biological transcription network can be represented using the above notations. Some common examples of coupled chemical reactions used in a biological network are given in Table 2.2. A double headed arrow represents a reversible reaction.

The law of mass action allows one to derive these reaction rates from the coupled chemical reactions. The assumption made in the law of mass action is that the system exists in a well-mixed solution and in dynamic equilibrium. The law of mass action states that the reaction rate is equal to the concentration of the reactants multiplied by a rate constant. So for a given chemical equation as the one shown

Table 2.2 Examples of common genetic coupled chemical reactions. p stands for promoter, and A represents a protein. A_2 is the dimer of protein A.

Event	Coupled chemical reaction	Rates
Transcription	$p \xrightarrow{k_1} p + RNA$	$k_1[p]$
Dimerization	$2A \underset{k_3}{\overset{k_2}{\rightleftharpoons}} A_2$	$k_2[A][A]$, $k_3[A_2]$
Promoter repression	$A_2 + p \underset{k_5}{\overset{k_4}{\rightleftharpoons}} p \bullet A_2$	$k_4[A_2][p]$, $k_5[p \bullet A_2]$
Activation	$A_2 + p \xrightarrow{k_6} p \bullet A_2 + RNA$	$k_6[A_2][p]$
Degradation	$A \xrightarrow{k_7} \emptyset$	$k_7[A]$

in Equation 2.1, the rate of the reaction is defined by:

$$rate = k[R_a][R_b],$$

where k is the rate constant.

2.2.1.2 Graphical representation of biochemical systems

It is common to represent coupled biochemical reactions graphically. In a graph, as shown in Figure 2.1, nodes represent the species and the edges represent an interaction between the species it connects, in which a transcription factor directly affects the transcription of a gene (Alon 2007). An arrow at the end of an arc represents activation, i.e. that when the transcription factor binds to the promoter the rate of transcription of the gene increases. A flat line perpendicular to the arc at the end of an arc represents repression, i.e. that when the transcription factor binds to the promoter the rate of transcription of the gene decreases (Alon 2007).

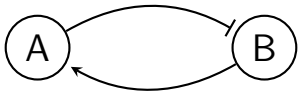

Figure 2.1 A graphical representation of a biochemical system. The two nodes, A and B represent species and the edges (arrows) a reaction between the two. An arrow represents activation and a flat line represents repression.

2.2.1.3 Systems Biology Markup Language (SBML)

The Systems Biology Markup Language (SBML) was developed by Hucka et al. (2003) in order to allow for the exchange of biochemical models between software. It is an

extension of the XML encoding (DuCharme 1999) with additional fields specific to biochemical models. Software like Copasi (Hoops et al. 2006) can be used to convert a set of coupled chemical reactions to an SBML model. SBML models have been a key resource for model sharing within the systems biology community (Wilkinson 2006) in databases like the BioModels database (Le Novère et al. 2006).

2.2.2 Transcriptional binding kinetics

The processes of transcription regulation in prokaryotes are complex and there have been a number of mathematical descriptions developed to approximate the dynamics observed. These include the Hill equation and the Shea-Ackers formalism.

2.2.2.1 Hill formalism

The Hill formalism is often used to describe a biochemical system where an activator or repressor is present (Hill 1910). The Hill function is often represented as

$$\frac{dP}{dt} = V_{max}\frac{x^n}{1+x^n},$$

if activation is being modelled. This is an increasing S-shaped function. Parameter n is the Hill coefficient and K the dissociation constant. V_{max} is the maximum amount of product and $x = \frac{S}{K}$, where S is the substrate concentration. The Hill function reaches a plateau at high substrate concentrations, as is often seen in biological reactions (Alon 2007). K represents the substrate concentration that results in half of the response and the Hill coefficient affects the steepness of the function and represents the cooperativity of the binding to the promoter (Alon 2007). If repression is being modelled, the Hill function is represented as

$$\frac{dP}{dt} = V_{max}\frac{1}{1+x^n}.$$

An example of the effect that the value of the Hill coefficient n has on the shape of the Hill function in both activation and repression is given in Figure 2.2. The higher the value of n, the more step-like the function becomes (Alon 2007).

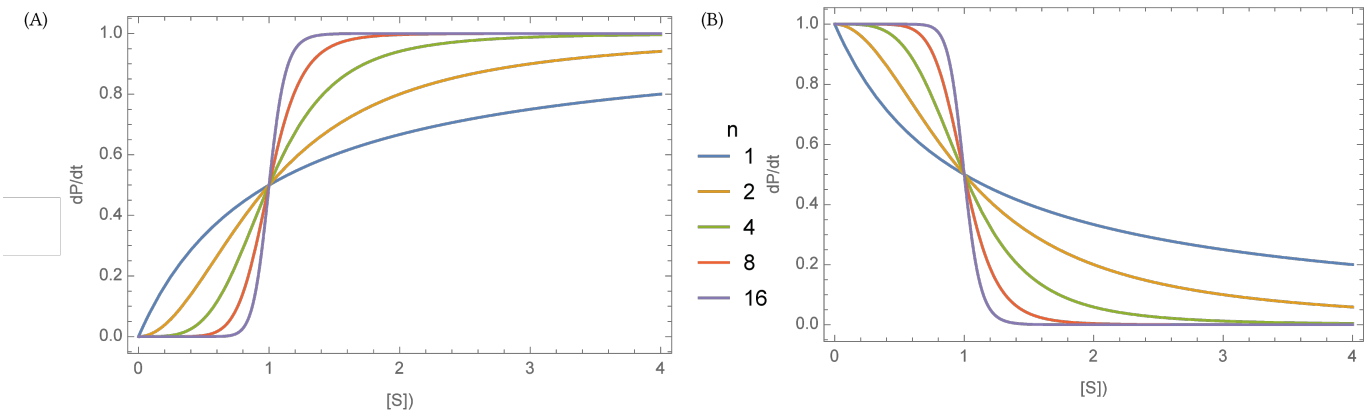

Figure 2.2 The effect of different values of n on the Hill function when K is kept constant in the case of (A) activation and (B) repression.

2.2.2.2 Shea-Ackers formalism

The Shea-Ackers formalism developed by Ackers, Johnson, & Shea (1982) uses a statistical thermodynamic model to represent the binding of transcription factors to their promoters. A system is described by the various states the promoter can have. An example of possible states is given in Figure 2.3. Each state has an associated term, or weight, and the probability of transcription is given by the ratio of the producing states over all possible states. This is referred to as the partition function.

$$P_T = \alpha \frac{k_1 + k_3 A^2}{1 + k_1 + k_2 R^2 + k_3 A^2}. \tag{2.2}$$

Here I assume that repression and activation is cooperative, thus two transcription factors must bind to the promoter to repress or activate it.

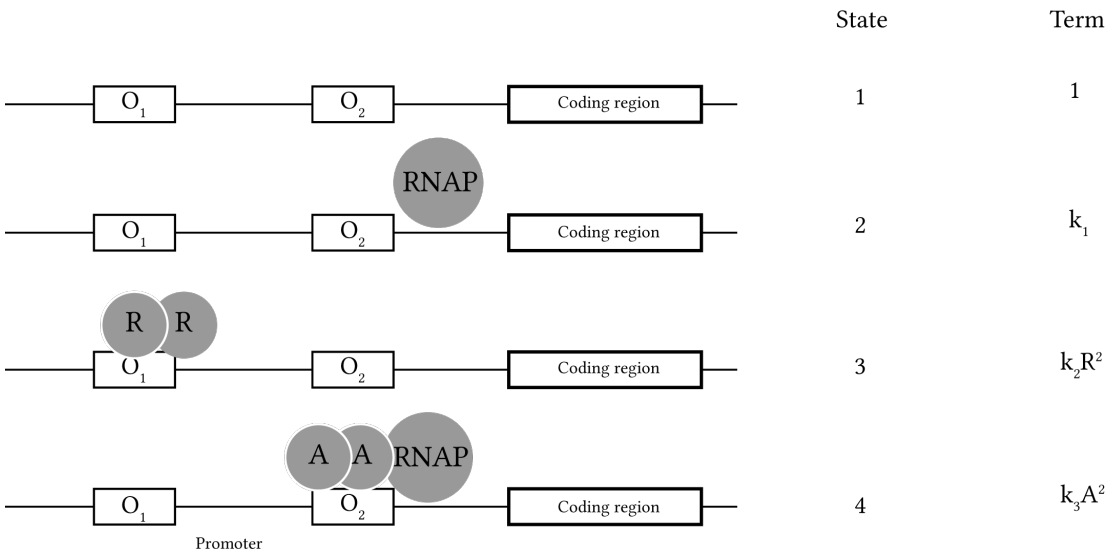

Figure 2.3 An example of a promoter regulated by a repressor (R) and an activator (A) modelled using the Shea-Ackers formalism. Figure adapted from Woods et al. (2016)

2.2.3 Simulation of deterministic dynamical systems

Deterministic modelling utilises Ordinary differential equation (ODE)s and models the concentrations of the species (proteins or other molecules) by time-dependent variables (de Jong 2002). Rate equations are used to model gene regulation where the rate of production of a species is a function of the concentrations of the other species (de Jong 2002).

2.2.3.1 Deterministic mass action kinetics

ODEs are used to represent the quantitative dynamics of a biochemical network. The ODEs describing a system can be derived from the coupled chemical reactions describing the system as well as their associated rates. This will be illustrated using a simple example, the Lotka-Voltera predator-prey model (Lotka 1925). This system describes the dynamics between two interacting species, a predator and a prey. The chemical reactions describing the system are given in Table 2.3. The rates of the system are organised in vector form,

$$h = \begin{pmatrix} k_1 x \\ k_2 xy \\ k_3 y \end{pmatrix}.$$

Table 2.3 Predator-prey chemical reactions

Name	Reaction	Rate
prey birth	$x \xrightarrow{k_1} 2x$	$k_1 x$
predation	$x + y \xrightarrow{k_2} 2y$	$k_2 xy$
predator death	$y \xrightarrow{k_3} \emptyset$	$k_3 y$

The stoichiometry matrix of the system is an $m \times n$ matrix, where m is the number of species and n the number of reactions and it summarises the stoichiometries of the system,

$$S = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix}. \tag{2.3}$$

The ODEs can then be constructed by multiplying the stoichiometry matrix S by the matrix containing the rates h. Therefore

$$s(t) = \frac{d}{dt}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix}\begin{pmatrix} k_1[x] \\ k_2[x][y] \\ k_3[y] \end{pmatrix}, \tag{2.4}$$

and thus we get the two ODEs describing the system as

$$\frac{dx}{dt} = k_1 x - k_2 xy \tag{2.5}$$

$$\frac{dy}{dt} = k_2 xy - k_3 y. \tag{2.6}$$

These differential equations can be simulated numerically over time using software packages like Mathematica (Mathematica 2016) and Python.

2.2.3.2 Assumptions of deterministic modelling

Two key assumptions are made when modelling a biochemical system using ODEs. Firstly, the species present in the system are measured continuously rather than discretely. This means that the species are measured in concentration over time and not the number of molecules over time. This assumption requires a large number of molecules to be present in order to be met (Ingalls & Iglesias 2010). The second assumption made is that the reactants are in a well-mixed solution. This means that the species in the system can interact each other constantly and freely.

2.2.4 Nonlinear dynamical modelling

2.2.4.1 Phase plane analysis

An alternative to studying the trajectory of a dynamical system over time is to study its behaviour in the phase plane. During a phase plane analysis the dependent variables x and y are plotted against each other. An example of a phase plane analysis of the predator-prey model given in Equations 2.6 is shown in Figure 2.4.

2.2.4.2 Steady states

For a system s, any point satisfying $\frac{d}{dt}s(t) = 0$ is considered a fixed point, or steady state. At that point the dynamics of the system are considered in equilibrium and will not change with increasing time. Using the example of the predator-prey system, a steady state exists when the system of Equations 2.6 are equal to 0:

$$\frac{dx}{dt} = f_x(x, y) = k_1 x - k_2 xy = 0 \tag{2.7}$$

$$\frac{dy}{dt} = f_y(x, y) = k_2 xy - k_3 y = 0 \tag{2.8}$$

By solving this system of equations, we get two steady states. One when $x = y = 0$ and one when $x = \frac{k_3}{k_2}$ and $y = \frac{k_1}{k_2}$. The stability of each steady state can then be determined.

2.2.4.3 Steady state stability

A stable steady state is defined as a fixed point whose nearby points approach the fixed point (Kaplan & Glass 1995). This means that after a small perturbation the system will quickly return to the steady state. An unstable steady state is one which if the system is perturbed slightly then it moves away from the steady state (Konopka 2007). The stability of the fixed points can be determined by the sign of the eigenvalues of the Jacobian matrix at each point. The Jacobian matrix is given by

$$\mathrm{J} = \begin{pmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} \end{pmatrix} \tag{2.9}$$

Using the predator-prey system as an example, the Jacobian matrix is given by

$$\mathrm{J} = \begin{pmatrix} k_1 - k_2 y & -k_2 x \\ k_2 y & k_2 x - k_3 \end{pmatrix} \tag{2.10}$$

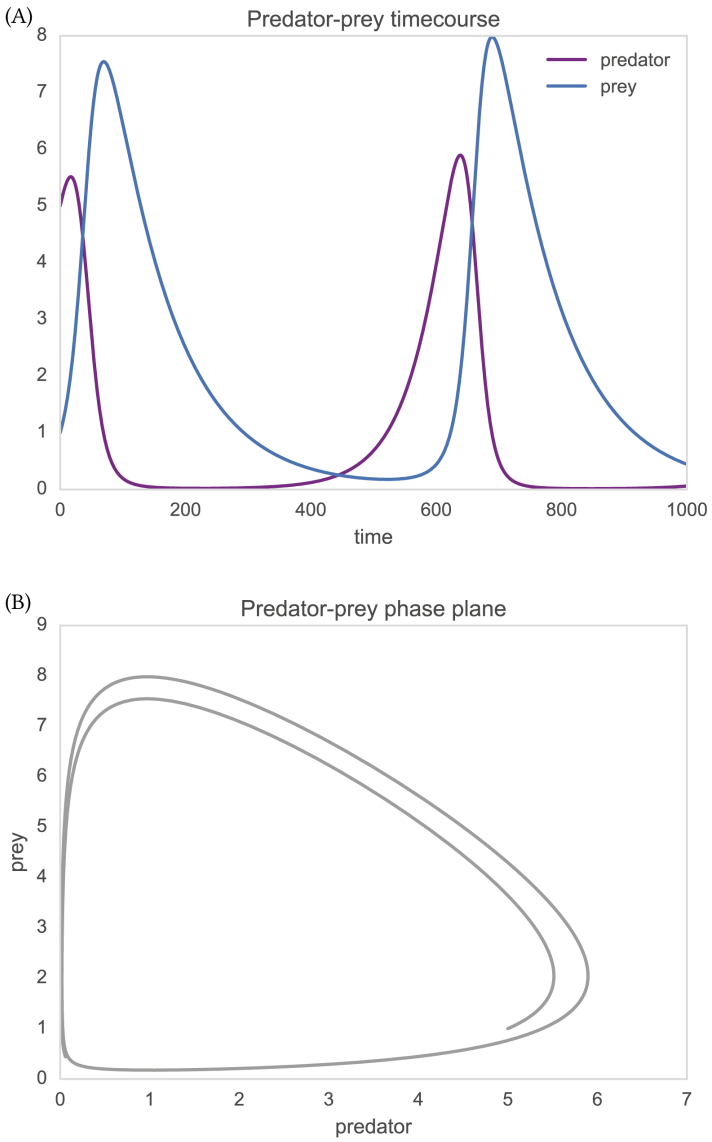

Figure 2.4 The predator-prey system is defined by Equations 2.6. (A) The trajectory over time (B) Phase plane plot of the predator-prey system of equations. The parameters used here are $k_1 = 2$, $k_2 = 1$ and $k_3 = 1$.

The eigenvalues λ are given by

$$\det(\mathbf{J} - \lambda\mathbf{I}), \tag{2.11}$$

where det is the determinant and $\mathbf{I}$ the identity matrix. If both eigenvalues are real and negative or imaginary with a negative real part then the steady state is stable.

If both eigenvalues have a positive real part then the steady state is unstable and if one has a positive and one has a negative fixed part the steady state is an unstable saddle node. If both eigenvalues are purely imaginary, the system oscillates around the fixed point. Solving the above for the fixed points in the predator-prey system, we find one stable steady state and one oscillatory fixed point.

2.2.4.4 Bifurcation analysis

A bifurcation analysis is carried out in order to study the effect that parameters have on the dynamical behaviour of the system (Strogatz 1994). In order to create a bifurcation diagram, all the parameters in the system remain constant while the value of one parameter is varied. We can then observe any changes in the number and stability of the steady states of the system, for example, whether a stable equilibrium in the system becomes unstable. The point where a major change occurs in the steady states of the system is called a bifurcation point (Ingalls & Iglesias 2010). The stability of the steady state, as well as its position, is depicted on a bifurcation diagram. By convention, the unstable branches are denoted by a dashed line and the stable branches by a solid line (Strogatz 1994).

One example of a bifurcation is the saddle-node bifurcation. This occurs when two stable states come closer together until they collide and destroy each other (Strogatz 1994). This can be illustrated using a simple example from Strogatz (1994). Consider the following system

$$\frac{dx}{dt} = r + x^2.$$

A bifurcation diagram of the above can be constructed by varying the value of parameter r. This gives the bifurcation diagram shown in Figure 2.5. This system has two steady states when r <0, one unstable steady state and one stable. When $r = 0$ these collapse into one steady state which then disappears when r >0.

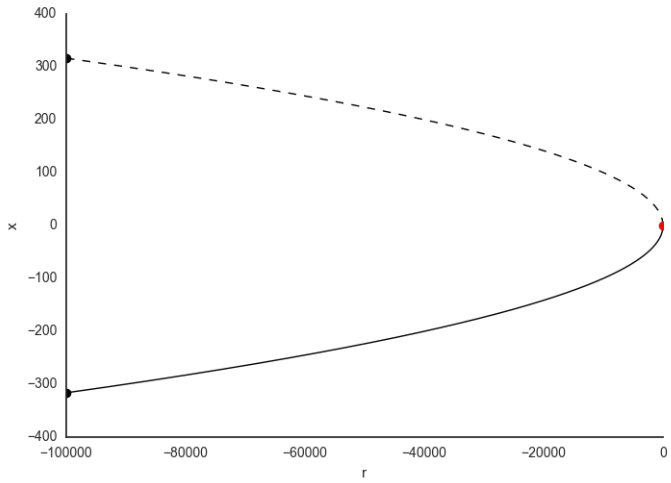

Figure 2.5 An example of a saddle-node bifurcation diagram. Figure adapted from Strogatz (1994).

2.2.5 Stochastic modelling of dynamical systems

The assumptions that have to be made to model a system deterministically cannot always be met. This can occur when the molecule numbers in the system are low. When this is the case, stochastic dynamics are more appropriate to model the dynamical system. In stochastic modelling species are measured in discrete amounts rather than concentrations and a joint probability distribution is used to express the probability that at time t the cell contains a number of molecules of each species (de Jong 2002; Khammash 2010). It takes probabilistic effects into account.

Biological processes are well known to include randomness. The source of this randomness originates from the random collisions between molecules that govern biological reactions (Khammash 2010). This randomness affects downstream events and the phenotypic behaviour of cells. This is known as cellular noise and it is known to be key for various cellular processes (Eldar & Elowitz 2010). Cellular noise can be classified into two categories, intrinsic noise and extrinsic noise. Intrinsic noise originates from the inherently random collisions between the species of the system under consideration. Extrinsic noise originates from fluctuations in the environment within which the system of interest resides, like the number of available RNA polymerases or other protein numbers (Khammash 2010). The noisiness of biological processes often makes stochastic dynamics more appropriate for

modelling cellular systems.

2.2.5.1 Simulating stochastic models

Stochastic models are often analytically intractable but can be studied using numerical simulation. A well known algorithm for the simulation of such models is the Direct method proposed by Gillespie (1977).

2.2.5.2 The Gillespie algorithm

In stochastic systems, the Gillespie algorithm is widely used to simulate the time-evolution of the state of the system (Wilkinson 2006). The algorithm, developed by Gillespie (1977) can be summarised in four steps:

1. Initialise time t and number of species s and state of system x
2. Draw a sample time step τ from the (exponential) distribution of time T
3. Draw a sample reaction from all reactions R
4. Update time by $t = t + \tau$ and state of system by $x = x + s_\mu$
5. Repeat from Step 2 until total simulation time reached

This algorithm results in one trajectory of the system. It has to be repeated a number of times to obtain enough realisations of the trajectory to compute appropriate summary statistics.

2.2.5.3 Stochastic mass action kinetics

Here I will consider the predator-prey system introduced in Section 2.2.3.1. A set of reactions is defined, as shown in Table 2.3, and each one has an associated stochastic rate constant c_i. The rate constant, or hazard function, of each reaction i is defined as $h_i(x, c_i)$, where x is the current state of each species in the system. The form of each hazard function is defined by the order of the given reaction (Wilkinson 2006), as shown in Table 2.4. When simulating a stochastic system using the Gillespie algorithm, the state of the system x is defined as the sum of all the reaction hazards, namely $h_0(x, c) = \sum_{i=1}^{n} h_i(x, c_i)$ (Wilkinson 2006).

Table 2.4 Defining reaction hazards

Order	Reaction	Hazard
Zeroth	$\emptyset \xrightarrow{c_i} X$	$h_i(x, c_i) = c_i$
First	$X_a \xrightarrow{c_i} ?$	$h_i(x, c_i) = c_i x_a$
Second	$X_a + X_b \xrightarrow{c_i} ?$	$h_i(x, c_i) = c_i x_a x_b$
Dimerization	$X_a + X_a \xrightarrow{c_i} X_{2a}$	$h_i(x, c_i) = c_i \frac{x_a(x_a-1)}{2}$

2.3 The Bayesian approach to parameter inference and system design

The parameters of a model represent the biochemical rates that are involved in the system under study, like degradation rates, transcription rates and polymerization rates. These rates cannot often be measured *in vitro* and taking generalised estimates from existing literature can be inaccurate. In order to make useful predictions about the biological system under consideration, the model parameters must be estimated (Zheng & Sriram 2010).

To address this challenge, statistical optimisation methods have been developed. These methods aim to infer the parameters of the model that can give rise to some experimentally observed behaviour. Parameter inference methods have the same general structure: there is a cost function that compares the model data to the experimental data and an optimisation function that aims to optimize the cost function (Toni 2010). There is a wide range of such optimisation algorithms that can be used like gradient descent (Levenberg 1944; Marquardt 1963), simulated annealing (Kirkpatrick, C D Gelatt, & Vecchi 1983) and evolutionary algorithms (Onbaşoğlu & Özdamar 2001; Wood, Alexander, & Bulger 2002).

Bayesian approaches to parameter inference have been shown to work well in biological problems (Barnes et al. 2011; Toni 2010; Liepe et al. 2014). Bayesian approaches to parameter inference have the advantage of offering a range of values that give rise to the data, rather than point estimates. In Bayesian approaches the aim is to obtain the posterior distribution, which is dependent on the prior distribution, the prior knowledge about the system, and the likelihood, which is obtained from the data. At the core of Bayesian statistics lies Bayes' rule which states that, for a set of data x and a model with a set of parameters θ:

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)} \propto p(x|\theta)p(\theta), \tag{2.12}$$

where $p(x|\theta)$ is the likelihood, and $p(\theta)$ is the prior. In continuous problems, Equation 2.12 becomes

$$p(\theta|x) = \frac{p(\theta)p(x|\theta)}{\int p(x|\theta)p(\theta)d\theta} \tag{2.13}$$

where $\int p(x|\theta)p(\theta)d\theta$ is the evidence. It is often not possible to obtain analytical expressions of the posterior, but methods to approximate it numerically have been developed (Barnes, Silk, & Stumpf 2011). One such class of algorithms are the Markov Chain Monte Carlo methods (Gilks, Richardson, & Spiegelhalter 1996). These are described in more detail in Section 2.3.1.

Parameter estimation problems typically involve a set of observed data and a mathematical model describing the biological system. Oftentimes there are a number of competing models under consideration. The challenge then is to fit the model, and model parameters in order to reconstruct the observed data (Ma et al. 2009). In system design a different but related problem must be addressed; Instead of experimental data the researcher has an idea of what the system output should be (Barnes, Silk, & Stumpf 2011). A set of carefully selected design objectives can then be used as substitute data (x_s), representing data one would like to observe, in the Bayesian inference problem (Barnes, Silk, & Stumpf 2011). This method has been successfully applied in synthetic biology system design (Barnes et al. 2011; Woods et al. 2016).

2.3.1 Approximate Bayesian Computation (ABC)

ABC methods are used for inferring the posterior distribution in cases where it is too computationally expensive to evaluate the likelihood function. Instead of calculating the likelihood, ABC methods simulate the data and then compare the simulated and observed data through a distance function (Toni et al. 2009). Given the prior distribution $p(\theta)$ we can approximate the posterior distribution, $p(\theta \mid x) \propto p(x \mid \theta)p(\theta)$, where $p(x \mid \theta)$ is the likelihood of a parameter, θ, given the data, x. There are a number of different variations of the ABC algorithm depending on how the approximate posterior distribution is sampled.

The simplest ABC algorithm is the ABC rejection sampler (Pritchard et al. 1999). In this method, parameters are sampled from the prior and data simulated through

the data generating model. For each simulated data set, a distance from that of the data is calculated, and if greater than a threshold, ε, the sample is rejected, otherwise it is accepted.

Algorithm 1 ABC rejection algorithm

1: Sample a parameter vector θ from prior $p(\theta)$
2: Simulate the model given θ
3: Compare the simulated data with the desired data, using a distance function d and tolerance ε. if $d \leq \varepsilon$, accept θ

The main disadvantage of this method is that if the prior distribution is very different from the posterior, the acceptance rate is very low (Toni et al. 2009). An alternative method is the ABC Markov Chain Monte Carlo (MCMC) developed by Marjoram et al. (2003). The disadvantage of this method is that if it gets stuck in an area of low probability it can be very slow to converge (Sisson, Fan, & Tanaka 2007).

An alternative method developed by Toni et al. (2009) takes advantage of Sequential Monte Carlo, and avoids issues faced by the rejection and MCMC methods. It propagates the prior through a series of intermediate distributions in order to arrive at an approximation of the posterior. The tolerance, ε, for the distance of the simulated data to the desired data is made smaller at each iteration. When ε is sufficiently small, the result will approximate the posterior distribution (Toni et al. 2009).

ABC SMC can identify the parameter values within a predefined range of values that can give rise to the data. It works by first sampling at random from the initial range set by the researcher, i.e. from the prior distribution of values. Each sample is called a particle. It then simulates the model given those values and compares that to the target behaviour. If the distance between the simulation and the target behaviour is greater than a predefined threshold distance ε, then the parameter values that produced that simulation are rejected. This is repeated for a predefined number of samples which are collectively referred to as a population. Each particle in a population has a weight associated with it, which represents the probability of it producing the desired behaviour. At subsequent iterations, the new samples are obtained from the previous populations and the ε is set to a smaller value, thus eventually reaching the desired behaviour. The algorithm proceeds as follows:

Algorithm 2 ABC SMC algorithm

1: Select ε and set population t = 0
2: Sample particles (θ). If t = 0, sample from prior distributions (p). If t > 0, sample particles from the previous population to obtain θ*.
3: If t > 0: Perturb each particle θ* using perturbation kernel K_t to obtain perturbed particle θ**
4: Simulate each particle to obtain time course.
5: Reject particles if d > ε.
6: Calculate the weight w for each accepted particle. At the first population assign a weight equal to 1 for all particles. In subsequent populations the weight of a particle is equal to the probability of observing that particle divided by the sum of the probabilities of the particle arising from each of the particles in the previous population:
7: $$w_t^{(i)} = \begin{cases} 1, & \text{if } t = 0 \\ \frac{p(\theta_t^{(i)})}{\sum_{j=1}^{N} w_{t-1}^{(j)} K_t(\theta_{t-1}^{(j)}, \theta_t^{(i)})}, & \text{if } t>0. \end{cases}$$

Details about each module of the ABC SMC algorithm are given in the sections below.

2.3.1.1 Particle sampling

For the first population, particles are sampled from the prior, which consists of the boundaries of a distribution for each parameter defined by the user based on biochemical knowledge or literature. For subsequent populations, particles are sampled from the previous population. The weight of each particle in the previous population dictates the probability of it being sampled. The number of samples to be drawn is specified by the user.

2.3.1.2 Perturbation

Each sampled particle is perturbed by a kernel defined by the distribution of the previous population, as developed by Toni et al. (2009).

$$K_t(\theta|\theta^*) = \theta^* + U(+s_p, -s_p), \tag{2.14}$$

where

$$s_p = \frac{1}{2}\big(max(\theta_{p-1}) - min(\theta_{p-1})\big) \tag{2.15}$$

If the θ** falls out of the limits of the priors then the perturbation is rejected and repeated until an acceptable θ** is obtained. This method is successful in perturbing

the particles by a small amount in order to explore the parameter space, but can be slow to complete.

2.3.1.3 Epsilon schedule

The algorithm uses an automated epsilon schedule, where the threshold of the next iteration is chosen from the range of values of the current population. This method is the quantile method. Another approach to the epsilon schedule is to use an adaptive epsilon schedule which is efficient in avoiding local minima (Silk, Filippi, & Stumpf 2013). Throughout this thesis, the quantile method was used with a tight quantile (0.3) to avoid the problem of local minima.

2.3.1.4 Particle simulation

Each particle is simulated using cuda-sim (Zhou et al. 2011). The model is provided by the user in SBML format and is converted into CUDA® code by cuda-sim. The model in CUDA® code format can then be run on NVIDIA® GPUs. This allows the user to take advantage of the speed of parallelised simulations without any CUDA® knowledge.

2.3.1.5 Weight calculation

For the first population the weights are all given a value of 1, and then normalised over the number of particles. For subsequent populations, the weights of the particles are calculated by considering the weights of the previous population (Toni et al. 2009). The weights are then normalised over the total number of particles.

$$w_t^{(i)} = \frac{p(\theta_t^{(i)})}{\sum_{j=1}^{N} w_{t-1}^{(j)} K_t(\theta_{t-1}^{(j)}, \theta_t^{(i)})} \text{ for n} > 0. \tag{2.16}$$

2.3.1.6 ABC SMC algorithm example

This algorithm is implemented on a simple example for illustration. A simple model was used, consisting of one species, A converting to another, B. The model is described by two differential equations, where A is the reactant and B the product, produced at a rate p.

$$\frac{d[B]}{dt} = p[A] \tag{2.17}$$

$$\frac{d[A]}{dt} = -p[A]. \tag{2.18}$$

The priors were set to $p \sim U(0, 10) nMs^{-1}$. Initial conditions for A and B were set to 1 and 0 respectively. The data to which the model was compared to was generated by simulating the same model with the parameter set to 1, as shown in Figure 2.6.

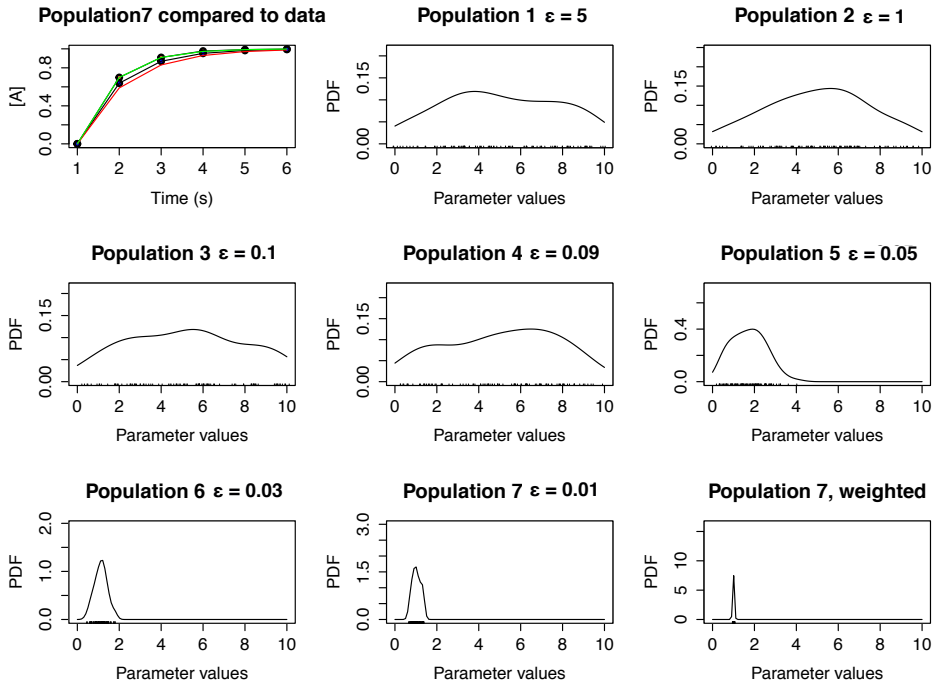

Figure 2.6 ABC SMC parameter inference. The true parameter value is equal to 1 and its time course is shown in red in the top left panel. The blue time course is that of the final population, green is the upper quartile and red is the lower quartile range of values. The progress of the selection process can be seen the as the ε schedule proceeds from the top left to the bottom right. The bottom far right panel is a weighted density plot of the posterior distribution of p at ε = 0.01.

Figure 2.6 illustrates the use of ABC SMC, using a simple example. During the course of 7 populations, the accepted distance ε of the simulated particles to the data is incrementally decreased. This leads to a final population where the distance of the data to the particles is very small, and there is a good agreement between the two. The algorithm concludes with a set of parameter values that produced this behaviour, which approximate the posterior distribution. The posterior distribution

found in this model is in good agreement with the parameter value used to generate the data.

2.3.1.7 Visualising posterior distributions

The posterior distribution has as many dimensions are there are parameters, thus can be challenging to visualise for models containing more than two parameters. In order to visualise the multi-dimensional posterior distributions in this thesis, the one and two-dimensional marginal distributions of the parameters will be shown. An example of such a plot is shown in Figure 2.7. The data shown in Figure 2.7 consists of 10000 random samples drawn from a bivariate normal distribution, of mean = 0 for all dimensions and variance σ

$$\sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}.$$

The two-dimensional distribution is plotted as shown in Figure 2.7.

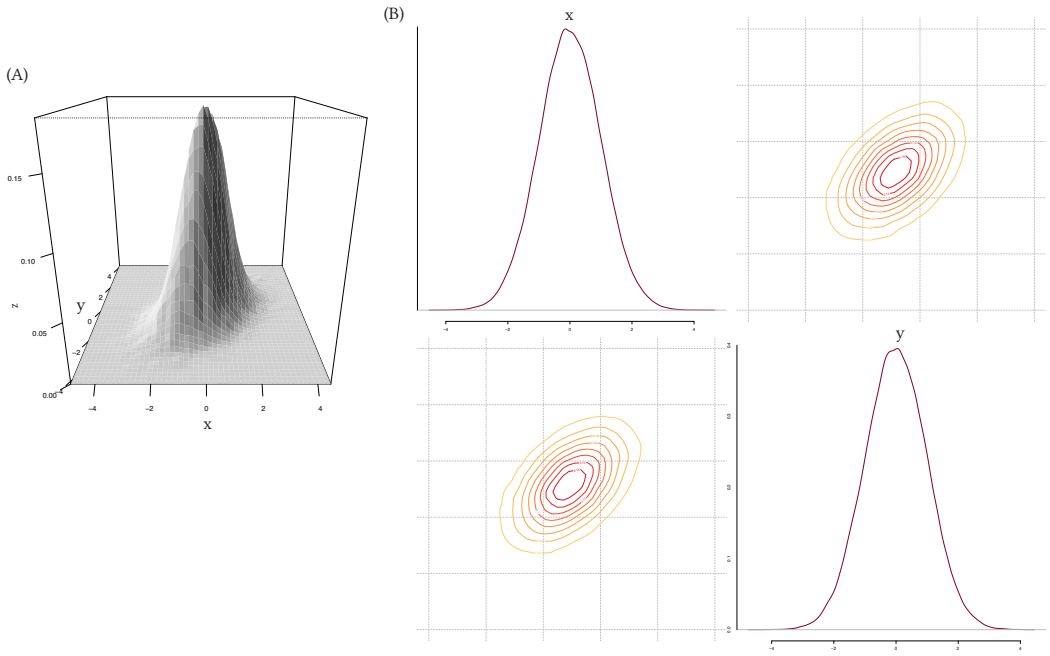

Figure 2.7 Visualising a two-dimensional distribution. (A) A bivariate normal distribution plotted in 3D. (B) A bivariate normal can be visualised by plotting the one-dimensional marginal distributions on the diagonal and the two-dimensional marginal distributions are on the off-diagonal. Correlation can be visualised using the 2D marginal distributions.

2.3.2 Derivation of model parametric robustness defined via Bayesian statistics

During this thesis I use the term robustness in its parametric meaning, i.e. as the ability of a system to retain its function despite parameter perturbations (Stelling et al. 2004). The robustness of biological systems has been studied extensively (Barkai & Leibler 1997; Stelling et al. 2004; Prill, Iglesias, & Levchenko 2005; Kim et al. 2006; Kitano 2007; Hafner et al. 2009; Shinar & Feinberg 2010; Zamora-Sillero et al. 2011; Woods et al. 2016). Below I show that the robustness of a model can be calculated by dividing the volume of its functional region by the volume of its priors. Starting with Bayes' rule that:

$$f(\theta|x) = \frac{f(\theta)f(x|\theta)}{\int p(x|\theta)p(\theta)d\theta}, \tag{2.19}$$

where x is the data, $p(x|\theta)$ is the likelihood, $p(\theta)$ is the prior, and $\int p(x|\theta)p(\theta)d\theta$ is the evidence. The evidence is the normalisation constant so that the distribution integrates to 1. For a given model design D_1 and objective O we define the functional region F as the region within the prior where O is satisfied. So within the prior we can assign 1 to any region that falls within F and 0 to any region outside that.

$$p(O|D_1) = \int p(O|\theta, D_1)p(\theta|D_1)d\theta, \tag{2.20}$$

for a design with three parameters this becomes

$$p(O|D_1) = \iiint_{\underline{\theta}} p(O|\underline{\theta})p(\underline{\theta}|D_1)d\underline{\theta}, \tag{2.21}$$

where $\underline{\theta}$ is a vector containing the three parameters $= (\theta_1, \theta_2, \theta_3)$, and each $\theta_i \in \mathbb{R}$. To calculate the robustness, or model evidence, we integrate this with respect to $\underline{\theta}$. We assume all parameters $\theta_1, \theta_2, \theta_3$ have uniform prior, $p(\underline{\theta}|D_1) \sim U(a, b)$. If we assume a = 0 this integral becomes:

$$p(O|D_1) = \iiint_{\underline{\theta}} p(O|\underline{\theta})\frac{1}{b_1}\frac{1}{b_2}\frac{1}{b_3}d\underline{\Theta}\text{ , and} \tag{2.22}$$

$$p(O|D_1) = \frac{1}{b_1}\frac{1}{b_2}\frac{1}{b_3}\iiint_{\underline{\theta}} p(O|\underline{\theta})d\underline{\theta}, \tag{2.23}$$

since $\frac{1}{b_1}\frac{1}{b_2}\frac{1}{b_3}$ is a constant. Then assuming that the likelihood is uniform Equation 2.23 becomes

$$p(O|D_1) = \frac{1}{b_1}\frac{1}{b_2}\frac{1}{b_3}\left[\iiint_{\underline{\theta}_F} 1 d\underline{\theta} + \cancelto{0}{\iiint_{\underline{\theta}_{\bar{F}}} 0 d\underline{\theta}}\right], \tag{2.24}$$

since we assign 1 to any region within F and 0 to any region outside it. This becomes:

$$p(O|D_1) = \overbrace{\frac{1}{b_1}\frac{1}{b_2}\frac{1}{b_3}}^{|P|} \underbrace{\iiint_{\underline{\theta}_F} 1 d\underline{\theta}}_{|F|}, \tag{2.25}$$

$$\therefore p(O|D_1) = \frac{|F|}{|P|}, \tag{2.26}$$

where |P| is the volume of the prior P and |F| the volume of the functional region F. Therefore, in the case where both the prior and the likelihood are uniform, the robustness R of the design is the ratio of the volumes of the two. If on the other hand we assume the likelihood is multivariate normal, with priors remaining uniform, Equation 2.23 becomes:

$$p(O|D_1) = \frac{1}{|P|}\iiint_{\underline{\theta}} f(\underline{\theta}; \mu, \Sigma) d\underline{\theta} \tag{2.27}$$

$$\therefore p(O|D_1) = \frac{1}{|P|}\underbrace{\frac{2\pi^{\frac{k}{2}}}{k\Gamma(\frac{k}{2})}\left[\chi^2_k(\alpha)\right]^{\frac{k}{2}}|\Sigma|^{\frac{1}{2}}}_{\text{The volume of an ellipsoid}} \tag{2.28}$$

$$\therefore p(O|D_1) = \frac{|F|}{|P|}. \tag{2.29}$$

We can use the Bayes factor in order to compare the robustness between two model designs. The Bayes factor is used to determine which model, M_a or M_b, can explain the data X better and is defined as follows:

$$B_{12} = \frac{p(X|M_a)}{p(X|M_b)}, \tag{2.30}$$

which represents the fraction of the evidence supported by model a over the evidence supported by model b. The evidence against model M_b, and thus in favour of M_a, the Bayes factor, can be interpreted as shown in Table 2.5. For a comprehensive review on the use of Bayes factors see Kass & Raftery (1995).

Table 2.5 Bayes factor evidence interpretation. Modified from Kass & Raftery (1995)

Bayes factor	Evidence
1 to 3.2	Not significant
3.2 to 10	Substantial
10 to 100	Strong
>100	Decisive

In a system design model, the Bayes factor is used to determine which design, D_1 or D_2 can fulfil the design objective O better. Therefore,

$$B_{12} = \frac{\int p(x|\theta, D_1)p(\theta, D_1)d\theta}{\int p(x|\theta, D_2)p(\theta, D_2)d\theta} = \frac{p(O|D_1)}{p(O|D_2)} \tag{2.31}$$

$$\therefore B_{12} = \frac{|F_1|}{|P_1|} / \frac{|F_2|}{|P_2|}. \tag{2.32}$$

We can thus use the ratio of the two robustness measures to calculate the Bayes factor. If two models have a different number of parameters, the robustness of the system will only increase if |F| increases by more than the proportion by which |P| increased (Woods et al. 2016). A model is penalised for an additional parameter if it does not increase the volume of the functional region by more than the volume that the added parameter added to the prior. This is also true for nested models, where one model is wholly contained in the other.

3 Positive feedback loops can increase the robustness of a genetic toggle switch

3.1 Introduction

In this chapter, I examine whether adding feedback loops to the genetic toggle switch increases its robustness to parameter fluctuations. To do this, I use ABC SMC to estimate the parameter values that allow the toggle switch model to behave like a switch. I then study the effect that adding feedback loops has to the toggle switch bistability, and finally I use model selection to select the most robust switch model out of the ones considered.

Structurally this chapter is organised as follows: In the first section, I examine the genetic toggle switch with no added feedback loops. I use a parameter scan to find the parameter values that make it bistable and then use ABC SMC for parameter inference for a switch-like behaviour. In the subsequent section, I examine the effect that the addition of feedback loops to the genetic toggle switch has on its stability, and select the switch architectures that are capable of bistable behaviour. Finally, I use ABC-SysBio model selection to select the most robust model out of the bistable switches.

3.2 Motivation

Creating synthetic devices that are robust to changing cellular contexts will be key to the success of synthetic biology. Unknown initial conditions and parameter values as well as the variability of the cellular environment, extracellular noise and crosstalk make the majority of synthetic genetic devices non-functional (Chen,

Chang, & Lee 2009). Designing devices robust to this environment will lead to reliable behaviour of the systems.

When faced with a set of competing designs for a given genetic circuit, one is likely to choose the simplest possible model that can achieve the desired behaviour. However, simple systems are often the least robust. Feedback loops are well known key regulatory motifs (Brandman et al. 2005). Negative feedback loops are essential for homoeostasis and buffering, thus increasing robustness to extrinsic noise sources, and positive feedback loops can generate multistationarity in a system (Thomas, Thieffry, & Kaufman 1995). Incorporating this kind of additional feedback interactions can make a design more robust and reliable.

One of the first computational approaches for the tuning of robust synthetic networks was that of Batt et al. (2007) where they examined the problem of finding a subset of the parameter space for which a given property was satisfied for all the parameters. Chen, Chang, & Lee (2009) used the fuzzy dynamic game method to solve the minimax regulation design problem of synthetic genetic networks. In that method, the worst case effect of all disturbances is minimised for a given network. An evolutionary algorithm has also been used to solve the robust design problem by evolving the parameters of the system in order to make it more robust to cellular disturbances by Wu, Lee, & Chen (2011). Here I use Bayesian model selection to examine the system structure in addition to the system parameters being adjusted to select a system that can robustly create the desired behaviour.

3.3 The bistable genetic toggle switch

The synthetic genetic toggle switch consists of two mutually repressing transcription factors. It was first developed by Gardner, Cantor, & Collins (2000), and consists of the following ODEs:

$$\frac{du}{dt} = \frac{a_1}{1 + v^{\beta}} - u \tag{3.1}$$

$$\frac{dv}{dt} = \frac{a_2}{1 + u^{\gamma}} - v, \tag{3.2}$$

where u is the concentration of repressor 1, v the concentration of repressor 2, a_1 and a_2 denote the effective rates of synthesis of repressors 1 and 2 respectively, β is the cooperativity of repression of promoter 1 and γ of repressor 2. This model is capable of bistable behaviour when a_1 and a_2 are balanced and when β, γ are > 1 (Gardner, Cantor, & Collins 2000). This model is derived from the biochemical rate

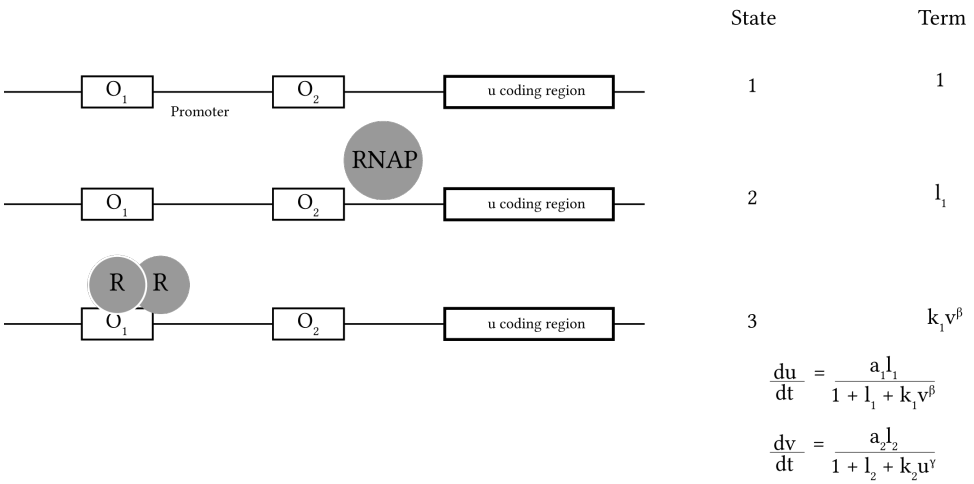

Figure 3.1 Simple toggle switch model using the Shea-Ackers formalism

equations of gene expression for the two promoters in the system. Using the Shea-Ackers formalism (as described in Section 2.2.2.2), these are shown in Figure 3.1. The model can thus also be described by the following ODEs:

$$\frac{du}{dt} = \frac{a_1 l_1}{1 + l_1 + k_1 v^{\beta}} - u \tag{3.3}$$

$$\frac{dv}{dt} = \frac{a_2 l_2}{1 + l_2 + k_2 u^{\gamma}} - v, \tag{3.4}$$

The model shown in Equations 3.1-3.2 is the dimensionless version of the model shown in Equations 3.3-3.4. This is constructed by measuring a_1 and a_2 in units of $k_1^{\frac{-1}{\beta}}$ and $k_2^{\frac{-1}{\gamma}}$ respectively (Phillips et al. 2013) and setting leakiness to zero. In Section 3.3.2 I will use the more realistic Shea-Ackers version of the model to show that it is also capable of bistable behaviour.

3.3.1 The quasi steady state approximation and the genetic toggle switch

In order for a system to be able to be studied mathematically, a number of assumptions have to be made. The system under consideration has to be reduced to very few equations and parameters in order to make the system solvable. This requires assumptions to be made about the system that cannot always be justified, such as the quasi-steady state approximation (QSSA). The QSSA assumes that the binding/unbinding processes are much faster than any other process (Loinger et al. 2007) thus

the bound intermediate is assumed to always be in steady state. The QSSA assumption is met *in vitro* but often does not hold *in vivo*. Its misuse can lead to large errors and incorrectly estimated parameters (Pedersen, Bersani, & Bersani 2007).

Equations 3.1-3.2 of the genetic toggle switch can be derived from the full model by using the quasi-steady state approximation (QSSA). In this section I will discuss how Equations 3.1-3.2 can be derived from the full model constructed under the mass action formalism by using the QSSA. Consider the set of reactions given in Table 3.1 representing the genetic toggle switch.

Table 3.1 Toggle switch model reactions under mass action kinetics

Equation	Description
$gu \xrightarrow{a_1} gv + u$	gene expression
$gv \xrightarrow{a_2} gu + v$	
$\beta u \underset{\mathrm{Km_u}}{\overset{\mathrm{Kd_u}}{\rightleftharpoons}} u_\beta$	dimerization
$\gamma v \underset{Kd_v}{\overset{Km_v}{\rightleftharpoons}} v_\gamma$	
$gu + v_\gamma \underset{Kf_u}{\overset{Kr_u}{\rightleftharpoons}} v_\gamma \bullet gu$	repression
$gv + u_\beta \underset{Kf_v}{\overset{Kr_v}{\rightleftharpoons}} u_\beta \bullet gv$	
$u \xrightarrow{D_u} \emptyset$	degradation
$v \xrightarrow{D_v} \emptyset$	

Using mass action kinetics, this set of reactions gives us the following ODEs:

$$\frac{du}{dt} = a_1 gu - Kd_u u^\beta + K_m u_\beta - D_u u \tag{3.5}$$

$$\frac{dv}{dt} = a_2 gv - Kd_v v^\gamma + K_m v_\gamma - D_v v \tag{3.6}$$

$$\frac{du_\beta}{dt} = Kd_u u^\beta - Km_u u_\beta - Kf_v gvu_\beta \tag{3.7}$$

$$\frac{dv_\gamma}{dt} = Kd_v v^\gamma - Km_v v_\gamma - Kf_u guv_\gamma \tag{3.8}$$

$$\frac{dgu \bullet v_\gamma}{dt} = Kf_u gvv_\gamma - Kr_u gu \bullet v_\gamma \tag{3.9}$$

$$\frac{dgv \bullet u_\beta}{dt} = Kf_v guu_\beta - Kr_v gv \bullet u_\beta. \tag{3.10}$$

The principal quasi steady state assumption being made is that the rate of binding and unbinding of the repressor to the promoter happens very fast. We assume that it

happens so much faster than any other reaction in the system that we can assume that $\frac{dgu \bullet v_\gamma}{dt}$ and $\frac{dgv \bullet u_\beta}{dt}$ are constant, i.e. in equilibrium. This assumption is also known as the separation of timescales in transcriptional regulation. In order for $\frac{dgu \bullet v_\gamma}{dt}$ and $\frac{dgv \bullet u_\beta}{dt}$ to be in equilibrium, we must assume that

$$\frac{dgu \bullet v_\gamma}{dt} = Kf_v gvv_\gamma - Kr_u gu \bullet v_\gamma = 0 \tag{3.11}$$

$$\frac{dgv \bullet u_\beta}{dt} = Kf_u guu_\beta - Kr_v gv \bullet u_\beta = 0, \tag{3.12}$$

therefore,

$$Kf_v gvv_\gamma - Kr_u gu \bullet v_\gamma = 0 \tag{3.13}$$

$$Kf_u guu_\beta - Kr_v gv \bullet u_\beta = 0. \tag{3.14}$$

Now we have a set of algebraic equations rather than differential equations. Solving for $\frac{dgu \bullet v_\gamma}{dt}$ and $\frac{dgv \bullet u_\beta}{dt}$ respectively we get:

$$gu \bullet v_\gamma = \frac{Kf_v gvv_\gamma}{Kr_v} \tag{3.15}$$

$$gv \bullet u_\beta = \frac{Kf_u guu_\beta}{Kr_u}. \tag{3.16}$$

This can now be substituted into the set of Equations 3.5-3.10. The second assumption that is made in this system is that the rate of formation and dissociation of the polymerised transcription factor is in steady state. This allows us to solve the system for u in a similar way as shown above and substitute it in Equations 3.5-3.10. This results in a simplified model, with fewer differential equations and parameters.

3.3.2 Phase space and bifurcation analysis

First, I study the model given in Equations 3.3-3.4 by conducting a bifurcation analysis in order to confirm that it is capable of bistable behaviour. A bifurcation analysis is used to determine the properties of a system in parameter space (Alon 2007). Here I used the PyDSTool (Clewley 2012), a python package used for the analysis of dynamical systems.

The parameters chosen here for the phase space analysis are within the range suggested by Gardner, Cantor, & Collins (2000). Parameters a_1 and a_2 are set to 10, and β, γ set to 2. A vector plot shows that the system has two steady states as shown

in Figure 3.2. Both states were found to be stable by examining the eigenvalues of the system at each steady state using Mathematica (Mathematica 2016).

I further study the system by conducting a bifurcation analysis, where all parameters remain constant to the values shown above, and only one parameter (a_1) is varied. The bifurcation diagram, given in Figure 3.2C shows a saddle-node bifurcation. We observe that by varying the parameter for the effective rate of synthesis of repressor 1 while all other parameters remain constant, the system is bistable when $7 \geq a_1 \leq 17$.

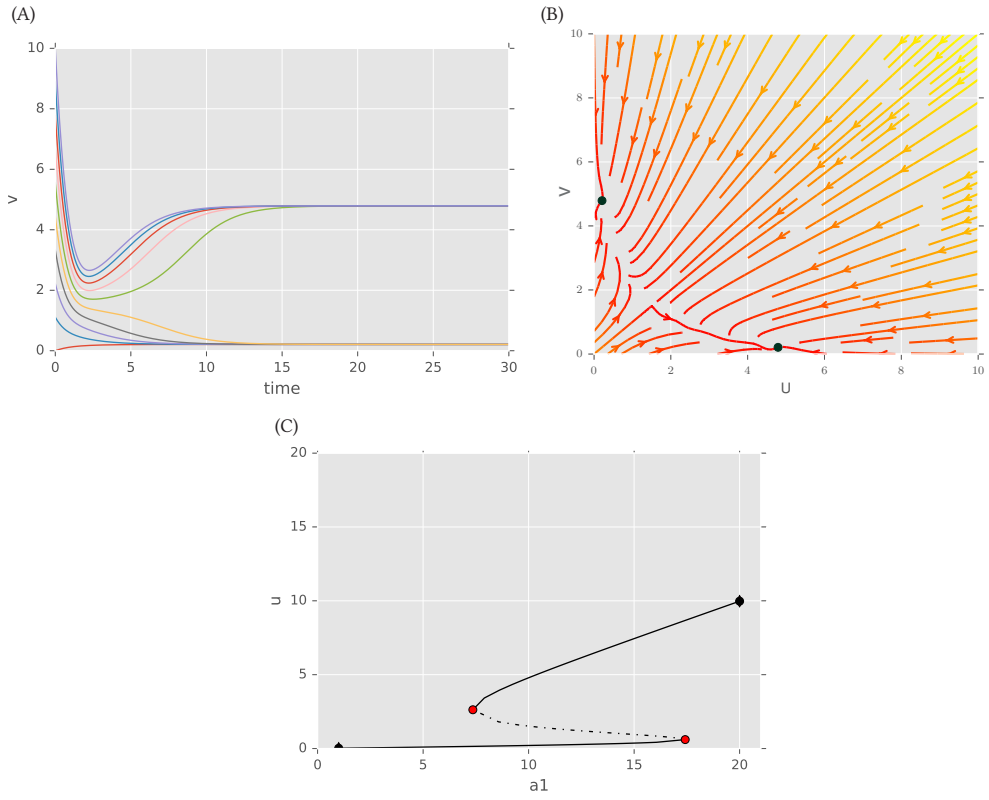

Figure 3.2 The Gardner, Cantor, & Collins (2000) toggle switch is capable of bistable behaviour given a_1 and a_2 = 10, and β, ɣ = 2. (A) The time course of the simulated model using multiple initial conditions for v. (B) The vector plot of the Gardner switch shows there are two stable steady states at (u, v) = (4.791, 0.208) and (u, v) = (0.208, 4.791). (C) A bifurcation diagram shows that the system is bistable when $7 \geq a_1 \leq 17$.

3.4 Designing a simple synthetic switch

It was demonstrated that the model used in Section 3.3.2 is bistable for the parameters given. Nevertheless, for a switch to be useful in synthetic biological applications, it must be capable of behaving like a switch over a range of parameter values rather than just point values, as these fluctuate within the cellular environment. Therefore, in this section, I study the parameter ranges that can give rise to a bistable switch. This indicates whether the bistability of the switch models is robust to small parameter fluctuations.

3.4.1 Development of the mass action model for the genetic toggle switch

In order to study the switch system in a more realistic way, I developed an extension to the Gardner, Cantor, & Collins (2000) switch. This new set of switches does not use the quasi-steady state approximation (QSSA), as described in Section 3.3.1. Avoiding the use of the QSSA the model becomes more flexible and does not assume that the formation of the polymerised transcription factors is in steady state or that the association and dissociation of the transcription factors to the promoter is in steady state.

Using mass action, this changes the two-equation system used in Equations 3.1-3.2 into a system of 18 equations. The equations describing the system are shown below and illustrated in Figure 3.3. The ODEs are given in Appendix A. The system consists of two genes, gA and gB. The products of the genes homodimerise and mutually repress each other. A symmetric model, where the parameters for equivalent reactions are set to be the same, was used for simplicity.

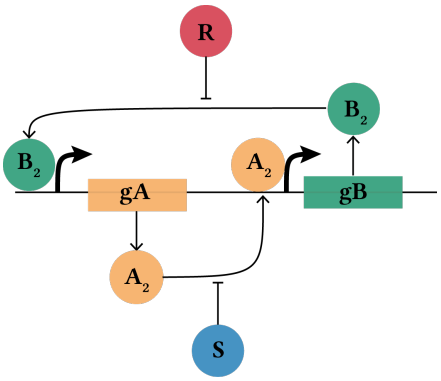

Figure 3.3 An illustration of the toggle switch model used in the parameter scan.

Table 3.2 Simple mass action switch reactions

Equation	Description
$gA \xrightarrow{ge} gA + A$ $gB \xrightarrow{ge} gB + B$	gene expression
$A + A \xrightarrow{dim} A_2$ $B + B \xrightarrow{dim} B_2$	dimerization
$A_2 \xrightarrow{dim\_r} A + A$ $B_2 \xrightarrow{dim\_r} B + B$	monomerization
$gA + B_2 \xrightarrow{rep} B_2gA$ $gB + A_2 \xrightarrow{rep} A_2gB$	repression
$B_2gA \xrightarrow{rep\_r} B_2 + gA$ $A_2gB \xrightarrow{rep\_r} A_2 + gB$	dissociation
$A \xrightarrow{deg} \emptyset$ $B \xrightarrow{deg} \emptyset$	degradation

3.4.2 Parameter scan for model stability

The bifurcation analysis carried out in Section 3.3.2 can give us the conditions for bistability of a model with respect to one parameter being varied while all the others remain constant. Here I want to determine the range of parameter values for which the above model is bistable without assuming a specific point value for any parameter. Therefore I developed a parameter scanning algorithm, that varies all the parameters in the model over a given range. The algorithm is outlined in Algorithm 3 below. This method involves the scan of parameter values as well as initial conditions for A_2 and B_2. Parameter values are sampled randomly from a uniform distribution, U(0, 10). Here, each set of samples will be referred to as a particle. For each particle, latin hypercube sampling is used to sample initial conditions (McKay, Beckman, & Conover 2000). The uniform priors of the two species in consideration represent a rectangular space, which is subdivided into equal parts. Then a random sample is drawn from each sub-part, as illustrated in Figure 3.4. This is used to ensure that the whole space is sampled uniformly. Latin hypercube sampling is done in two dimensions, in order to sample initial conditions for the two dimers, A_2 and B_2. All other species have fixed initial conditions, with $gA = 1$ and $gB = 1$ while all other species are equal to zero.

Algorithm 3 Parameter scan algorithm

```
Select 100 sets of parameter values from a random uniform distribution between 0 and 10.
for each set of parameter values do:
    step ← 1
    for i in range(0, 10) do
        for j in range(0, 10) do
            A2 sample = random sample from U(i, i + step)
            B2 sample = random sample from U(j, j + step)
        end for
    end for
    for  each set of initial conditions do
        Integrate ODEs of the model to t = 100 mins
        Let s = {s1, s2, ..., s8}, the values of al species at last time point
        Find roots of system using s as starting estimates
    end for
    Plot phase plot of roots for A2, B2
end for
```

The roots of each particle for each initial condition were found using the Python

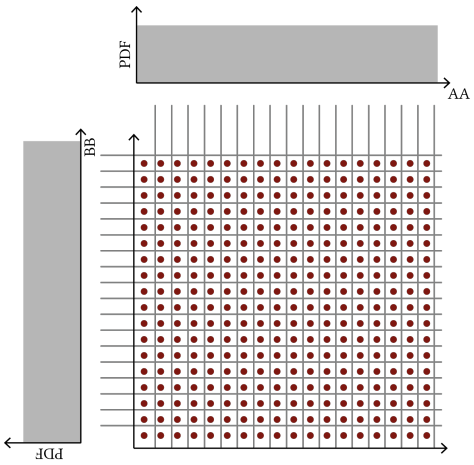

Figure 3.4 Latin hypercube sampling ensures that the whole space is sampled evenly. For the two species concerned, A_2 and B_2, we assume uniform distributions, shown in grey. The joint space of the two distributions is divided into smaller equal parts and a random sample is drawn from within each subspace. Diagram adapted from Youssef et al. (2013).

package SciPy (Jones, Oliphant, & Peterson 2001). A phase plot was made for each particle, which consists of the last time point value of one dimer plotted against the other. The parameter scan uncovered the presence of bistable and monostable systems given a different set of parameter values. An example of the phase plots for each of the two types of stabilities found during the scan is shown in Figure 3.5.

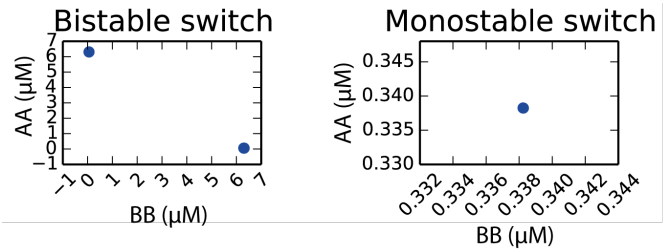

Figure 3.5 An example of each type of switch found during the parameter scan. Each graph represents the steady state values of one dimer plotted against the other, from one parameter set and 100 initial conditions.

A total of 1000 parameter sets were sampled. Out of these, 56% were monostable, 35.2% did not reach steady state and 8.8% where bistable. The distributions of the parameters that produced each stability are shown in Figure 3.6. From Figure 3.6 it can be seen that the parameter for gene expression (ge) tends to be relatively

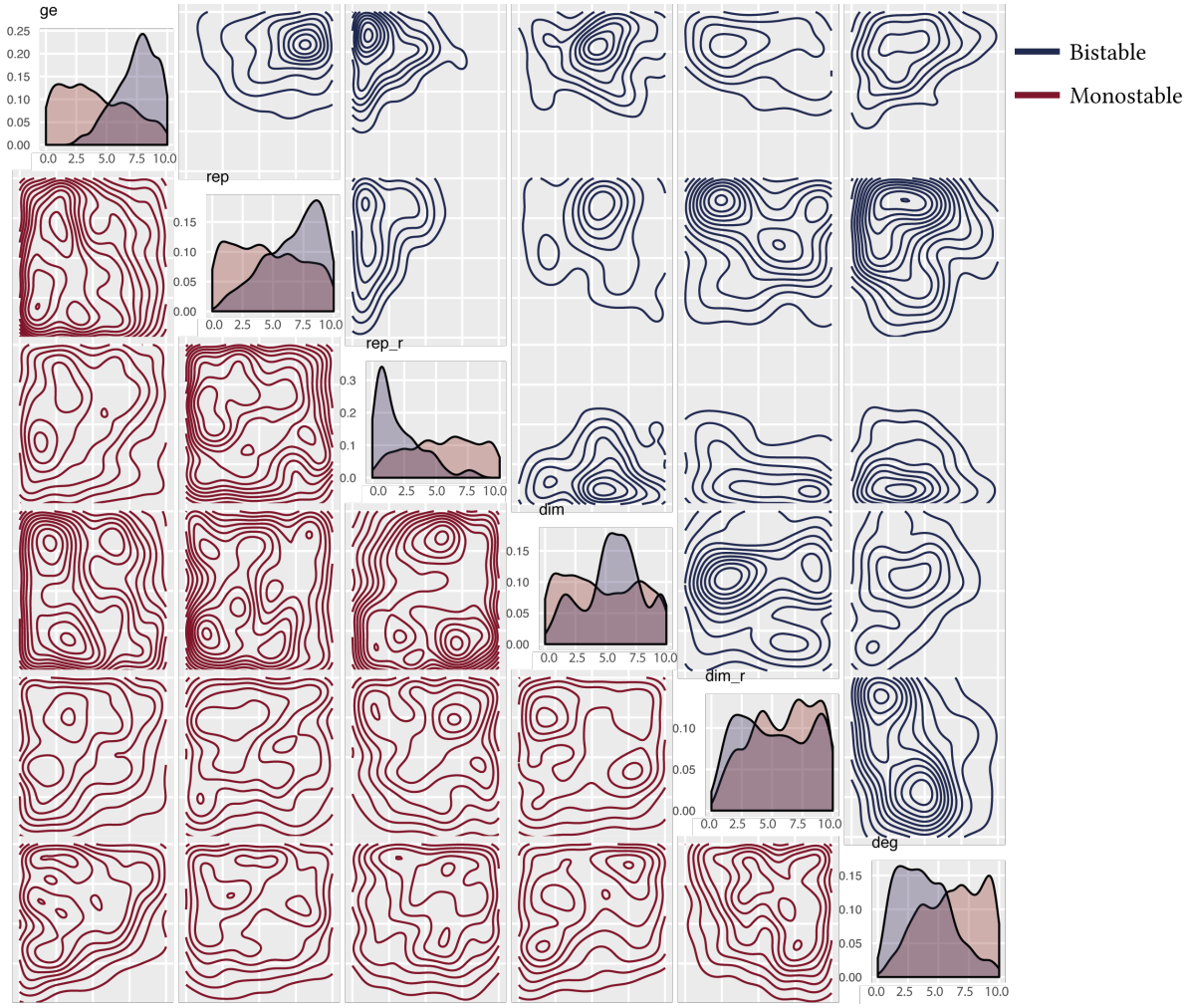

Figure 3.6 The distribution of parameter values that resulted in monostable and bistable switches in the parameter scan. The one-dimensional marginal distribution of each parameter is plotted on the diagonal and the two-dimensional marginal distributions are on the off-diagonal.

high when bistability arises. The parameter for repression (rep) tends to be high whereas the parameter for its reverse reaction (rep_r) tends to be low. From this analysis I showed that the toggle switch is capable of bistable behaviour for a range of parameter values.

3.4.3 Toggle switch parameter inference

In this section, I extend the analysis carried out in Section 3.4.2 to study the problem of how to rationally design a synthetic biological system to perform a behaviour of

Table 3.3 Toggle switch inducer equations

Equation	Description
$S + A_2 \xrightarrow{rep\_dim} SA_2$	Inducer repression
$R + B_2 \xrightarrow{rep\_dim} RB_2$	
$SA_2 \xrightarrow{rep\_dim\_r} S + A_2$	Inducer dissociation
$RB_2 \xrightarrow{rep\_dim\_r} R + B_2$	
$R \xrightarrow{deg} \emptyset$	Inducer degradation
$S \xrightarrow{deg} \emptyset$	

choice. In order to address this question, I use a Bayesian approach, known as Approximate Bayesian Computation and described in Section 2.3.1, implemented in a software package, ABC-SysBio (Liepe et al. 2010).

This approach is capable of approximating the posterior distribution that gives rise to the behaviour of choice (Toni et al. 2009). By simulating the model in question, this approach can identify an approximate posterior distribution via a series of intermediate distributions. This method can be used for the rational design of synthetic biological systems by defining some design objectives to which the model is fitted to (Barnes et al. 2011). By specifying the inputs to the system and the outputs required, the posterior of the model that can produce this behaviour can be identified.

Here I use ABC-SysBio to fit a model to the design objectives of a switch-like behaviour. I extend the model used in Section 3.4.2 by adding two inducers to the system, S and R. S removes the A homodimer (A_2) from the system by binding to it thus removing the repression on gene B. R removes B_2 from the system with the same mechanism. These inducers represent the stimuli that will turn the switch ON and OFF in a biological setting. The equations shown in Table 3.3 are added to the existing set of equations for the mass action switch:

The range of parameter values shown to produce a bistable switch in Section 3.4.2 was used as priors and are shown in Table 3.4. Initial conditions of the two genes, gA, gB were set to 1, and species A and A_2 were set to 50 and 10 respectively, in order to set the starting conditions of the switch to B_2 OFF. All other species initial conditions were set to 0. The system was simulated using ODEs.

Table 3.4 The prior distributions used for the standard toggle switch parameter inference. The values indicate the lower and upper limits (inclusive) of a uniform distribution.

ge (min^{-1})	**rep** (μM^{-1} min^{-1})	**rep_r** (min^{-1})	**dim** (μM^{-1} min^{-1})	**dim_r** (min^{-1})	**deg** (min^{-1})	**rep_dim** (μM^{-1} min^{-1})	**rep_dim_r** (min^{-1})	**deg_sr** (min^{-1})	**deg_dim** (min^{-1})
6-9	4-10	1-4	4-10	2-7	2-5	0.05-0.1	0.01-0.05	0.01-0.05	0-1

3.4.4 Design specifications

The following were defined as the design specifications for a bistable genetic toggle switch. The two inducers, S and R, are used as inputs to flip the switch ON and OFF respectively. The required output is the switch flipping from the OFF state to the ON state and then to the OFF state again (Figure 3.7).

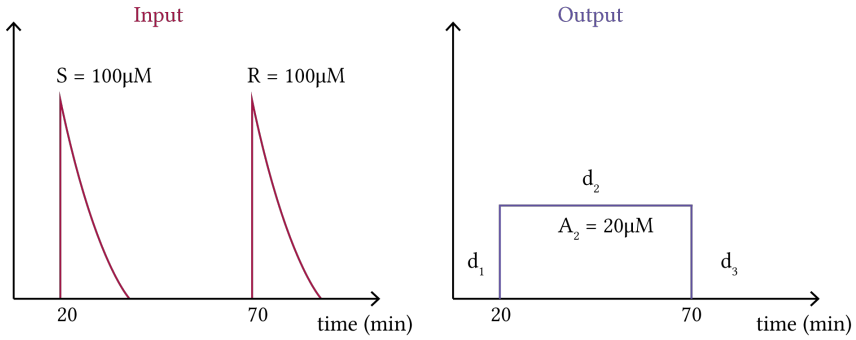

Figure 3.7 Design specification for ABC SMC parameter inference. The input to the system consists of an event turning on the stimulus (S) at t = 20mins and another turning on the repressor (R) at t = 70mins. The output specification is the response to the switch to these stimuli.

3.4.4.1 Distance function

In order to fit the switch model to the behaviour specified above, a distance function must be defined. The distance function defines the quantity that is minimised at each successive iteration of ABC SMC. Three distances are measured, one for each state of the switch, OFF-ON-OFF. Each distance is the sum of the distances of the simulated protein levels to the desired protein levels at each time point. All three distances must reach the minimum threshold for the process to be complete.

$$d_1 = \sum_{i=0}^{20} (s_i - t_1)^2 \tag{3.17}$$

$$d_2 = \sum_{i=21}^{70} (s_i - t_2)^2 \tag{3.18}$$

$$d_3 = \sum_{i=71}^{100} (s_i - t_3)^2, \tag{3.19}$$

where i represents the time points, s_i the simulation result at each time point and t_1, t_2, t_3 represent each target behaviour. t_1, t_2, t_3 were set to 0, 20, 0 respectively.

3.4.5 Results

The results of the parameter inference of the toggle switch are shown in Figure 3.8. The model was shown to successfully behave like a switch within the parameter range used here. The resulting time course of the last population matches the design specifications. It can be seen from the posterior distribution that gene expression rate (ge) must be high relative to the prior. Repression (rep) and degradation must both be low and the rate of dimerisation (dim) must be high relative to the prior. The posterior constitutes the specifications that can be used when building a synthetic switch in the lab, as the appropriate components can be tweaked for a successful circuit. More generally, the posterior distribution shows that the parameter space that can give rise to this behaviour is limited. A very small portion of the prior is capable of producing the desired design specifications. In the next section, I will examine whether the addition of feedback loops can increase the robustness of this behaviour.

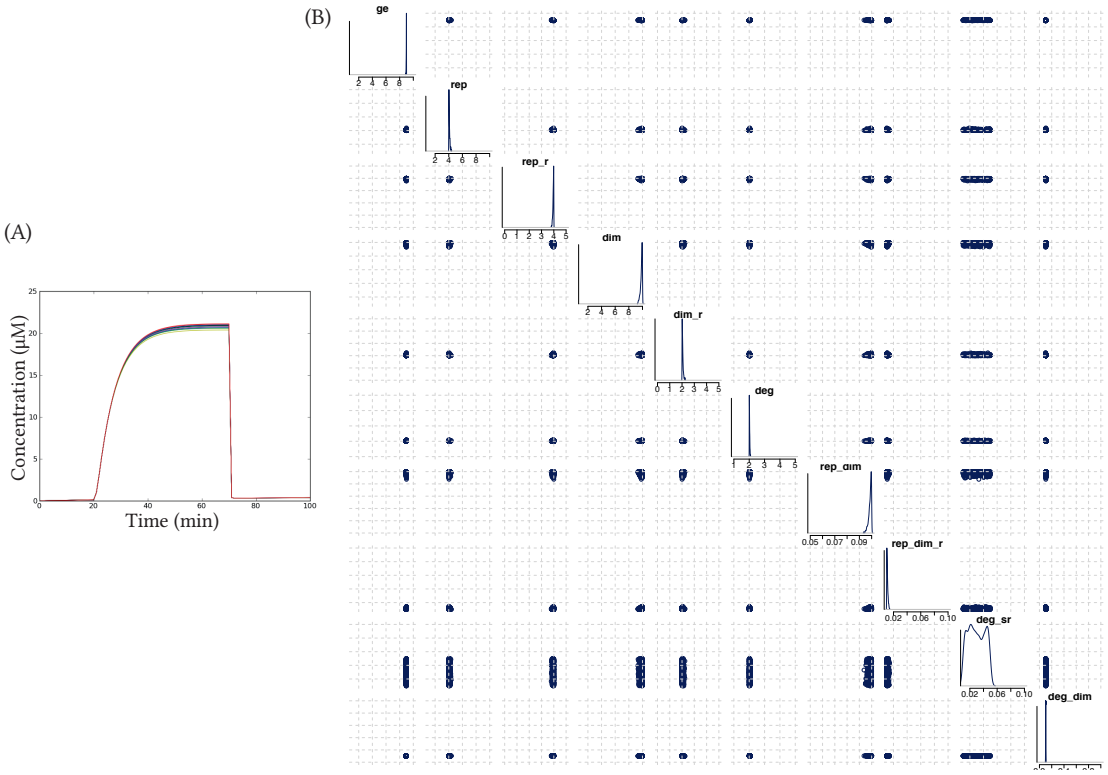

Figure 3.8 (A) The time series of the final population (for final ε = 2) of the standard toggle switch ABC-SMC parameter inference. The stimulus, that represses A_2, is added at t = 20mins and the repressor, that represses B_2 is added at t = 70mins. (B) The posterior distribution of the toggle switch. The one-dimensional marginal distribution of each parameter is plotted on the diagonal and the the two-dimensional marginal distributions are on the off-diagonal.

3.5 Designing a more robust genetic toggle switch

In this section, I examine whether the addition of feedback loops to the toggle switch can increase its robustness to parameter fluctuations. Here I define a robust system as a device that can withstand fluctuations in parameter values and still produce the desired behaviour (parametric robustness). Feedback loops are well known key regulatory motifs (Brandman et al. 2005). As was shown in Section 3.4, the posterior distribution of the simple toggle switch was narrow, and thus the behaviour of choice will not be robust. Here I examine whether adding feedback loops to the genetic toggle switch can increase parametric robustness for the desired design specifications.

3.5.1 Models of the genetic toggle switch

Both positive and negative feedback loops were considered. Therefore 7 models were examined for their capability to behave like a switch. The simple toggle switch, switches with positive autoregulation in either or both nodes and switches with negative autoregulation in either or both nodes. The models considered are illustrated in Figure 3.9.

In order to study each model mathematically I built extensions to the (Gardner, Cantor, & Collins 2000) toggle switch in order to incorporate positive and negative feedback to the system. These were constructed using the Shea-Ackers formalism, as described in Section 2.2.2.2, as shown in Figure 3.10. The models for the double autoregulation models are shown below. For the single autoregulation models, the unnecessary autoregulation term is set to 0.

Double negative autoregulation:

$$\frac{du}{dt} = \frac{a_1 l_1}{1 + l_1 + k_1 v^{\beta} + k a_1 u^2} - u \tag{3.20}$$

$$\frac{dv}{dt} = \frac{a_2 l_2}{1 + l_2 + k_2 u^{\gamma} + k a_2 v^2} - v. \tag{3.21}$$

Double positive autoregulation:

$$\frac{du}{dt} = \frac{a_1 (l_1 + k a_1 u^2)}{1 + l_1 + k_1 v^{\beta} + k a_1 u^2} - u \tag{3.22}$$

$$\frac{dv}{dt} = \frac{a_2 (l_2 + k a_2 v^2)}{1 + l_2 + k_2 u^{\gamma} + k a_2 v^2} - v, \tag{3.23}$$

where k represents the effective binding of the transcription factor to the other promoter, ka_1 represents the binding rate of the transcription factor dimer to it

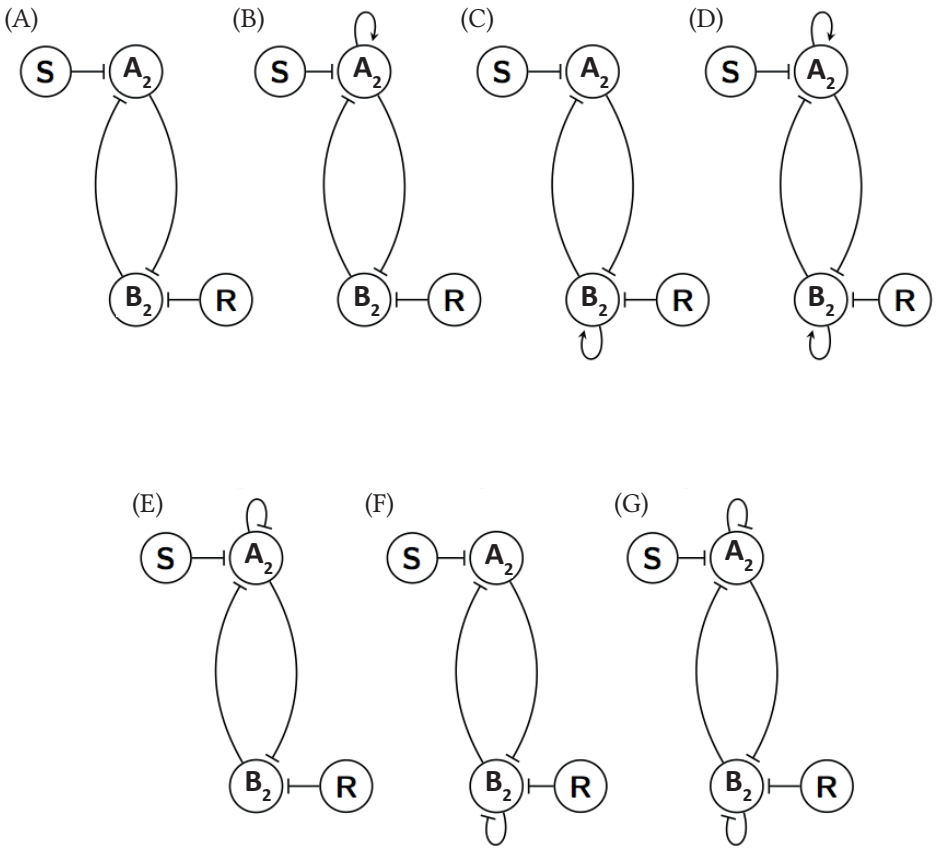

Figure 3.9 Toggle switch designs considered for model selection. A and B are the two protein species involved in the switch, S is the stimulus inducer turning the switch ON and R the repressor turning the switch OFF. 7 models were used: (A) The simple toggle switch (B-D) the switches with positive autoregulation on either or both nodes and (E-G) the switches with negative autoregulation on either or both nodes.

own promoter and β, γ represent the polymerisation of the bound transcription factors.

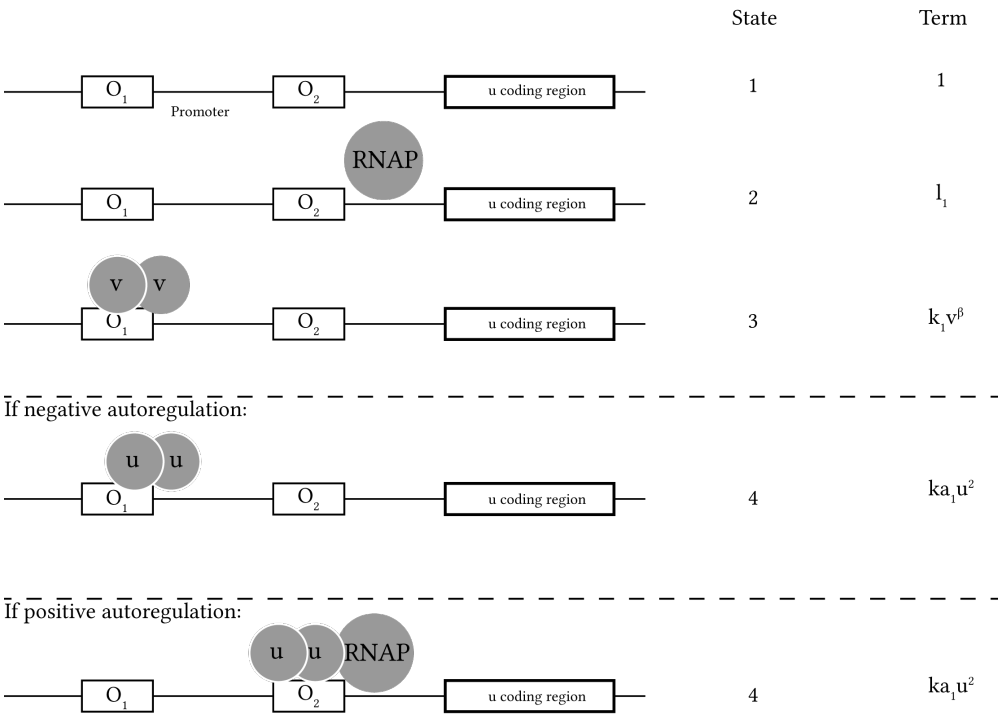

Figure 3.10 Toggle switch models with autoregulation using the Shea-Ackers formalism

3.5.1.1 Autoregulatory switches phase space and bifurcation analysis

In order to use these models for model selection, it must first be determined whether they are capable of behaving like a switch. ABC SMC model selection is used to select models that can produce the same behaviour, over a greater range of parameter values. If a model is not capable of producing the desired behaviour for the prior range, then it will not be used for model selection.

I used the PyDSTool (Clewley 2012) in order to determine whether each of the 7 switches is capable of bistable behaviour. The same analysis as Section 3.3.2 was used to identify the steady states of the systems and their stabilities. As shown in Figure 3.11, both single and double positive autoregulation are consistent with bistable behaviour. Two stable states were found for both cases when $k = 2$ and $\delta = 1$.

On the other hand, negative autoregulation was not consistent with bistable behaviour for the parameter values used here. The vector plot of the switch with single negative autoregulation shows that there is one stable steady state when the levels of the unregulated protein are high and the levels of the negatively regulated protein low. The vector plot for the switch with double negative autoregulation shows one stable steady state when the levels of both proteins are low. A bifurcation plot of the switch with single positive autoregulation highlights this result; the switch is bistable for only very small amounts of ka_1, the parameter for negative autoregulation. This is shown in Figure 3.12.

Negative autoregulation occurs when the protein binds to its own promoter and represses production. Therefore the protein levels rise to a lower level than what would be expected for an unregulated system (Alon 2007). In the case of the double negative autoregulation, the concentration levels of either protein would therefore not be sufficient to dominate over the other and create bistability. In the case of the single negative autoregulation, only the protein without autoregulation is able to reach sufficient levels to dominate the system, and thus the only steady state of the system is when the unregulated protein is high.

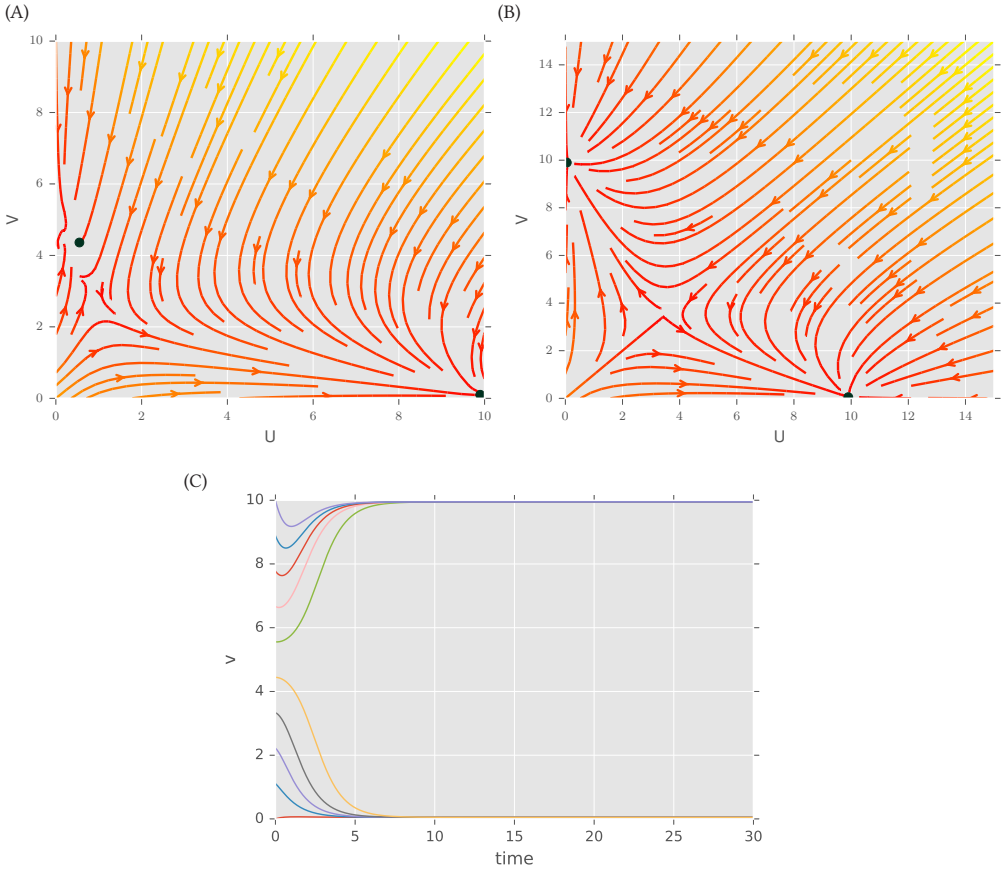

Figure 3.11 Both single and double positive autoregulation are consistent with bistable behaviour. (A) Single positive autoregulation. Two stable steady states are found in the single positive autoregilation model at (u, v) = (9.89, 0.55) and (u, v) = (0.1, 4.36). (B) Double positive autoregulation. The model with double positive autoregulation has two stable steady states at (u, v) = (0.051, 9.89) and (u, v) = (9.89, 0.051).

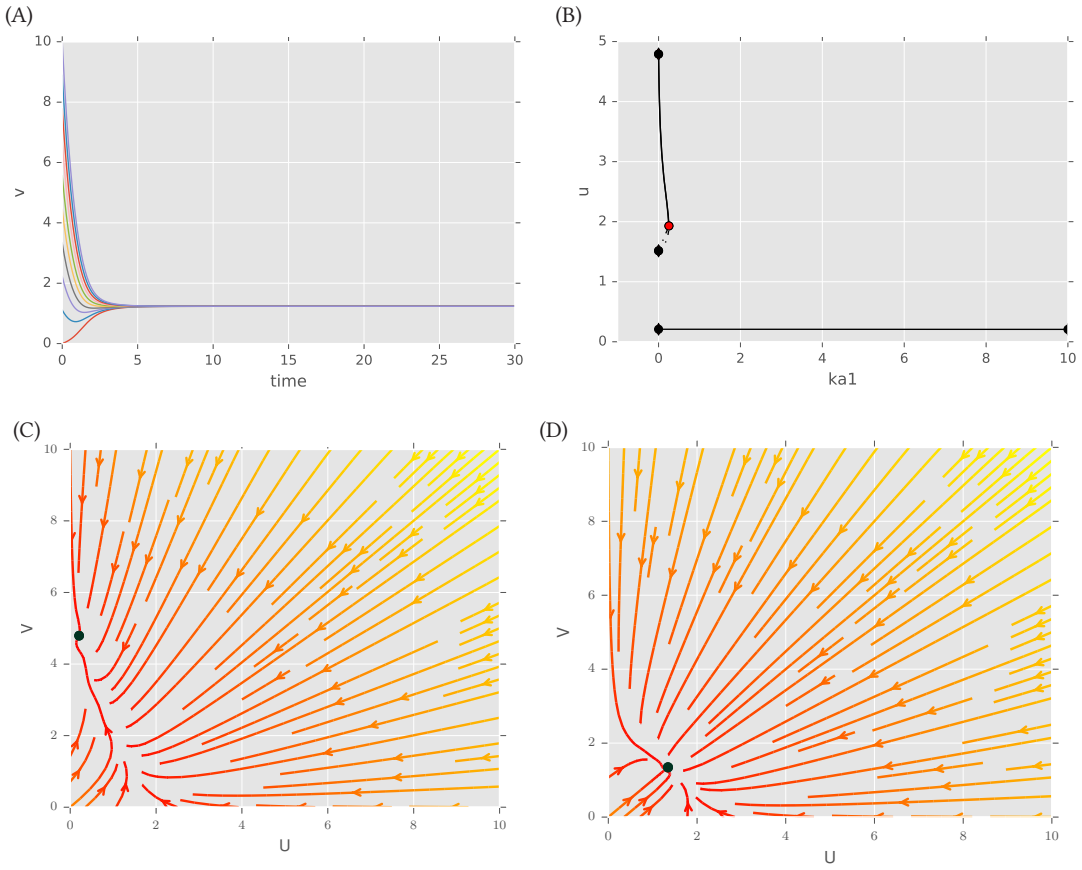

Figure 3.12 (A) Time course of the single negative autoregulation switch, when k_u = 2 with varying initial conditions. The system is monostable. (B) Bifurcation diagram with respect to parameter ka_1 of the switch with single negative autoregulation. The system is bistable only for very small amounts of ka_1. (C) One stable steady state is found for the single negative autoregulation switch when k_u = 2 at u = 0.208 and v = 4.79. (C) The vector plot of the switch with double negative autoregulation has one steady state at u = 1.346 and v = 1.346.

The models shown here to be capable of bistable behaviour will be used in the following section for model selection, to determine whether the addition of positive feedback loops increases the robustness of the system to parameter fluctuations.

3.5.2 ABC SMC for model selection

In Section 3.4.3 I used ABC SMC to infer the parameters of a model that can produce the desired behaviour. In this Section I extend the analysis by asking which model, out of the 5 candidate models shown above, can best produce the desired behaviour. Bayesian model selection can be used to rank models according to their posterior distributions.

ABC SMC model selection works by treating each model as another 'parameter'. ABC SMC then proceeds to approximate the posterior distribution over the joint space of all models (Toni et al. 2009). The marginal posterior distribution (also known as model evidence) for each model (m) can then be used to rank all the models (M).

$$p(m|D) = \frac{\text{accepted particles from } m}{\text{all accepted particles}} \tag{3.24}$$

ABC SMC model selection accounts for model complexity and parametric robustness in order to avoid the problem of overfitting. It automatically applies the rule of parsimony, or Occam's razor, which dictates the simplest model to account for the data is the best model (Toni 2010). This is done by penalising the addition of a parameter, corresponding to an increase in the volume of the prior if it does not result in a larger increase in the volume of the approximated posterior distribution (Woods et al. 2016). Bayesian model selection can be used to rank models according to how well they describe the data, or how likely they are to give rise to the data we wish to see in a system design setting.

Here I apply ABC SMC model selection using the package ABC-SysBio (Liepe et al. 2010). I study whether the addition of feedback loops to the standard toggle switch can increase its parametric robustness. The standard toggle switch was compared to switches with positive autoregulation in one or both nodes, which were shown to be capable of bistable behaviour in Section 3.5.1.1. The mass action models were used for model selection, in order to represent the system in a more realistic way. The equations shown in Table 3.5 are added to the equations of the simple toggle switch used in Section 3.4.3. Toggle switches with autoregulation on both

nodes have both sets of equations added. The same design specifications as used in Section 3.4.3 were used.

Table 3.5 Autoregulated switches additional equations

Equations	Description
Positive autoregulation A	
$A_2 + gA \xrightarrow{aut\_1} A_2gA$	dimer self-association
$A_2gA \xrightarrow{aut\_2} A + A_2gA$	self-induced expression
$A_2gA \xrightarrow{aut\_3} A_2 + gA$	dimer self-dissociation
Positive autoregulation B	
$B_2 + gB \xrightarrow{aut\_1} B_2gB$	dimer self-association
$B_2gB \xrightarrow{aut\_2} B + B_2gB$	self-induced expression
$B_2gB \xrightarrow{aut\_3} B_2 + gB$	dimer self-dissociation

Given the models shown above and the parameter priors shown in Table 3.6, ABC SMC model selection was carried out. The priors for model selection were set to wider that in the ones used during parameter inference of the simple switch in order to make them more flexible for the additional models. For all models initial conditions of the two genes, gA, gB were set to 1, and species A and A_2 were set to 50 and 10 respectively, in order to set the starting conditions of the switch to B_2 OFF. All other species initial conditions were set to 0.

Table 3.6 The prior distributions used for model selection. The values indicate the lower and upper limits of a uniform distribution.

ge (min^{-1})	**rep** ($\mu M^{-1}\ min^{-1}$)	**rep_r** (min^{-1})	**dim** ($\mu M^{-1}\ min^{-1}$)	**dim_r** (min^{-1})	**deg** (min^{-1})	**rep_dim** ($\mu M^{-1}\ min^{-1}$)	**rep_dim_r** (min^{-1})	**deg_sr** (min^{-1})	**deg_dim** (min^{-1})	**aut_1** ($\mu M^{-1}\ min^{-1}$)	**aut_2** (min^{-1})	**aut_3** (min^{-1})
1-10	1-10	0-5	1-10	0-5	1-5	0.05-0.1	0.01-0.1	0.01-0.1	0-1	0-10	0-2	0-10

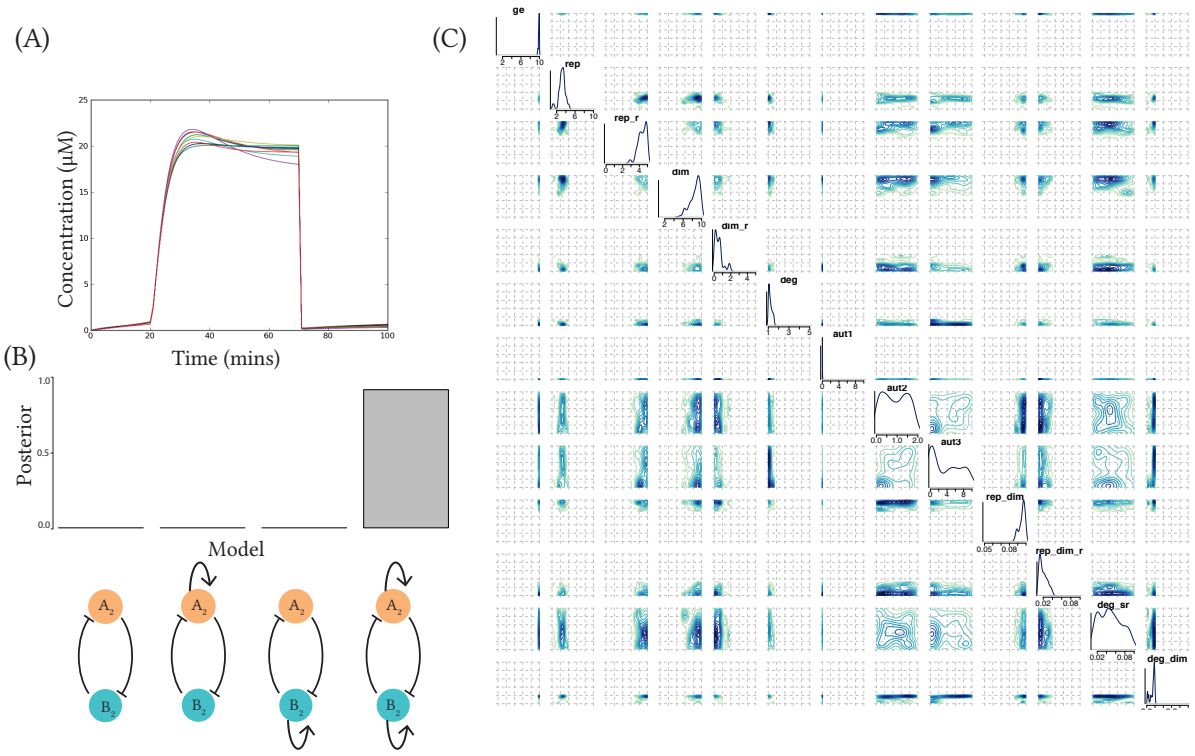

Figure 3.13 (A) The simulated time course of the last population of the switch with double positive autoregulation. (B) The toggle switch with positive autoregulation on both nodes was found to be the most robust to parameter fluctuations. Three repeats of the model selection were carried out, and the median values are shown here. Upper and lower quartile error bars are included but are too small to be visible. (C) The posterior distribution of the toggle switch with positive autoregulation on A and B.

The results are shown in Figure 3.13. The toggle switch with positive autoregulation on both nodes was found to be the most robust model. This indicates that although all the models considered are capable of behaving like a switch, the model with double positive autoregulation could do that over a greater parameter range.

3.6 Discussion

Here I developed a more realistic model for the genetic toggle switch, using mass action which does not use the QSSA. In Chapter 4 I explore this model further and show that the QSSA cannot always be justified. I showed that this model is capable of bistable behaviour, within a given parameter range. I further studied this by using ABC SMC parameter inference to fit the toggle switch to a switching behaviour of choice. The parameter inference revealed the range of parameter values that can produce the behaviour of choice. These parameter values can be used to design a synthetic toggle switch that will behave in a specified manner.

Further, I showed that negative autoregulation is not consistent with bistable behaviour for the parameter values examined here. Adding small levels of single negative autoregulation to the system caused it to revert to monostability. The only steady state was when the unregulated protein was high and the negatively regulated protein was low. This makes sense, as the negatively regulated protein cannot reach high enough levels to repress the other protein and dominate the system, whereas the unregulated protein is free to reach a higher steady state. In the case of double negative autoregulation, neither protein is free to reach sufficient levels to dominate the other, and the only steady state was when the levels of both proteins are low. This indicates that the system reaches a deadlock situation where both proteins are repressed and cannot reach a higher steady state. The models used here are deterministic and simplified down to two equations. Stochastic dynamics where noise is added to the system or a more complex model could be capable of overcoming this deadlock situation. More specifically, if transcriptional or translational bursting is included, the protein that receives the first boost can dominate the system in time and escape the deadlock situation (Strasser, Theis, & Marr 2012).

Finally, I showed that the addition of positive feedback loops makes the genetic toggle switch more robust to parameter fluctuations. This means that the model was capable of producing the desired behaviour over a greater parameter range. This indicates that small fluctuations in parameters in the cellular environment will not affect the system's ability to be bistable, and thus makes it more suitable for use

in synthetic biological applications where a very constrained parameter set can be too restrictive. This makes it a better candidate for building new synthetic devices based on the toggle switch design.

The volume of the posterior distribution of even the most robust switch out of the ones examined here was still not large. This means that the behaviour of choice is still constrained, even after the addition of the positive feedback loops. A caveat of the analysis used here that has to be considered is that the parameter space is not searched for simply combinations that can produce a bistable switch. The behaviour that is required is very specific, and it is probable that the plethora of constraints put on the system result in the discarding of parameter combinations that create a bistable switch. Firstly, a specific steady state level is required, for both the ON and OFF states. When the switch is OFF, the protein levels must be as close to zero as possible, and when the switch is ON, the protein levels are required to approach 20μM. This requirement will discard any switches that have a higher or lower ON state, or a slightly higher OFF state. Additionally, the design specifications used here dictate that the time to reach steady state has to be quick. There is not much transition time allowed between the ON and OFF states, and the protein is required to reach steady state within a few time points. Here, model time was taken to represent minutes, thus a time point in the simulation represents a minute. This dictates that the switch has to occur very fast. This results in the exclusion of systems that reach steady state more slowly, but still act as a bistable switch. Therefore, the switches examined here, follow a very constrained behaviour. In the next Chapter I develop a method that is more flexible in the type of switches it can analyse.

It is important to discuss the assumptions made in the models presented in this Chapter. As was discussed in Chapter 2, the assumptions made about the underlying model can affect the behaviour observed in the model. Firstly, the numerous steps involved in transcription and translation have been condensed into one gene expression step. This simplifies the model by reducing the number of parameters and reactions included, and it has been shown that it does not affect the stability of the switch (Warren & ten Wolde 2005). The other assumption being made here is that repression is cooperative. This means that a transcription factor dimer has to bind to the promoter region to repress it successfully. This has been shown to be critical for a bistable switch (Gardner, Cantor, & Collins 2000; Warren & ten Wolde 2005; Warren & ten Wolde 2004; Cherry & Adler 2000), as discussed in Chapter 2. Lipshtat et al. (2006) found that a switch can be exhibit bistability without cooperativity, only if stochasticity is taken into account. Since the switch models

were simulated using deterministic dynamics here, the assumption of cooperativity was necessary to make it bistable. Finally, the models used here correspond to a general switch, i.e. a switch where both repressors A_2 and B_2 are free to bind to the promoters simultaneously. This is in contrast to the exclusive switch, where the two promoters are assumed to overlap and the two proteins cannot bind at the same time (Loinger et al. 2007). The exclusive switch can be found in natural systems (Ptashne 1992) and has been shown that the exclusive switch is more robust than the general switch (Loinger et al. 2007; Barnes et al. 2011). Nevertheless, here I model a synthetic genetic switch, like the one build by Gardner, Cantor, & Collins (2000), which consists of two separate promoters, making the general switch model more appropriate.

3.7 Summary

In this chapter, I studied the Gardner, Cantor, & Collins (2000) toggle switch and showed that it is bistable. I further studied the genetic toggle switch model using mass action. I identified the parameter ranges that produce a bistable behaviour and could be used as prior distributions for parameter inference. Further, I studied the effect of adding feedback loops has on the robustness of the genetic toggle switch. I found that the switch with double positive autoregulation is the most robust to parameter fluctuations. In the next chapter I address some of the shortcomings of the method used here by developing a new algorithm, StabilityFinder.

4 Dynamics of multi-stable switches

4.1 Introduction

In this chapter, I aim to uncover the underlying principles that govern the stability of a given switch. To do this, I developed an algorithm, called StabilityFinder, that can identify the parameter value ranges that can produce the desired stability in a given model. I use this algorithm to examine a variety of switch architectures using different modelling abstractions.

Structurally, this chapter is organised as follows: In the first section I examine the current understanding of the stability landscape of the genetic toggle switch. Then, I discuss the development of StabilityFinder, justify the choices made and the drawbacks of this method. In the sections following I apply StabilityFinder to a variety of models and finally I discuss the implications these findings have to the overall understanding of the toggle switch stability.

4.2 Contributions to this Chapter

The phase plots of the Lu switch and the characterisation of their steady states shown in Figure 4.6 was carried out by Mae Woods, PhD.

4.3 Motivation

Synthetic biology puts an emphasis on creating modular, standardized parts that can be used to create larger systems (Agapakis & Silver 2009). When faced with the creation of a new model design, the researcher can select the appropriate parts from the BioBrick registry (Müller & Arndt 2011) and combine them to create the system of choice. Synthetic circuit design presents a challenge as the collection of assembled parts have to work together to create the target behaviour (Nielsen,

Segall-Shapiro, & Voigt 2013). Parts can be fine tuned by developing component libraries (Lu, Khalil, & Collins 2009), but this will be of little use if the required parameter ranges for parts to make a functional system are unknown, and will only perpetuate the cycles of trial-and-error. A computational method to find the range of parameter values that will produce the behaviour of choice is crucial to the design process by enabling the informed selection of appropriate parts from the libraries. For example, if it is known that gene expression must be low for a given stability, one can select a weak promoter or a low copy plasmid for the desired construct.

Both analytical and computational approaches have been deployed for the study of the toggle switch. Analytical approaches are limited to simpler models and thus require a number of assumptions to be made. The system under consideration has to be reduced to very few equations and parameters in order to make the system solvable. This requires assumptions to be made about the system that cannot always be justified, such as the quasi-steady state approximation (QSSA). The QSSA assumes that the binding/unbinding processes are much faster than any other process (Loinger et al. 2007), thus the bound intermediate is assumed to always be in steady state. The QSSA assumption is met *in vitro* but often does not hold *in vivo* and its misuse can lead to large errors and incorrectly estimated parameters (Pedersen, Bersani, & Bersani 2007). Moreover, it is generally not possible to solve even simple stochastic models analytically, and these methods are restricted to deterministic models. The computational and graph-theoretic approaches developed for the study of multistationarity generally focus on deciding on whether a given system is incapable of producing multiple steady states (Conradi et al. 2007; Banaji & Craciun 2010; Feliu & Wiuf 2013). For example, Feliu & Wiuf (2013) developed an approach using chemical reaction theory and generalised mass action modelling (Feliu & Wiuf 2013). No approach exists that can handle both deterministic and stochastic systems in an integrated manner.

For this purpose, I developed a computational framework based on sequential Monte Carlo that takes a model and determines whether it is capable of producing a given number of steady states and the parameter space that gives rise to the behaviour. Uniquely, this can be done for both deterministic and stochastic models, and also complex models with many parameters, thus removing the need for simplifying assumptions. This framework can be used for comparing the conclusions drawn by various modelling approaches and thus provides a way to investigate appropriate abstractions. I have made this framework into a python package, called StabilityFinder.

I use this methodology to investigate genetic toggle switches and uncover the design principles behind making a bistable switch, as well as those necessary to make a tristable and a quadristable switch (4 steady states). The number of stable steady states will be referred to as the desired stability of the model in this thesis. I also demonstrate the ability of StabilityFinder to examine more complex systems and examine the design principles of a three gene switch. The examples I used demonstrate that StabilityFinder will be a valuable tool in the future design and construction of novel gene networks.

4.4 StabilityFinder algorithm

To investigate the multistable behaviour of systems, I had to make a number of extensions to existing approaches. Firstly, a wide range of initial condition samples are required in order to determine the stability of a system. For a given set of parameter values, sample points are taken across initial conditions using latin hypercube sampling (McKay, Beckman, & Conover 2000), and the ensemble system simulated in time until steady state. As a distance function I use the desired stability of the simulated model. An overview of the algorithm is given in Section 4.4.1.

4.4.1 Algorithm overview

The StabilityFinder algorithm is summarised below. StabilityFinder is available as a Python package, and can be downloaded from https://github.com/ucl-cssb/StabilityFinder.git.

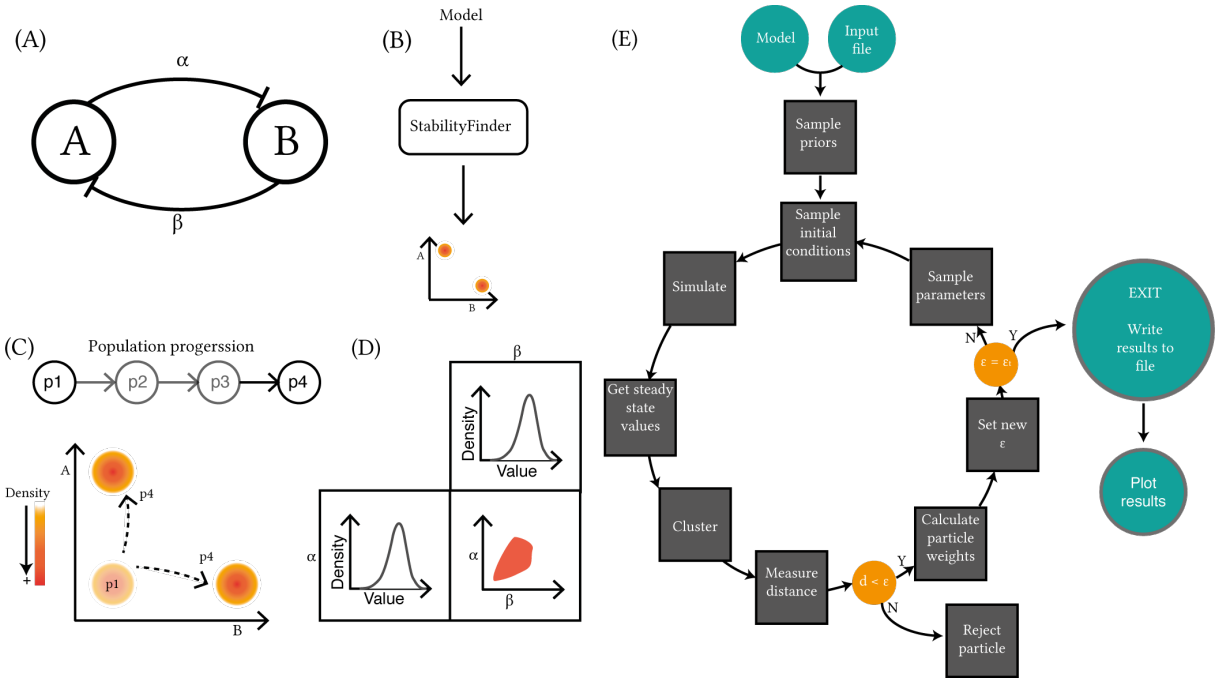

Figure 4.1 Using sequential Monte Carlo to examine system stability. The algorithm takes as input a model (A) and evolves it (B) to the stability of choice via (C) intermediate populations. In this example model shown in (A), There are two species and two parameters. For the model to be bistable, the phase plot of the two species of interest must have two distinct densities, as shown in (D). The parameter space of the model is searched through our algorithm until the resulting simulations give rise to bistability. The parameter values for the model that demonstrated the desired behaviour are given as an output (D). The output consists of the accepted values for each parameter, as well as each density plotted against the other. This allows us to uncover correlations between parameter values. This algorithm is available as a python package, called StabilityFinder. The overview of the algorithm is shown in (E).

The user provides an SBML model file (Hucka et al. 2003) and an input file that contains all the necessary information to run the algorithm, including the desired stability and the final tolerance, ε, for the distance from the desired behaviour necessary for the algorithm to terminate. The flow of execution is illustrated in Figure 4.1E. Since the algorithm is computationally intensive, all deterministic and stochastic simulations are performed using algorithms implemented on Graphical Processing Units (GPU)s, which are used for mutli-threaded computation (Kirk & Hwu 2010).

4.4.2 Initial condition sampling

In StabilityFinder, latin hypercube sampling is used to sample initial conditions (McKay, Beckman, & Conover 2000). This is used to ensure that the whole space is sampled uniformly. Latin hypercube sampling is done in two dimensions in StabilityFinder. This is the same algorithm used for initial condition sampling in Chapter 3, and the reader is referred to Section 3.4.2 for a description of this algorithm. StabilityFinder could easily be implemented to be used for a larger number of species, but here it has only been used for stability analyses concerning two species. Stability landscapes involving more that two species are beyond the scope of this thesis.

4.4.3 Clustering methods

Whether the model was simulated using ODEs or the Gillespie algorithm (Gillespie 1977) dictated the method of clustering that I used. For the deterministic models I used an algorithm I developed, that will be referred to as the delta clustering algorithm in this thesis. This algorithm consists of defining the number of clusters by counting a new cluster every time a data point is more than a distance δ away from any existing clusters. The benefits of the delta clustering algorithm are that it is fast and can be used on deterministic solutions, where steady state values tend to be identical if all the particles have reached steady state.

Steady states of stochastic models are clustered using the K-means clustering (Lloyd 1982) and the number of clusters determined using the Gap statistic (Tibshirani, Walther, & Hastie 2001). This method is more suited to stochastic solutions, where the delta clustering method would fail as the steady state solutions tend to be more widely dispersed than in the deterministic case. The detailed algorithms used are shown in Appendix C.

The method used for clustering can be altered by the user if he/she wants to add their own preferred clustering algorithm that might be more appropriate for their specific purposes. For the models I used here, the above methods were successful in clustering the steady state solutions.

4.4.4 Distance function

The distance function is used to compare the desired behaviour to the behaviour observed in each particle (Toni et al. 2009). In StabilityFinder the distance function consists of three distances. The first one is the difference between the number of desired clusters and the number of clusters observed in the phase plot. For this distance metric the number of clusters in the phase plot must be calculated. The clustering methods used are outlined in Section 4.4.3.

The other two distance metrics used in StabilityFinder are the variance within each cluster and the overall (between cluster) variance. The within cluster variance ensures that the clusters are tight, and the between cluster variance is used to ensure the clusters are far apart from each other. In the context of this thesis, the ideal behaviour of a system is tight, widely separated clusters. This means that the genetic system has distinct steady states, and the difference in the protein levels between each steady state is observable.

$$d_1 = C = M_c \tag{4.1}$$

$$d_2 = V_{tot} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu) \tag{4.2}$$

$$d_3 = V_{cl} = \underset{m_c=1}{\overset{Mc}{\text{median}}}\{\frac{1}{n_{m_c}}\sum_{c=1}^{n_{m_c}}(x_c - \mu_c)\}, \tag{4.3}$$

where n denotes the total number simulations per particle, x refers to each simulation steady state of a given particle and μ the mean value of all the simulations steady states in each particle. M_c denotes the total number of clusters per particle and m_c refers to each cluster. x_c and μ_c represent the simulation steady state of a given particle in the current cluster and the mean of the current cluster respectively.

Once the distance from the desired behaviour has been calculated, the algorithm rejects any particles whose distance is farther than the current ε. The distances taken into account are the number of clusters (C), the total variance (V_{tot}) and the within cluster variance (V_{cl}) as outlined in Equations 4.2-4.3. In addition to these distances I have included another two checks for the particles. Firstly, StabilityFinder checks if the simulation of a particle has reached steady state. If the standard deviation of the last ten time points in the simulation (denoted as SS_{ss}) is larger than a user-specified value, then the particle is rejected. This is to ensure that only particles that have reached steady state are considered. Secondly, there is a check for the minimum level of the steady states (denoted as SS_l). This is to allow the user to select for steady states whose protein levels are above a certain threshold. This has to be added as an additional check as the steady state levels must be experimentally observable if they are to be used to design new systems. Two steady state levels of very low levels would be biologically indistinguishable and thus meaningless in an experimental setup. This check is optional to the user, and can be set to zero if not desired. In summary, a particle must satisfy all of the criteria given below in order to be accepted:

$$M_c \leq \varepsilon_{M_c}$$
$$V_{tot} \leq \varepsilon_{V_{tot}}$$
$$V_{cl} \leq \varepsilon_{V_{cl}}$$
$$SS_{ss} \leq \varepsilon_{SS_{ss}}$$
$$SS_l \leq \varepsilon_{SS_l}$$

4.4.5 Model checking

A problem that can arise by using this method with stochastic simulations is that the behaviour observed may not be the true behaviour but it might be a result of noise. We need to ensure that the resulting behaviour is reproducible. Therefore, I added model checking to the algorithm. Model checking consists of resampling from the posterior distribution and simulating each sample. If the resulting behaviour is the same as what we expected we can be confident that it is the true behaviour of the system and not a result of noise.

Algorithm 4 StabilityFinder algorithm

1: Initialise $t = 0$,
2: $i = 0$
3: **if** t $= 0$ **then**
4: Sample particle from prior, $\theta^{**} \sim \pi(\theta)$
5: **else**
6: Sample θ^{*} from the previous population $\{\theta^{i}_{t-1}\}$ with weights w_{t-1}.
7: Perturb the particle, $\theta^{**} \sim K_t(\theta|\theta^{*})$ where K_t is the perturbation kernel.
8: **end if**
9: Sample k initial conditions $\{x^{k}_{0}\}$ via latin hypercube sampling.
10: Simulate k datasets to steady state, $\{x^{*k}\}$, from the the model, $x^{*} \sim f(x|\theta, x_0)$
11: Apply clustering in phase space on $\{x^{*k}\}$
12: Calculate the distance $d = \rho(\{x^{*k}\}, y)$.
13: **if** $d \leq \epsilon_t$ **then**
14: $\theta^{i}_{t} = \theta^{**}$. $i = i + 1$
15: **if** $i \leq N$ **then** GoTo step 3
16: **else**
17: Calculate weight for each accepted θ^{i}_{t}
18: $$w^{(i)}_{t} = \begin{cases} 1, & \text{if } t = 1 \\ \frac{\pi(\theta^{(i)}_{t})}{\sum_{j=1}^{N} w^{(j)}_{t-1} K_t(\theta^{(j)}_{t-1}, \theta^{(i)}_{t})}, & \text{if } t \geq 1. \end{cases}$$
19: Normalise weights
20: $t = t + 1$.
21: **if** $t \leq N_t$ **then**
22: GoTo step 3
23: **end if**
24: **end if**
25: **end if**

4.5 Calculating robustness

Unlike Toni et al. (2009), StabilityFinder does not have model selection integrated into the method. This is because the purpose of StabilityFinder is not necessarily to compare models for robustness but to elucidate the stability a given model is capable of. Nevertheless, robustness analysis is an outcome that Bayesian methods are well suited for. Therefore, here I discuss another algorithm I developed in order to extract robustness information from the results of StabilityFinder and apply model selection.

As discussed in Section 2.3.2, two models M_a and M_b can be compared for their robustness using the Bayes factor, defined as follows:

$$B_{ab} = \frac{p(D|M_a)}{p(D|M_b)}, \tag{4.4}$$

which represents the fraction of the evidence supported by model a over the evidence supported by model b. This can be interpreted as the ratio of the fraction of the volume of the functional region to the volume of its prior of model a and model b:

$$B_{ab} = \frac{|F_a|}{|P_a|} / \frac{|F_b|}{|P_b|}, \tag{4.5}$$

where $|F|$ is the volume of the functional region of model and $|P|$ the volume of the prior. Equation 4.5 represents the ratio of the robustness measure of each model which in turn is defined as the ratio between the volume of functional region F and the volume of the prior P. The reader is referred to Section 2.3.2 for a further discussion.

In order to calculate the Bayes factor we must first be able to approximate the volume of the viable parameter space. The viable parameter space is the space that approximates the posterior distribution that can give rise to the desired behaviour. I tested two methods of approximating the volume of the viable space, which are outlined in Algorithm 5. The first method is based on the method used by (Hafner et al. 2009), where the volume of the cuboid containing all the viable space is calculated. I modified this part of their method by only including the area of the viable space where the majority of the last population lies. Therefore only the 1^{st} and 99^{th} percentile of the viable space are taken into account. This is necessary in order to exclude outliers in the distribution that would skew the volume calculation significantly. Each parameter represents a side in the cuboid and since the volume of a

cuboid is equal to the product of its sides, the volume of the viable space is equal to the product of the ranges of all the parameters. This cuboid method will be prone to overestimating robustness especially in cases of correlation between parameters. This caveat could be alleviated if a Principal component analysis (PCA) (Fukunaga 2013) is done on the data before the cuboid is calculated (Hafner et al. 2009). This would align the axes of the cuboid to the major axes of the distribution. This would still be a crude estimation of the volume, since if the posterior distribution is assumed to be normally distributed the volume would still be overestimated.

Thus I used a second method, where the volume of the viable space was represented by a hyper-ellipsoid, an ellipse in higher dimensions. This method should not be as prone to overestimation of robustness as the cuboid method as an ellipsoid can take correlation into account. For this method the distribution of the viable space is assumed to be normal. The method calculates the covariance matrix of the distribution, whose volume is given by Equation 4.6. Just as in the cuboid method, the 1st and 99th percentile of the data is ignored.

$$V = \frac{2\pi^{\frac{k}{2}}}{k\Gamma(\frac{k}{2})}\left[\chi_k^2(\alpha)\right]^{\frac{k}{2}}|\Sigma|^{\frac{1}{2}}, \tag{4.6}$$

where k is the number of dimensions, Γ is the Gamma function, α is the confidence interval required and $|\Sigma|$ is the determinant of the covariance matrix.

To validate these methods I compare them to ABC-SysBio model selection (Liepe et al. 2014). ABC-SysBio has been used extensively for model selection (Toni et al. 2009; Toni et al. 2011; Barnes et al. 2011) and is thus used as a benchmark to the algorithm used here. I use two examples used in the ABC-SysBio package (Toni et al. 2009) as well as in Toni (2010).

Algorithm 5 Approximating robustness

k ← number of parameters
i ← marginal prior distribution of each parameter

for each model m of M **do**
    Prior~ $U(a, b)$
    $V_{prior}^{m} = \prod_{i=1}^{k}(i_b - i_a)$

    Get $1^{st} < data < 99^{th}$ percentiles
    if Cuboid calculation **then**
        $V_{post}^{m} = \prod_{i=1}^{k}(i_{max} - i_{min})$
    end if
    if Ellipsoid calculation **then**
        Calculate data covariance matrix
        $V_{post}^{m} = \frac{2\pi^{\frac{k}{2}}}{k\Gamma(\frac{k}{2})}\left[\chi_k^2(\alpha)\right]^{\frac{k}{2}}|\Sigma|^{\frac{1}{2}}$
    end if

    $R^m = \frac{V_{post}^{m}}{V_{prior}^{m}}$
    $R_{norm}^{m} = \frac{R_i^m}{\sum_{i=1}^{M} R_i^m}$
end for

4.5.1 Case study 1: Infectious diseases

As described in Toni et al. (2009), the models used for the first case study describe the spread of an infectious disease through a population over time. The population is made up of susceptible, infected or recovered individuals, denoted as S, I and R respectively. Three models are compared for the robustness of their posterior distributions. The first model (Model 1), is the simplest model of the three. Each individual S or R can be infected once and then it can immediately infect other individuals (Toni et al. 2009).

Model 1:

$$\frac{dS}{dt} = \alpha - \gamma SI - dS$$
$$\frac{dI}{dt} = \gamma SI - \upsilon I - dI$$
$$\frac{dR}{dt} = \upsilon I - dR,$$

where α denotes the birth rate, d the death rate, γ the infection rate, and υ the recovery rate. The second model, Model 2, includes a time delay between an individual getting infected and being infectious. δ denotes the rate of transition of a non-infectious infected individual to an infectious one.

Model 2:

$$\frac{dS}{dt} = \alpha - \gamma SI - dS$$
$$\frac{dL}{dt} = \gamma SI - \delta L - dL$$
$$\frac{dI}{dt} = \delta L - \upsilon I - dI$$
$$\frac{dR}{dt} = \upsilon I - dR,$$

Finally the third model, Model 3, extends Model 1 and includes the recovered individuals being able to become susceptible again. This is denoted by rate e.

Model 3:

$$\frac{dS}{dt} = \alpha - \gamma SI - dS + eR$$
$$\frac{dI}{dt} = \gamma SI - \upsilon I - dI$$
$$\frac{dR}{dt} = \upsilon I - dR - eR,$$

The three models are simulated using ODEs. In ABC-SysBio model selection is used. Parameter inference is also used for each model separately without the use of model

selection. I used the two methods outlined in Algorithm 5 to calculate the robustness of the posterior distributions of all three models resulting from the parameter inference. This robustness measure was then compared to the result of ABC-SysBio model selection of the same models. As shown in Figure 4.2, there is good agreement between the three measures of robustness. The posterior distributions of all three models are also shown in Figure 4.2.

4.5.2 Case study 2: Population growth

The second example I will use to demonstrate the effectiveness of the methods used here for robustness calculation is a population growth model. This is example is also used in the ABC-SysBio package (Toni et al. 2009). The data was obtained by simulating an immigration-death model shown in Equation 4.7. This model (referred to as Model 1) and a model of logistic growth are compared for robustness of their posterior distributions.
Model 1:

$$\frac{dI}{dt} = \alpha - \beta I \tag{4.7}$$

Logistic growth, Model 2:

$$\frac{dI}{dt} = \gamma - I(\delta - \epsilon I) \tag{4.8}$$

As in Section 4.5.1, two analyses were carried out on these two models. First, ABC-SysBio model selection was used to find the most robust model. Then parameter inference was done on each model. The resulting posterior distributions (shown in Figure 4.3), were compared for robustness using the cuboid and the ellipsoid approximation methods. All three robustness measures find that Model 1 is the most robust model. The analysis was repeated for ODE, Markov jump process (MJP) and Stochastic differential equation (SDE) simulations, all arriving to the same result of Model 1 being the most robust. The results are shown in Figure 4.3.

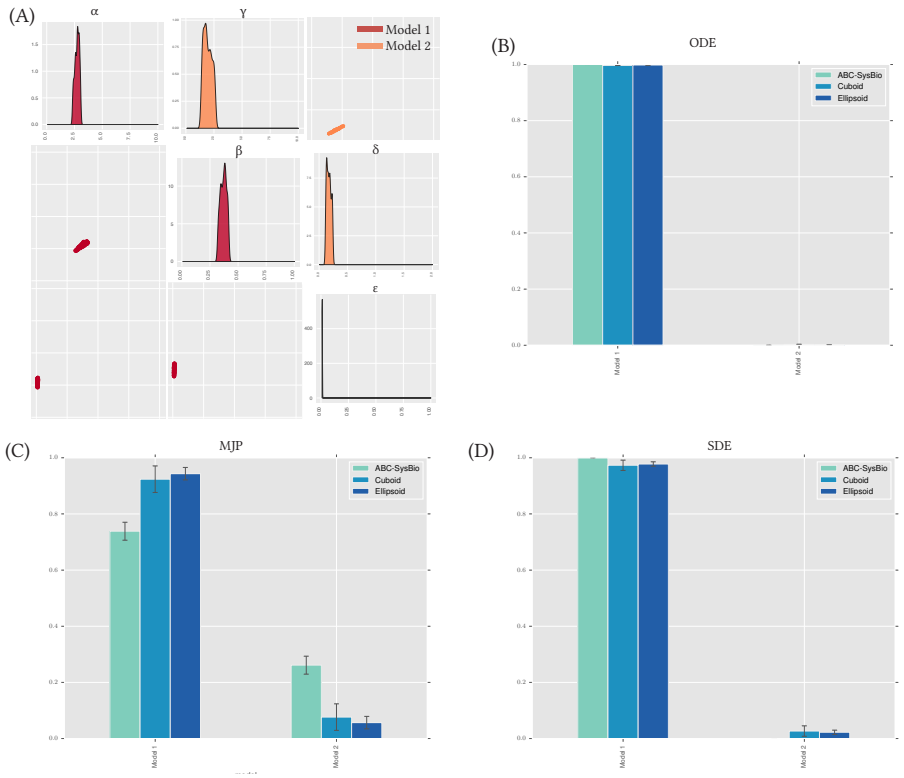

Figure 4.2 Robustness analysis of the three models for the spread of infectious diseases. (A-C) The posterior distributions of the three models compared. (D) I use three methods to calculate robustness, ABC-SysBio model selection, the volume of the hyper-cuboid approximation of the posterior distribution and the volume of the hyper-ellipsoid approximation of the posterior distribution. Each analysis was repeated three times. The height of the bars indicate the mean robustness from the three repeats and the error bars represent the standard deviation. There is good agreement between all three methods. All three methods show that Model 1, the simplest model, is the most robust model.

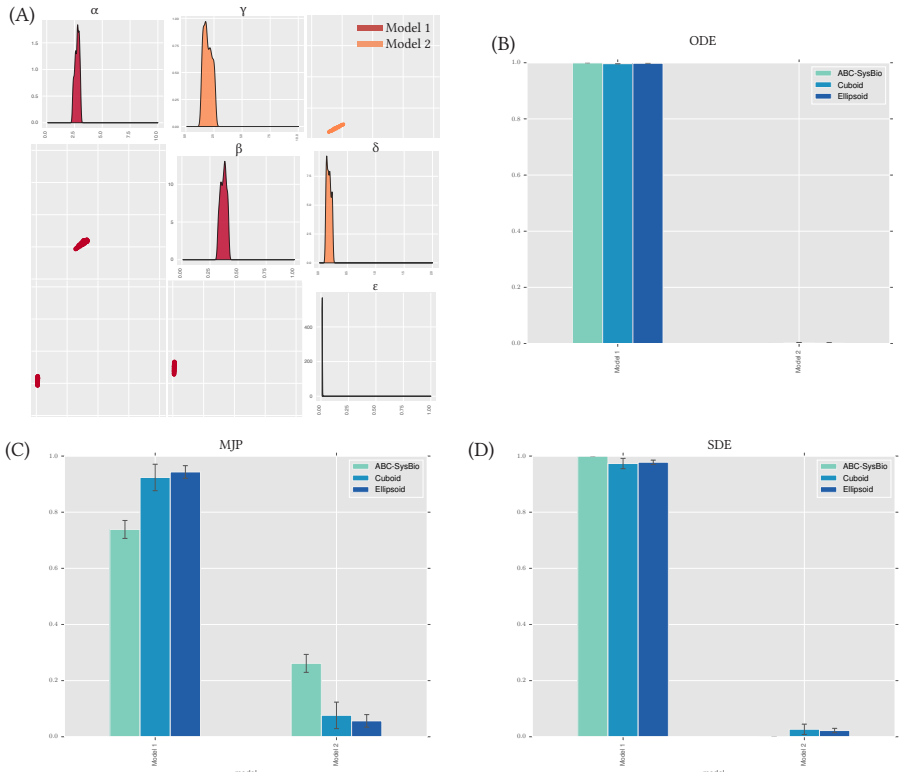

Figure 4.3 Robustness comparison of two population growth models. (A) The posterior distributions of the two models. (B-D) The models were simulated using ODE, MJP and SDE. Both the cuboid and the ellipsoid approximations agree with ABC-SysBio model selection results. Each analysis was repeated three times. The height of the bars indicate the mean robustness from the three repeats and the error bars represent the standard deviation.

The two case studies used above show that the cuboid and the ellipsoid approximation of model robustness agree with the results obtained from ABC-SysBio model selection. A point I must draw attention to is that for ABC-SysBio model selection where model selection is incorporated in the process, each model is also considered a particle with an associated weight (Toni et al. 2009). If a model is performing poorly it does not proceed in the algorithm and is dropped when the weight falls low enough so that the model is not sampled (Toni et al. 2009). This can save time in the analysis as computational resources are not wasted on 'dead' models, models that perform the required behaviour poorly. Using StabilityFinder for model selection, each model must reach the given final ε in order for the cuboid and ellipsoid methods to be valid. This means that time and computational power will be spent on models that are potentially a bad fit, or that have posterior distributions so small compared to the prior that it will take a long time for StabilityFinder to find it. Despite this, the results agree between the all three methods of model selection. This shows that the requirement for all models to reach the final ε does not affect the results for the models used in the above case studies. The potentially wasted computational resources on 'dead' models is a compromise made in order to be able to run the models separately, as model selection is not the primary purpose of StabilityFinder.

4.6 Applications of StabilityFinder

In this section I apply StabilityFinder to toggle switch models in order to find the design principles underlying their stabilities. First I apply it to a simple model with known results, the Gardner, Cantor, & Collins (2000) toggle switch. This model can serve as a test for StabilityFinder, as the conditions for bistability are derived in Gardner, Cantor, & Collins (2000).

4.6.1 StabilityFinder used on the Gardner toggle switch

Gardner, Cantor, & Collins (2000) constructed the first synthetic genetic toggle switch (Gardner, Cantor, & Collins 2000). Their model consisted of two mutually repressing transcription factors, as shown in Figure 4.4, and in the deterministic case is defined by the following ODEs:

$$\frac{du}{dt} = \frac{a_1}{1 + v^\beta} - u \tag{4.9}$$

$$\frac{dv}{dt} = \frac{a_2}{1+u^\gamma} - v, \tag{4.10}$$

where u is the concentration of repressor 1, v the concentration of repressor 2, a_1 and a_2 denote the effective rates of synthesis of repressors 1 and 2 respectively, β is the cooperativity of repression of promoter 1 and γ of repressor 2. Gardner, Cantor, & Collins (2000) studied the deterministic case and concluded that there are two conditions for bistability for this model; that a_1 and a_2 are balanced and that β, γ >1 (Gardner, Cantor, & Collins 2000). I test StabilityFinder by using it to find the posterior distribution for which this model exhibits bistable behaviour. Therefore, the desired behaviour is set to two steady states, and using a wide range of values as priors as shown in Table 4.1, I used StabilityFinder to find the parameter values necessary for bistability to occur. The posterior distribution calculated by StabilityFinder for the Gardner deterministic case is shown in Figure 4.5.

Table 4.1 Gardner switch priors in the deterministic and stochastic cases

Parameters				Species	
a_1	β	a_2	γ	s_1	s_2
0-60	0-5	0-60	0-5	0-100	0-100
μM time^{-1}	μM^{-1} time^{-1}	μM time^{-1}	μM^{-1} time^{-1}	(μM)	(μM)

These results agree with the results reported by Gardner, Cantor, & Collins (2000). For this switch to be bistable a_1 and a_2 must be balanced while β and γ must both be >1, as can be seen in the marginal distributions of β and γ in Figure 4.5A.

I next applied StabilityFinder to the case of the Gardner switch under stochastic dynamics using the same priors as the deterministic case, and again searched the parameter space for bistable behaviour. The posterior distribution is shown in Figure 4.5B. We can see that the conditions on the parameters required for bistability in the deterministic case generally still stand in the stochastic case. There appears to be slightly looser requirements on the parameters of the stochastic model (wider marginal distributions). Some difference between the deterministic and stochastic posteriors is expected as different clustering algorithms are used for the stochastic and the deterministic cases. The Gap statistic is used in the case of the stochastic case, as it is capable of dealing with noisier data whereas a simpler and faster algorithm is used for clustering the deterministic solutions. These results demonstrate that StabilityFinder can be used to find the parameter values that can produce a desired stability and can be confidently applied to more complex models.

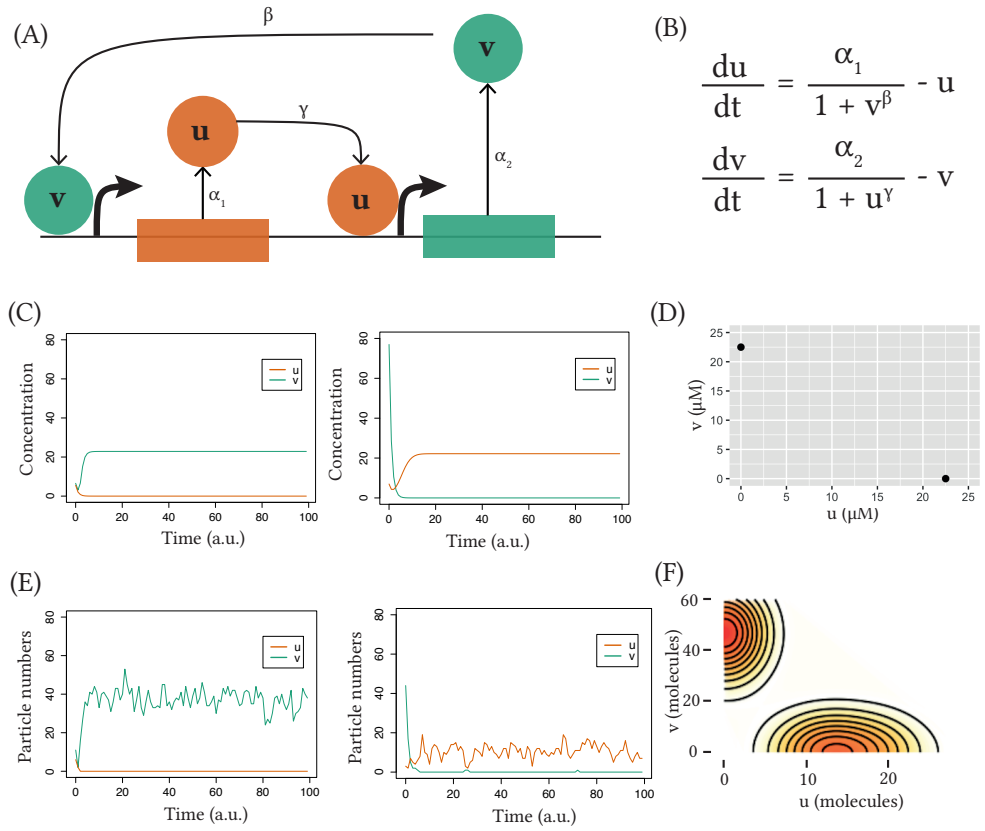

Figure 4.4 The Gardner switch model used to test StabilityFinder. The Gardner model (A) consists of two mutually repressing transcription factors. (B) It can be reduced to a two-equation system, where u and v are the two transcription factors, a_1,a_2 are their effective rates of synthesis, u, v are their concentrations and β, γ represent the cooperativity of each promoter. (C) Two samples of deterministic simulated time courses of the Gardner switch and (D) The resulting phase plot. (E) Two samples of time courses of the stochastic simulations and (F) the resulting stationary distributions.

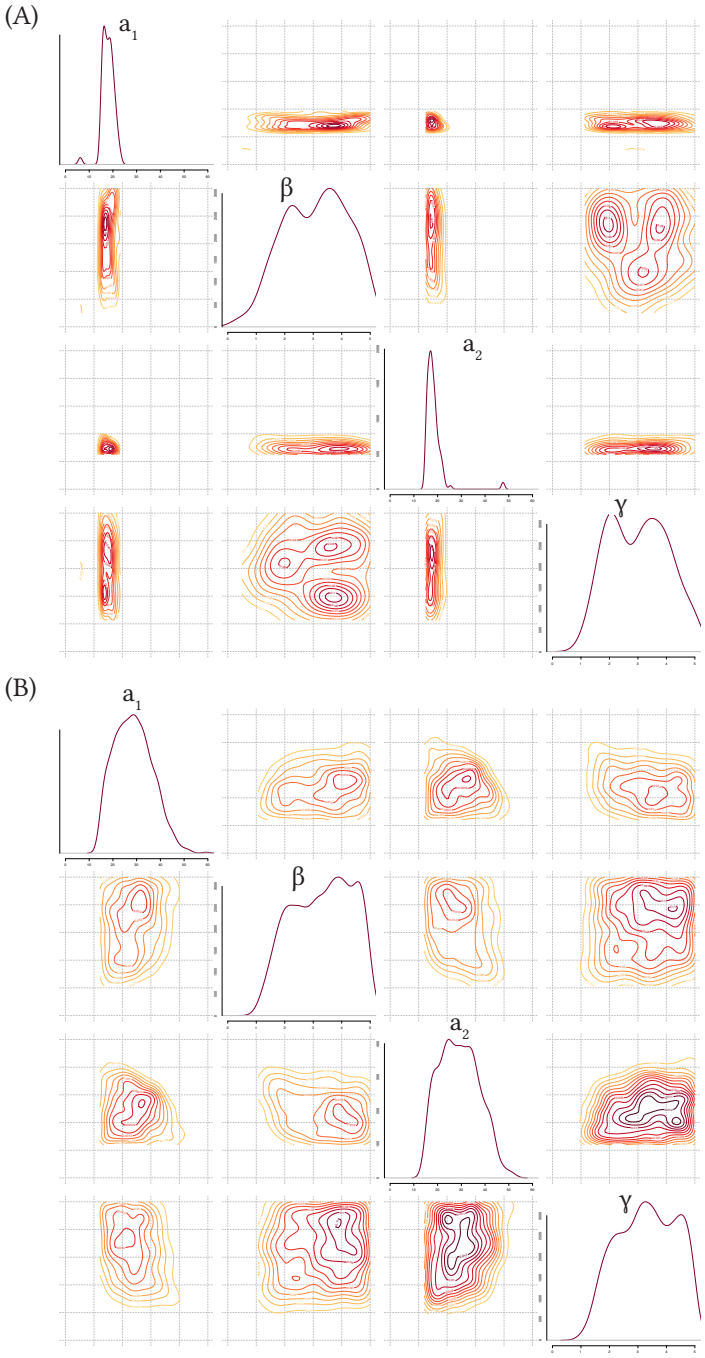

Figure 4.5 Elucidating the stability of the Gardner switch. The Gardner model has four parameters, for which I want to find the values for which this system is bistable. I use StabilityFinder to find the posterior distribution of the bistable Gardner switch, deterministically (A) and stochastically (B). The posterior distributions are shown as the density plots of each parameter as well as each one plotted against the other.

4.6.2 Lu toggle switch models

Next I analyzed an extension of the Gardner switch model developed by Lu, Onuchic, & Ben-Jacob (2014). I use these models as they are of increased complexity from the Gardner model. Lu, Onuchic, & Ben-Jacob (2014) considered two types of switches, the classic switch consisting of two mutually repressing transcription factors (model CS-LU), as well as a double positive switch DP-LU. The CS-LU switch was found to be bistable given the set of parameters used, while the DP-LU switch was found to be tristable (Lu, Onuchic, & Ben-Jacob 2014). The CS-LU model used in their study is given by the following system of ODEs, as given in Lu, Onuchic, & Ben-Jacob (2014).

$$\dot{x} = g_x H^S_{xy}(y) - k_x x \tag{4.11}$$

$$\dot{y} = g_y H^S_{yx}(x) - k_y y, \tag{4.12}$$

where:

$$H^S_I(x) = H^-_I(x) + \lambda_I H^+_I(x) \tag{4.13}$$

$$H^-_I(x) = 1/\left[1 + (x/x_I)^{n_I}\right] \tag{4.14}$$

$$H^+_I(x) = 1 - H^-_I(x), \tag{4.15}$$

and the DP-LU model is given by

$$\dot{x} = f_x(x,y) = g_x H^S_{xy}(y) H^S_{xx}(x) - k_x x \tag{4.16}$$

$$\dot{y} = f_y(x,y) = g_y H^S_{yx}(x) H^S_{yy}(y) - k_y y, \tag{4.17}$$

g_I represents the production rate, k_I the degradation rate, n_I the Hill coefficient, x_I the Hill threshold concentration and λ_I the fold change of the transcription rates, and $I \in \{xy, yx, xx, yy\}$.

For the parameter values used in the Lu study, the CS-LU switch exhibits three steady states, two of which are stable and one is unstable. The CS-LU switch exhibits five steady states, of which three are stable and two are unstable. Bifurcation diagrams of the two Lu models are shown in Figure 4.6.

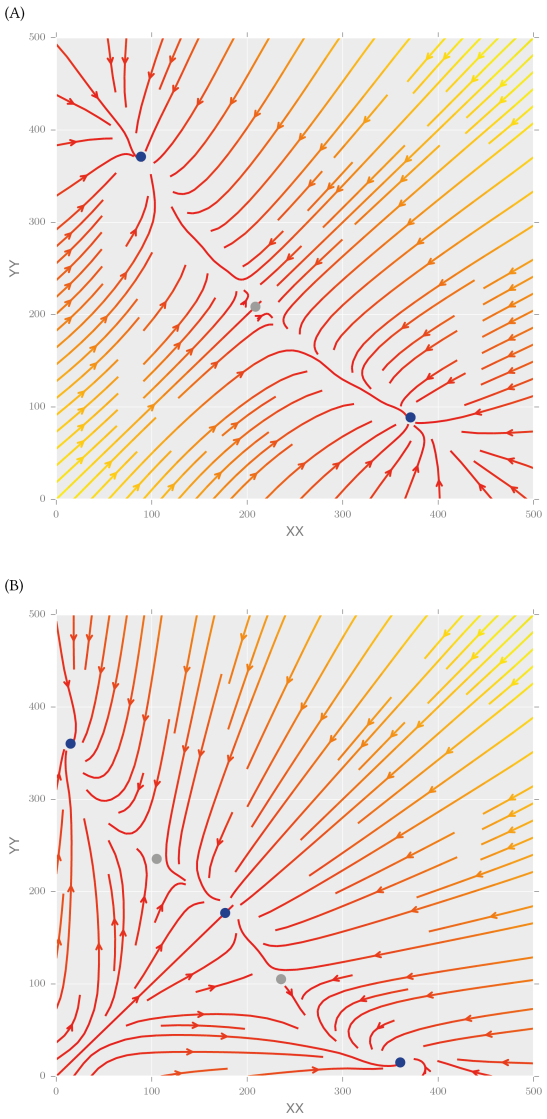

Figure 4.6 Stream plot of the vector plot of the (A) CS-LU and (B) DP-LU switches. The colours indicate the magnitude of the vectors, with yellow indicating high and red low values. The blue points represent stable steady states and the grey points represent unstable steady states.

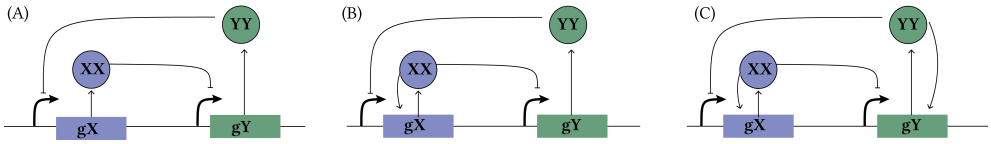

Figure 4.7 The three LU toggle switch models. (A) CS-LU, (B) SP-LU and (C) DP-LU.

4.6.2.1 Extending the Lu models

I start the analysis of the Lu models by extending their analytical approach to solving the system. I use StabilityFinder to explore a larger parameter space which allows us to distinguish between rare events and robust behaviours. The advantage of using StabilityFinder over solving the system analytically is that the full parameter space is explored rather than solving the system for a single set of parameters. This allows us to deduce model properties that could not otherwise be identified. Robustness to parameter fluctuations can be explored, as well as parameter correlations and restrictions on the values they can take while still producing the desired behaviour.

It is known that that the addition of positive autoregulation to the classical toggle switch can induce tristability (Lu, Onuchic, & Ben-Jacob 2014). Here I investigate the interplay of positive autoregulation on the values of the other parameters in the model. I extended the analysis presented in Lu, Onuchic, & Ben-Jacob (2014) by including the switch with single positive autoregulation (model SP-LU), where an asymmetry of positive feedbacks is present between the two genes. The three switches considered in this analysis are shown in Figure 4.7. The SP-LU switch is modelled using the following ODE system

$$\dot{x} = g_x H^S_{xy}(y) H^S_{xx}(x) - k_x x \tag{4.18}$$

$$\dot{y} = g_y H^S_{yx}(x) - k_y y. \tag{4.19}$$

Using StabilityFinder with priors centred around the parameter values used in the original paper (see Table 4.2), we can identify the most important parameters for achieving the models' stability. The phase plots of the final populations of the models are shown in Figure 4.8 and the posterior distribution of these models are shown in Figure 4.9A. We find that the parameters representing the rates of degradation of the transcription factors in the system (k_x, k_y) must both be large in relation to the prior ranges for bistability to occur. Protein degradation rates have been

shown to be important for many system behaviours including oscillations (Woods et al. 2016).

Table 4.2 Priors of the classical (CS-LU), single positive (SP-LU) and double positive (DP-LU) models.

Parameter	Symbol	CS-LU	SP-LU	DP-LU
Production rate	gx	30-50	1-2	1-100
(Proteins/Minute)	gy	30-50	20-25	1-100
Degradation rate	kx	0-0.5	50-55	0-1
($Minute_{-1}$)	ky	0-0.5	48-52	0-1
Hill coefficient	nxy	1-5	30-35	0-10
	nyx	1-5	0.1-0.2	0-10
Hill thresholds	xxy	100-300	2-3	100-1000
concentration (Proteins)	xyx	100-300	0.4-0.6	100-1000
Transcription rate fold	lxy	0-0.5	0.02-0.04	0-1
change	lyx	0-0.5	0.02-0.04	0-0.2
Hill coefficient	nXX	-	25-30	0-10
	nYY	-	0.01-0.02	0-10
Hill thresholds	xXX	-	0.4-0.5	50-500
concentration (Proteins)	xYY	-	1-3	50-500
Transcription rate fold	lXX	-	65-72	1-20
change	lYY	-	0.02-0.04	1-20

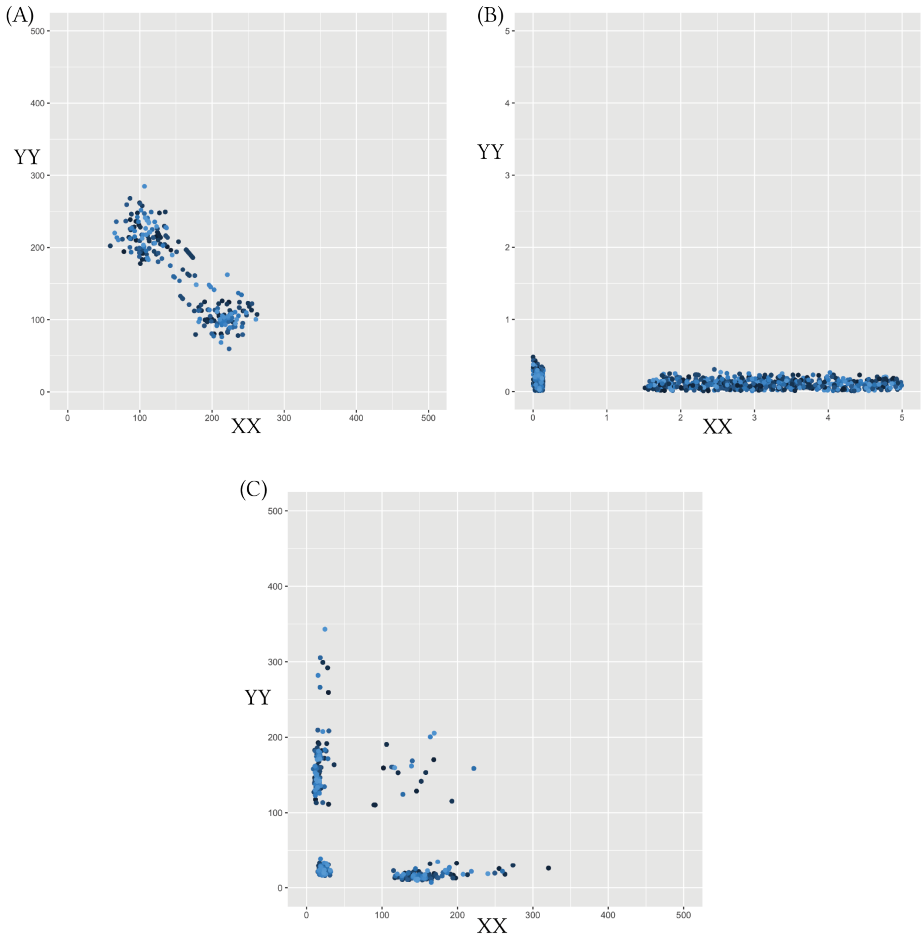

Figure 4.8 The phase plots of 100 particles from the last population of the three Lu switches. (A) The bistable CS-LU (B) The bistable SP-LU and (C) The tristable DP-LU. There are two types of tristable behaviour, one where the third steady state is zero-zero and one where the third state is high (non-dead).

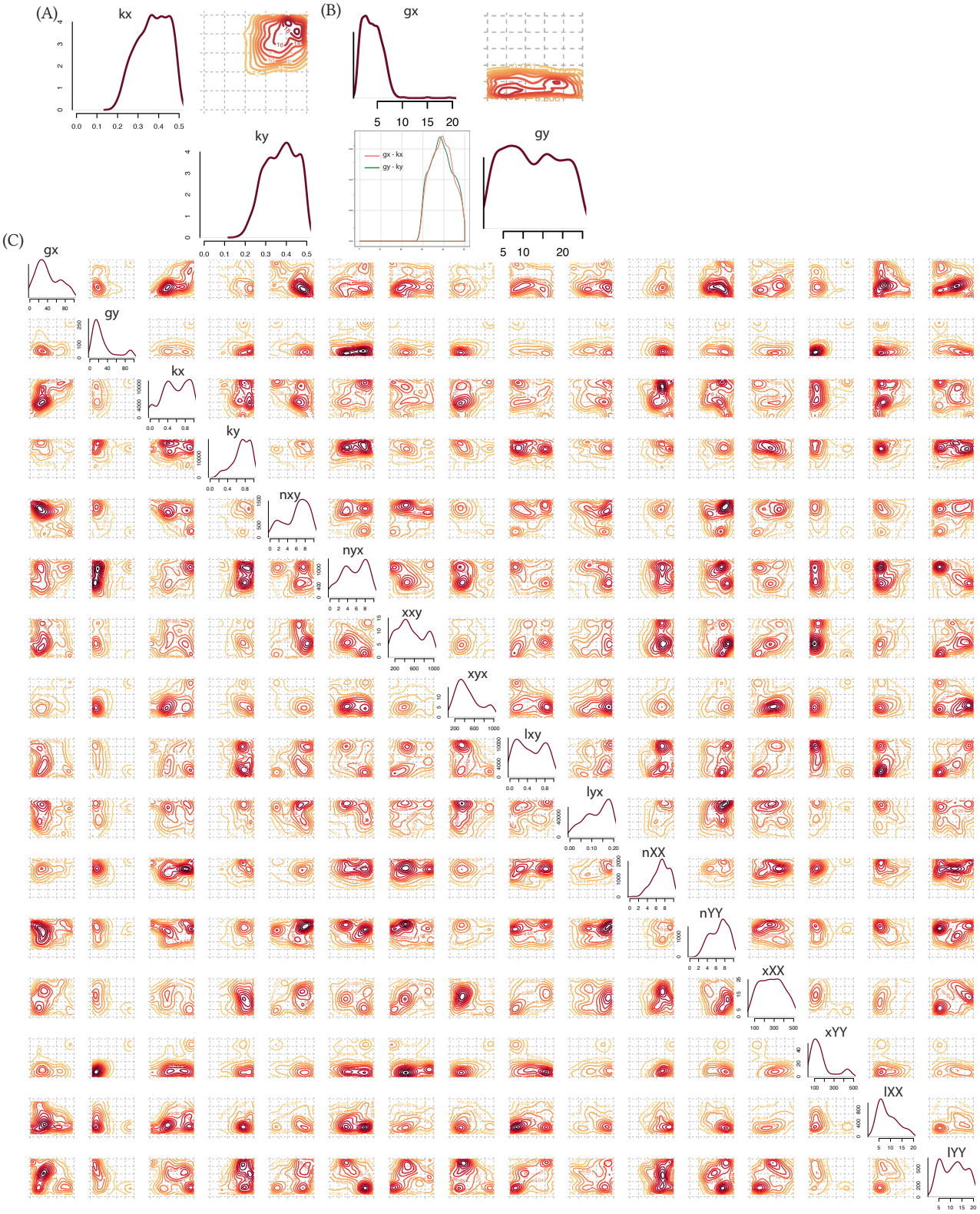

Figure 4.9 The three variants of the Lu models. (A) The CS-LU switch is bistable. The most restricted parameters for this behaviour are kx and ky which both have to be high relative to the prior. (B) The extended Lu model with a single positive autoregulation on X. This model is bistable when gx is small, but the net production of protein is equal for the two nodes. (C) The Lu model with double positive autoregulaiton is tristable, and its posterior distribution shown here.

We find that the switch with single positive autoregulation is capable of bistable behaviour as seen in Figure 4.9B, but this is only possible when the strength of the promoter under positive autoregulation, gx, is small (Figure 4.9). There appear to be no such constraints on the strength of the original, unmodified, promoter, gy.

Upon examination of the DP-LU model, we also find that tristability in the switch is relatively robust, as tristability is found across a large range of parameter values, with no parameters strongly constrained. Two types of tristable behaviour are identified, one where the third steady state is at (0,0) and and one where the third steady state has non-zero values, as seen in Figure 4.8. This result agrees with previous work by Guantes & Poyatos (2008), who found that a switch can exhibit two kinds of tristability, one in which the third steady state is high (III_H) and one in which it is low (III_L) (Guantes & Poyatos 2008).

4.6.2.2 Multistability in the Lu models

The DP switch is capable of both bistable and tristable behaviour as well as 4 co-existing states under deterministic dynamics (quadristability) (Guantes & Poyatos 2008). It is of great interest to understand the conditions under which these three behaviours occur. A bifurcation analysis of the DP switch was carried out using the PyDSTool (Clewley 2012) in order to get an indication of the stabilities this model is capable of, and at which parameter ranges these are found.

Since the Lu models can be solved analytically, the bifurcation diagram of the DP-LU can be obtained by keeping all parameters constant apart from gene expression (gx). The result shown in Figure 4.10B, the system can exhibit 2, 3 or 4 steady states depending on the value of the gene expression rate. We observe that if $100 \leq gx \leq 120$ the system exhibits four steady states, if $9 \leq gx \leq 10$ the system is tristable and if $10 \leq gx \leq 100$ the system is bistable. I use the whole range tested above ($0 \leq gx \leq 140$) as prior distributions in StabilityFinder and searched parameter space for 2, 3 and 4 steady states.

Using StabilityFinder a more complex picture of the parameter space that can produce each behaviour can be obtained. This is because, unlike the bifurcation analysis, StabilityFinder does not require any of parameters to be fixed. Since there are no such restraints on the value each parameter can take we obtain a bigger range of parameters that can produce each behaviour than the ranges found during the bifurcation analysis. The priors used for each analysis are identical and include the whole range of values found in the bifurcation diagram, varying only the required number of steady states. In addition, unlike the bifurcation analysis the values for

gx and gy are not forced to be equal in the analysis done on StabilityFinder.

Using StabilityFinder, the posterior distributions for bistable, tristable and quadristable behaviours in the DP-LU model were obtained and then the posterior parameter distributions compared (Figure 4.10). Upon examination of the posterior distributions for all three switches we observe that a subset of the posterior parameter values is different under the three behaviours. Differences are found in the univariate distribution of the parameters for gene expression, gx, as highlighted in Figure 4.10, box 1. This parameter must be small for a quadristable switch to occur but there are no such restraints for a bistable or a tristable switch. Furthermore, parameter xXX must be small for three and four steady states to be achieved but there are no such restraints for a bistable switch, as can be seen in Figure 4.10, box 2. Parameter xXX represents the Hill threshold concentration, and is equivalent to to the Hill constant described in Section 2.2.2.1. This parameter dictates the substrate concentration at which the switch occurs. We find that the Hill constant has to be small in order to observe three or four steady states.

We also find a difference in the bivariate distributions in the posterior. Most notably, we find that parameters xXX and gX are tightly constrained in the tristable and the four steady state cases, where both parameters are required to be small, but less so in the bistable case (Figure 4.10, box 3). Another notable difference is between parameters xXX and nXX shown in Figure 4.10, box 5, where they are constrained in the tristable and quadristable cases but not the bistable case. There parameters represent the Hill constant and the Hill coefficient respectively. The Hill constant dictates the substrate concentration that results in half of the response, i.e. it is substrate concentration at which switching is observed. The Hill coefficient affects the steepness of the switching curve, as a higher Hill coefficient results in a steeper response, as illustrated in Figure 2.2.

Interestingly, we also find parameter correlations conserved between the three behaviours, as seen in Figure 4.10, box 4, where parameters lXX and gx, positive autoregulation and gene expression are negatively correlated in both cases. This highlights the importance of treating unknown parameters as distributions rather than fixed values when studying the parameter values of a model, as they are capable of uncovering not only the ranges and values needed but also the correlations between parameters that would not have otherwise been detected.

I further analyse these models by studying the phase plots resulting from simulating the particles from the posterior distribution to steady state. The phase plots from 100 particles from each posterior are shown in Figure 4.11. We find that there

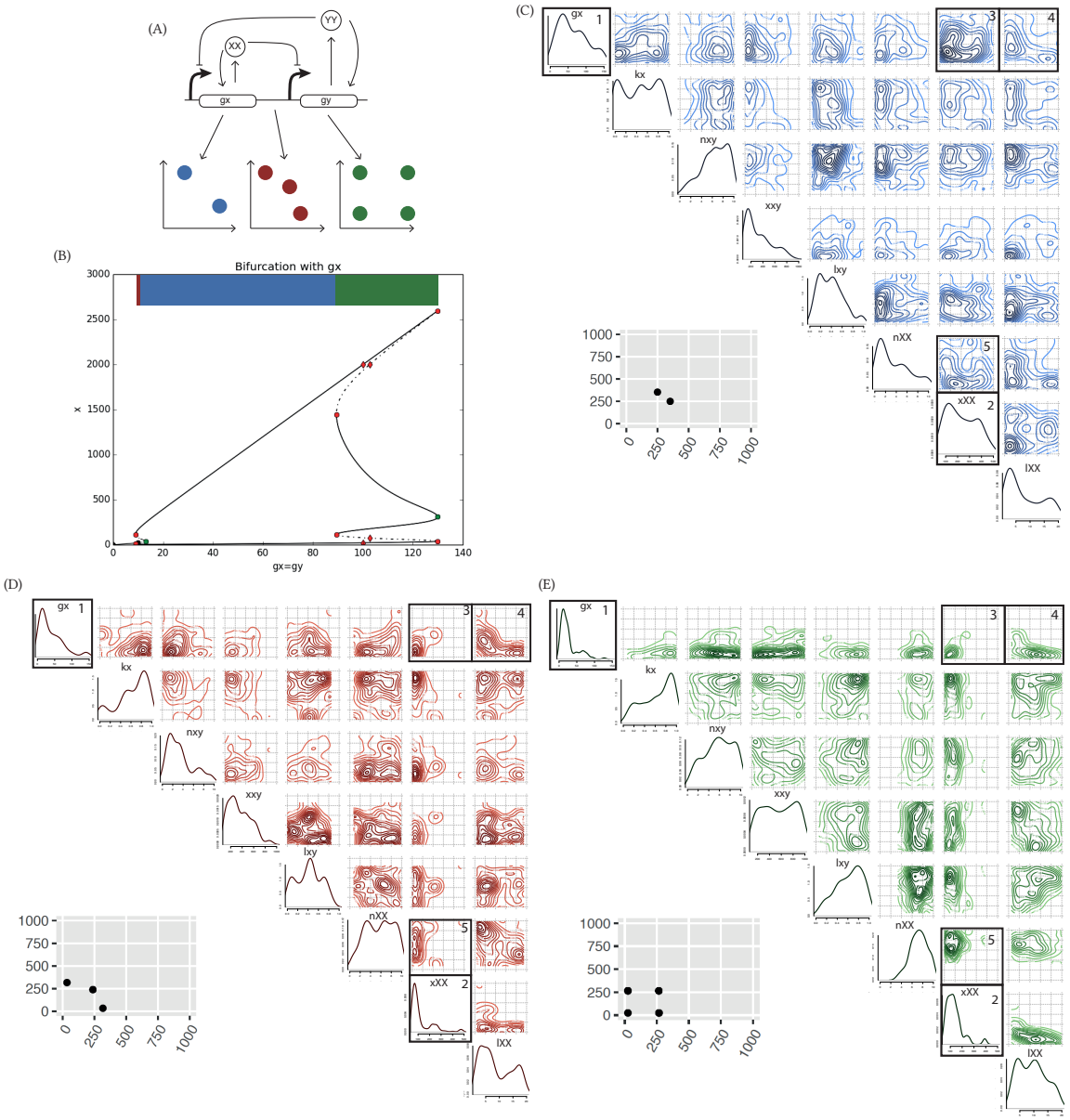

Figure 4.10 Design principles of multistable switches. (A) Using the Lu model with added positive autoregulation we uncover the design principles dictating if a switch will be bistable, tristable, or will have 4 steady states. (B-D) By considering the bivariate distributions of the parameters we can uncover the differences in the parameters of a bistable switch compared to a tristable switch, compared to a quadristable switch . The posterior distribution of the bistable switch is shown in blue, of the tristable switch in red and of a quadristable in green. The bivariate distributions for which a difference is observed between the stabilities are in black boxes. An example of a phase plot from each behaviour is shown next to the corresponding posterior distribution.

is a strong conservation on the locations of the steady states between each particle. This indicates that the steady states in a two-node toggle switch tend to be symmetrical. This gives rise to the patterns seen in Figure 4.11. It is important to highlight that this was not set as a requirement for the behaviour of the switch in StabilityFinder, but the behaviour that these models gave rise to. There were no constraints on the level or location of the steady states.

The symmetrical steady states are especially evident in the quadristable switch. For every steady state at (0, 0) there is another steady state on its diagonal, at $XX = YY$. All the combinations of these two steady states form the straight line seen in Figure 4.11C. This indicates that two of the four steady states exist where $XX = YY$. The other two exist where one of the two proteins dominates the other. There are four distinct states of the system: both proteins high, both low, XX high/YY low and XX low/YY high.

This same principle can be seen in the bistable and the tristable switches. In the bistable switch the two steady states are also symmetrical and one never completely dominates the other. For the tristable case we observe that two of the steady states exist where the levels of one protein is much larger than the other, and a third steady state exists where $XX = YY$. This finding can be exploited in a synthetic biology application. Building a switch whose states are always symmetrical makes it easy to distinguish which state the system is in. By measuring one of the proteins in the system it can be inferred what the levels of the other are. We also observe that the third steady state is not necessarily a 'dead' state, but they can exist over a range of values for XX and YY.

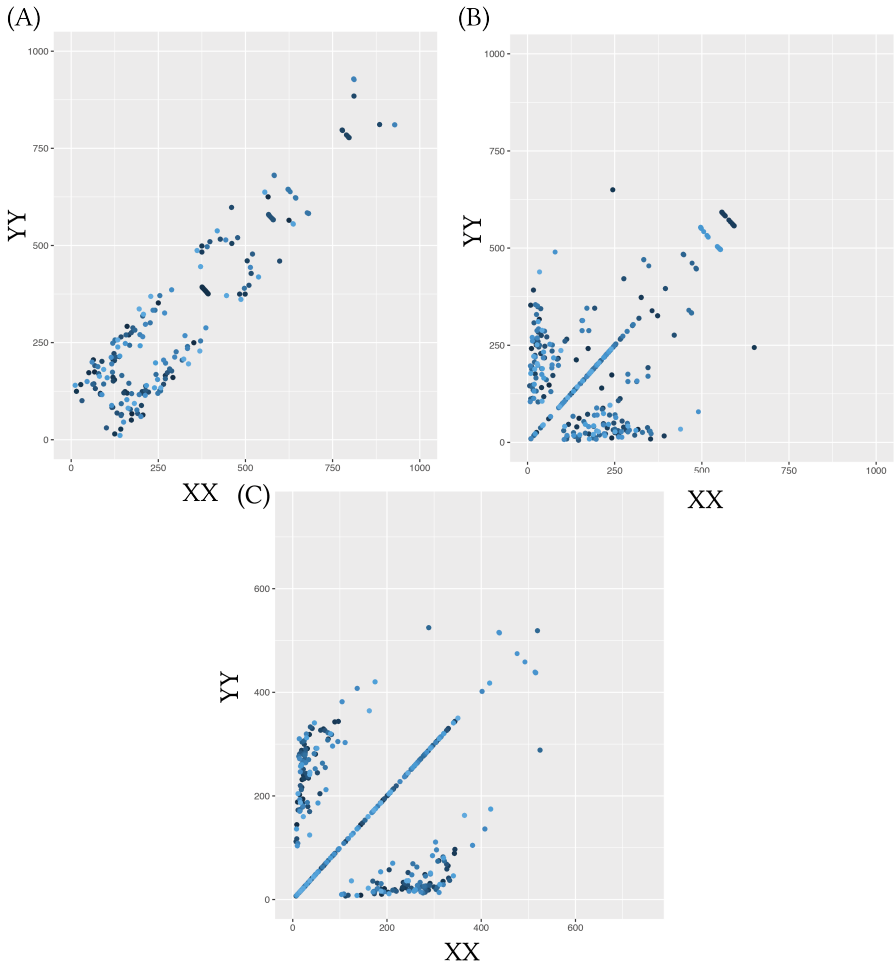

Figure 4.11 The phase plots from 100 particles from each posterior. (A) The CS-LU (B) SP-LU and (C) DP-LU. Each particle is represented by a different shade of blue. There is a strong conservation on the location of the steady states between particles.

4.6.2.3 Extending the Lu switch to three nodes

To further demonstrate the flexibility of StabilityFinder I investigated a system capable of higher stabilities. Multistability is found in differentiating pathways, like the myeloid differentiation pathway (Ghaffarizadeh, Flann, & Podgorski 2014; Cinquin & Demongeot 2005). I allow for these more complex dynamics by extending the DP-LU model by adding another gene, making it a three gene switch. This new system is depicted in Figure 4.12A. This model has symmetric parameters, which means that the parameters for equivalent reactions (e.g. gene expression) are the same. In StabilityFinder I look for six steady states, the output being in nodes X and Y and using the priors shown in Table 4.3. The system is capable of six steady states, as shown in Figure 4.12C.

Table 4.3 Priors used in the three-node switch

Parameter	Symbol	Range
Production rate (Proteins/Minute)	gx	3-5
Degradation rate (Minute^{-1})	kx	0-0.2
Hill coefficient	nxy	0-2
Hill thresholds concentration (Proteins)	xxy	140-160
Transcription rate fold change	lxy	0-0.2
Hill coefficient	nxx	2-4
Hill thresholds concentration (Proteins)	xxx	90-110
Transcription rate fold change	lxx	8-12

We find that the most constrained parameters for this behaviour are again the degradation rate of the proteins, kx. If they are too large or too small the system will not exhibit hexa-stability. Additionally we find that the Hill coefficients for the repressors, nxy, are constrained to be smaller than 1.5 as seen in Figure 4.12D.

Consistently with the results found in Section 4.6.2.2, we find that the steady states are symmetric (Figure 4.12B). Each of six steady states exists in symmetry with another one, in tightly constrained regions. This example demonstrates that StabilityFinder can be used to elucidate the dynamics of more complex network architectures, which will be key to the successful design and construction of novel gene networks as synthetic biology advances.

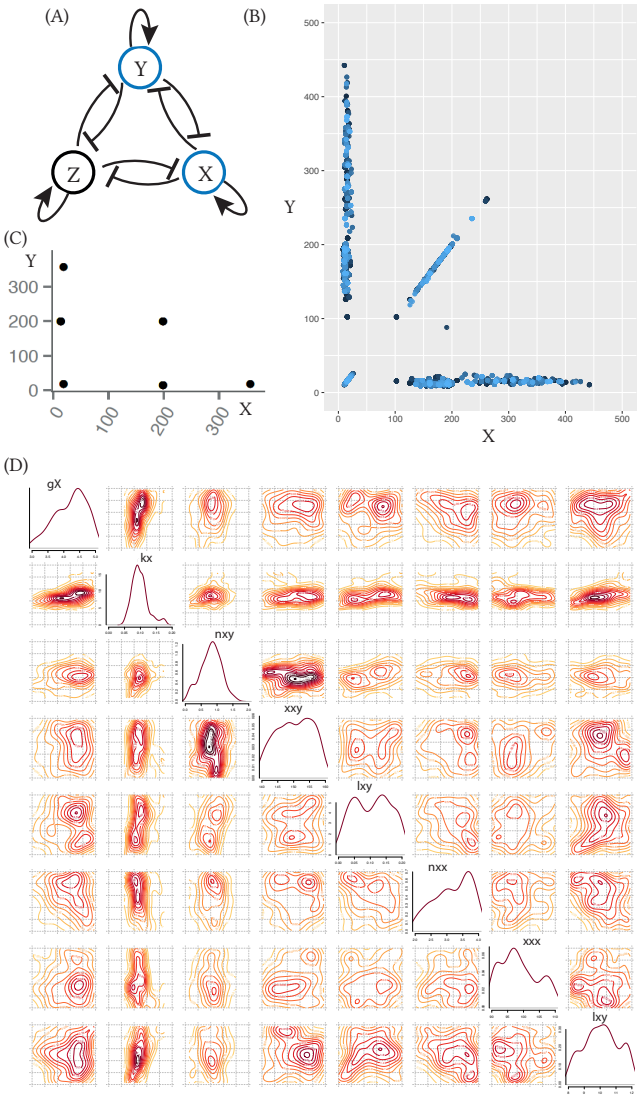

Figure 4.12 The three-node mutual repression model, with added positive autoregulation on each node. (A) The model. The model is studied in two dimensions using StabilityFinder, for nodes X and Y. (B) The phase plot of 100 particles from the posterior found by StabilityFinder. (C) An example phase plot from one particle. There are 6 steady states. (D) The posterior distribution of the 6-steady state three-node system. Parameters kx and nxy are the most constrained.

4.6.3 StabilityFinder used on the more general mass action switches

In order to study the switch system in a more realistic way, I developed an extension to the switches used in Sections 4.6.1 and 4.6.2. This new set of switches does not use the quasi-steady state approximation (QSSA) that is often used in modelling the toggle switch. Using mass action, this changes the two-equation system used in Gardner, Cantor, & Collins (2000) and Lu, Onuchic, & Ben-Jacob (2014) into a system of 8 ODEs and 10 parameters in the classical switch case with no autoregulation (model CS-MA). The equations describing the system are shown below.

$$\begin{gathered}
gA \xrightarrow{geA} gA + A \\
gB \xrightarrow{geB} gB + B \\
A + A \xrightarrow{dim} A_2 \\
A_2 \xrightarrow{dim\,r} A + A \\
B + B \xrightarrow{dim} B_2 \\
B_2 \xrightarrow{dim_r} B + B \\
gA + B_2 \xrightarrow{repA} B_2gA \\
B_2gA \xrightarrow{rep_r} B_2 + gA \\
gB + A_2 \xrightarrow{repB} A_2gB \\
A_2gB \xrightarrow{rep_r} A_2 + gB \\
A \xrightarrow{deg} \emptyset \\
A_2 \xrightarrow{deg_dim} \emptyset \\
B_2 \xrightarrow{deg_dim} \emptyset
\end{gathered}$$

For the model with added double positive autoregulation (model DP-MA) the following equations are added to the system:

$$\begin{gathered}
A_2 + gA \xrightarrow{aut_1} A_2gA \\
A_2gA \xrightarrow{aut_2} A + A_2gA \\
A_2gA \xrightarrow{aut_3} A_2 + gA \\
B_2 + gB \xrightarrow{aut_1} B_2gB \\
B_2gB \xrightarrow{aut_2} B + B_2gB \\
B_2gB \xrightarrow{aut_3} B_2 + gB
\end{gathered}$$

The ODEs describing the above switches are shown in Appendix A. These models are too complex to be solved analytically and I use StabilityFinder to fit the mod-

els to a bistable behaviour using the prior distributions shown in Table 4.4. The prior values were chosen using the parameter scan used in Chapter 3. All priors given assume a uniform distribution. The two models used and the resulting phase plots are shown in Figure 4.13 and the posterior distributions obtained are shown in Figure 4.14.

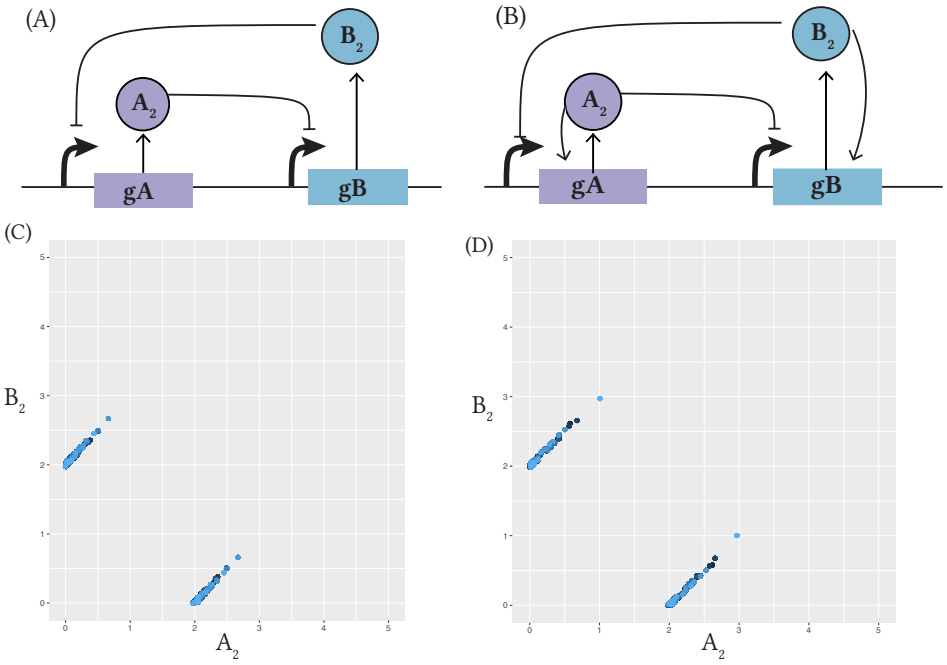

Figure 4.13 Illustrations of the two mass action switches I developed. (A) The simple switch CS-MA (B) The switch with double positive autoregulation DP-MA. (C, D) The phase plots of 100 particles simulated from the posterior distributions of the bistable mass action switches.

By examining the posterior distributions shown in Figure 4.14 we see that the CS-MA is much more constrained that the DP-MA switch. We find that gene expression must be low for bistability to occur in the CS-MA model but there is no such constraint in the DP-MA model. We also find that the monomerization rate dim_r and the monomer degradation rate deg must both be larger than 2. This is not found in the DP-MA model.

Next I compare the two models for robustness using the ellipsoid method described in Section 4.5. We find that the addition of positive feedback loops greatly increases the system's robustness to parameter fluctuations as seen in Figure 4.14A,

Table 4.4 The priors used in the classic (CS) and double positive (DP) mass action deterministic and stochastic models

Description	Parameter	Models						
		Deterministic		Stochastic				
				CS			DP	
		CS	DP	Bistable	Tristable	Bistable	Tristable	
Gene expression	geA, B	5-10	1-10	0-10	0-10	0-10	0-10	
Dimerization	dim	5-10	1-10	7-15	0-3	7-15	0-3	
Monomerization	dim_r	0-5	0-5	0-10	0-10	0-10	0-10	
Repression	repA, B	2-10	1-10	0-10	0-10	0-10	0-10	
Dissociation of repressor	rep_r	0-5	1-10	0-10	0-10	0-10	0-10	
Protein degradation	deg	1-5	0-10	0-10	0-10	0-10	0-10	
Dimer degradation	deg_dim	0-0.1	0-0.5	0-1	0-1	0-1	0-1	
Dimer promoter self-association	aut_1		1-10			0-10	0-10	
Dimer promoter self-activation	aut_2		5-10			0-10	0-10	
Dimer promoter self-dissociation	aut _3		1-5			0-10	0-10	

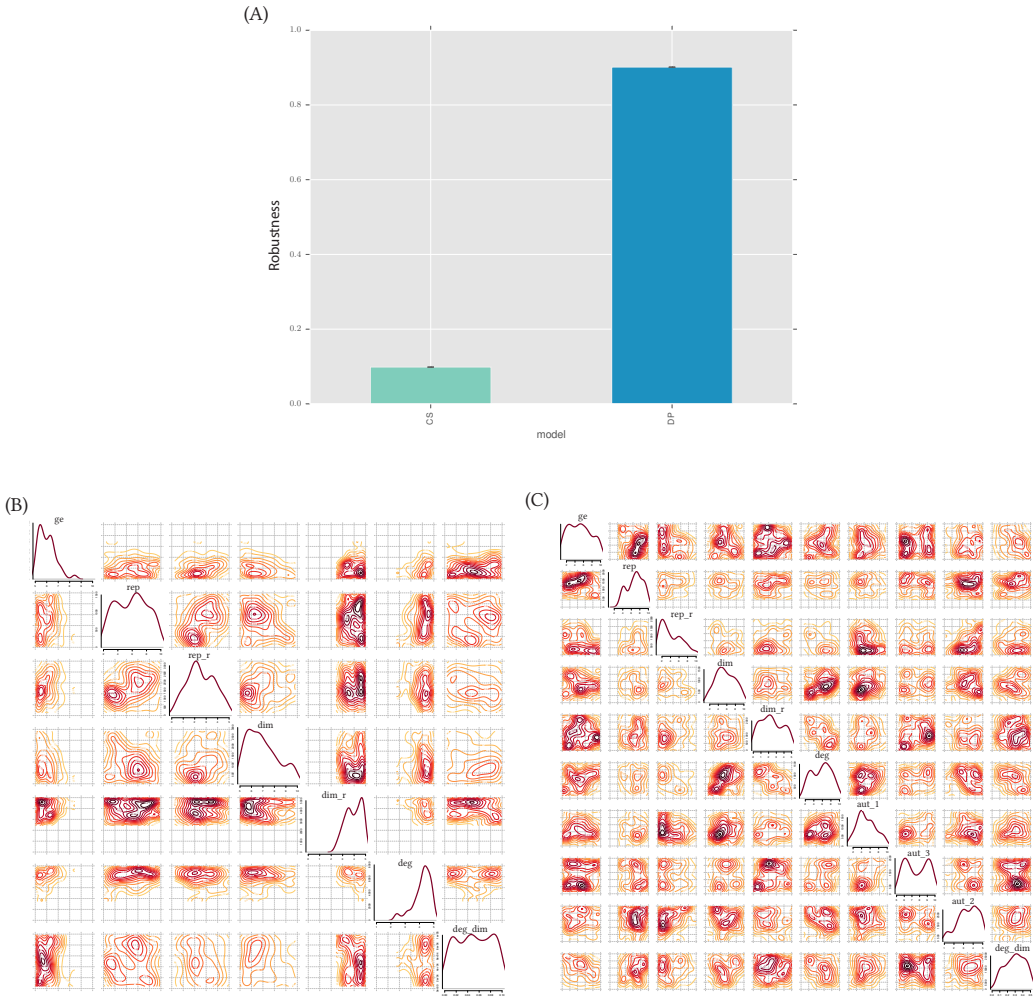

Figure 4.14 (A) Robustness comparison of the CS-MA and DP-MA switches. The Bayes factor of $\frac{p(B|CS)}{p(B|DP)}$ was found to be 9.14. The posterior distributions of (B) CS-MA and (C) DP-MA switches.

The Bayes factor of $\frac{p(B|CS)}{p(B|DP)}$, where B is bistable behaviour, was found to be 9.14. Adding positive feedback loops to the model allows it to be bistable over a greater range of parameter values. This indicates that small fluctuations in parameters in the cellular environment will not revert it to monostability and thus makes it more suitable for use in synthetic biological applications where inconsistent stability profile of a system could be detrimental. This makes it a better candidate for building new synthetic devices based on the toggle switch design. We identified the parameter region within which these models are bistable, information that is important when building such a device in the lab.

The models used in the above analysis assume the parameters for equivalent reactions are equal. This is a constraint that simplifies the model. When building this model into a synthetic system in the lab, this assumption is not necessarily justified. When choosing promoters to build this synthetic system two promoters can be chosen to have similar strength but their strength will not necessarily be identical. In order to study how this might affect the results, I further eliminate modelling assumptions made in the toggle switch by making the parameters representing gene expression (ge) and repression (rep), as well as the protein degradation parameters asymmetric (independent parameters for each protein, versus fixed to be equal). We find that the features of the posterior distributions of the symmetric and the asymmetric models remain the same. The reader is referred to Appendix D for the posterior distributions of the asymmetric models.

I further study the asymmetric mass action models by examining the QSSA. As stated in Section 3.3.1, the QSSA is a common analytical tool for model simplification. By examining the posterior distributions of the CS-MA we can determine whether the QSSA is justified in these models. As stated in Section 3.3.1, the assumption that has to be made for the QSSA to hold is that the rate of binding and unbinding of the transcription factors to the promoters is very fast. The rates have to be much faster than the rates of their production and decay in the system in order to justify that the reaction take place in separate time scales and can thus be assumed to always be at steady state. The QSSA is also made for the dimerization of the transcription factors. Therefore, for the QSSA to be justified in the toggle switch, the rates for association (rep) and dissociation (rep_r) of the transcription factors to the promoters, as well as the rates for dimerization (dim) and monomerization (dim_r) of the transcription factors have to be much larger than any other rate.

In order to determine whether this is the case here, I plot the marginal distribu-

tions of each of the parameters assumed to be very large (rep, rep_r, dim, dim_r) against each of the parameters involved in the expression and decay of the proteins. This is shown in Figure 4.15. We find that the QSSA can be justified only with respect to the rate of degradation of the transcription factor dimers (deg_dim). All parameters under consideration for the QSSA (rep, rep_r, dim, dim_r) are found to be much larger than deg_dim. On the other hand, the QSSA cannot be justified with respect to gene expression (ge) or protein degradation (deg). It is seen in Figure 4.15 that the rates under consideration for the QSSA are not much larger than the rates of protein production and decay, and thus it cannot be assumed that they are always at steady state. This means that the CS-MA model functions as a switch even when the QSSA does not hold. These assumptions, necessary for the reduction of the model, are therefore not always justified in real systems.

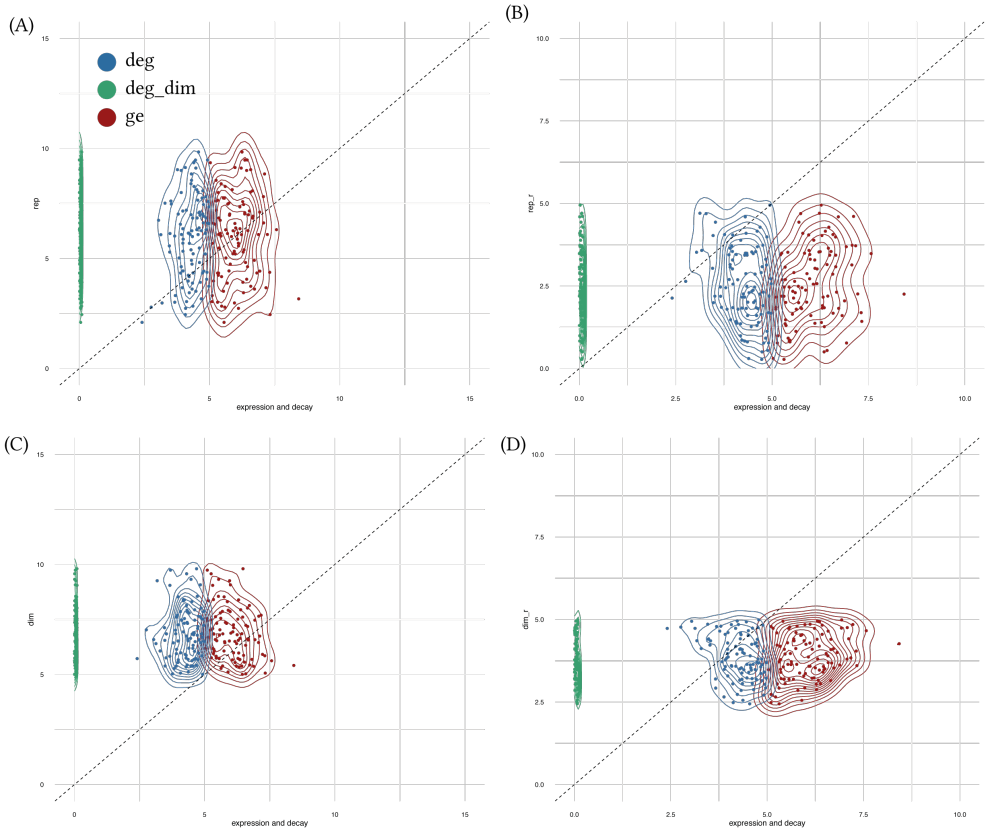

Figure 4.15 The QSSA cannot be justified for the CS-MA model. The dotted line denotes the line where $x = y$. The posterior distributions of the rates under consideration, rep, rep_r, dim, dim_r, are all much larger than the rate of degradation of the transcription factor dimer. (A) The rate of binding of the repressors to the promoter is not much larger than the rate of protein expression and degradation. (B) The rate of protein expression and decay are both larger than the rate of dissociation of the dimer to the promoter. (C) The rate of transcription factor dimerization is larger than the rates of protein production and decay, but not by a big amount. (D) The rate of monomerization of the transcription factors is smaller than the rates of protein expression and decay.

Table 4.5 Design principles of the stochastic MA bistable and tristable switches

	CS-MA		**DP-MA**	
	Bistable	*Tristable*	*Bistable*	*Tristable*
dimerisation	High	Low	High	Low
protein degradation	-	-	-	Low
dimer degradation	Low	-	Low	-

4.6.3.1 Multistability in the stochastic mass action switches

To investigate how the level of abstraction affects switch design principles, I expand the analysis under the assumption of mass action kinetics and stochastic dynamics. The asymmetric CS-MA and DP-MA models are simulated using the Gillespie algorithm (Gillespie 1977). The priors used are given in Table 4.4.

Ma et al. (2012) found that the stochastic fluctuations in a system involving such a small number of molecules, like the toggle switch, uncovers effects that can not be predicted by the fully deterministic case (Ma et al. 2012). We find that in the stochastic case, both the simple switch, CS-MA , and positive autoregulation switch, DP-MA, are capable of both bistable and tristable behaviour. The fact that tristability can occur in the classical model is consistent with the effect of small molecule numbers; if gene expression remains low, it provides the opportunity for small number effects to be observed, and the third steady state to stabilise (Ma et al. 2012). In order to ensure that the tristable switches found in the stochastic case are truly tristable, I re-sample the posterior distributions and simulate to steady state. If the resulting phase plots are tristable then we know that the posterior truly represents tristability.

As can be seen in Figure 4.16, differences in the parameter values are observed between the bistable and tristable switches, in both CS-MA and DP-MA models. We find that the simple switch is tristable when dimerisation rate is low and bistable when it is high. The degradation of the dimer proteins must have a low rate for bistability but there are no restraints in the case of the tristable switch. For the case of the DP switch, we find that the rates for dimerisation, degradation and dimer degradation are different for the bistable and tristable behaviours (Figure 4.16). The rate of dimerisation must be low for tristability to occur and large for bistability, as observed for the simple switch. The parameter for protein degradation must be low for tristability whereas there are no constraints for the bistable case. Finally, the parameter for dimer degradation must be low for bistability whereas it has no

constraints for tristability, as observed in the simple switch. The design principles for both the CS-MA model and the DP-MA model are summarised in Table 4.5

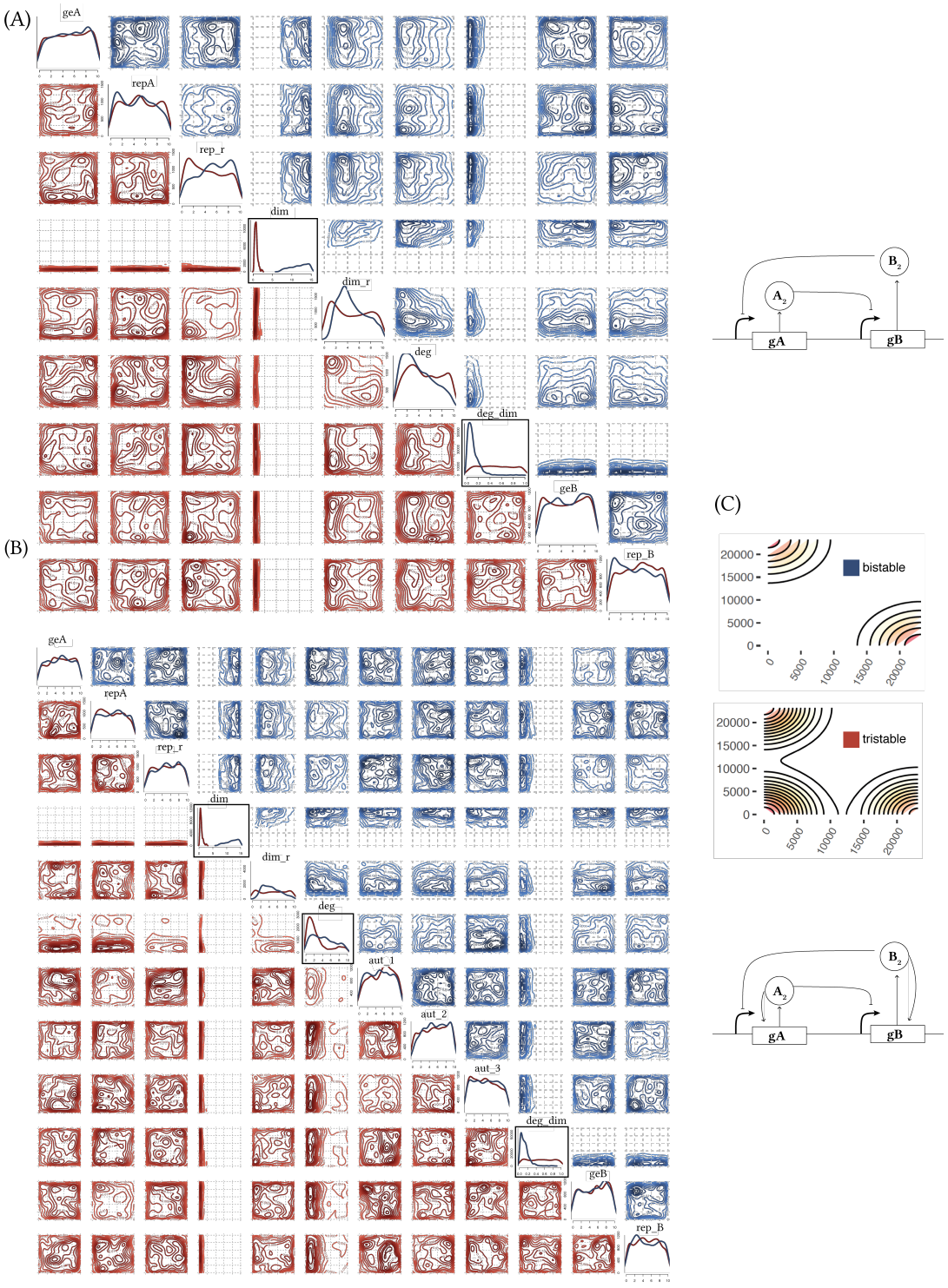

Figure 4.16 : Tristability is possible in the mass action toggle switch models when simulated stochastically. (A) The simple toggle switch with no autoregulation can be both bistable and tristable. The two posteriors are shown, where the posterior distribution of the bistable switch is shown in blue and of the tristable switch in red. From the posterior distribution we can deduce the the dimerization parameter must be small for tristability to occur but large for bistability. The switch with double positive autoregulation and its posterior distributions for the bistable and tristable case are shown in (B). (C) A sample phase plot of a stochastic tristable and bistable mass action switch.

4.6.3.2 Bayes factors depend on the choice of priors

An important aspect of Bayes factors, and thus robustness, that must be investigated is its dependence on the prior distributions. From Equation 2.32 we can expect that the measure for robustness will depend on the size of the prior. The prior distributions for model comparison are ideally chosen by using related data that can inform the choice. But this situation can be rare and priors are selected by using information from the literature in combination with rough guesses (Kass & Raftery 1995). Often simplifications are made and the choice of large prior ranges can seem like like an attractive option, as to impose less bias to the priors. Nevertheless this can have an effect on the Bayes factors calculated. The choice of improper, or very large priors can skew Equation 2.32 to favour one model over the other (Kass & Raftery 1995). In this section I demonstrate that the choice of priors for the toggle switch models has a significant effect on the calculated robustness of the models. This has to be taken under consideration when analysing candidate models for use in synthetic biology applications, as a poor choice of priors can skew the robustness analysis in favour of one model over the other, when no such robustness gain will be observed *in vivo*.

In order to demonstrate the effect of prior choice on robustness, I use the CS-MA model. I use StabilityFinder to approximate the posterior distribution that makes this model bistable using two ranges of prior distributions. The two sets of prior distributions used here are denoted as very narrow (VN2) and wide (W) in Table 4.6. The posterior distributions obtained corresponding to the VN priors and W priors are shown in Figure 4.17A and B respectively. It can be seen that the posterior distribution of the model with VN2 priors is more constrained than the model with W priors. This indicates that if one of the parameters is able to have a larger value, then the constraints on the rest of the parameters are not necessary any more. This results in the unconstrained posterior distribution observed in Figure 4.17B.

Furthermore, I test the effect of the prior range to the robustness of the model for each parameter separately. For each test, the prior distributions of all the parameters correspond to the range given in column VN2 in Table 4.6 except for the parameter being tested. For each run one parameter is being tested and has a prior range equal to the range given in column W in Table 4.6. StabilityFinder is then used to approximate the posterior distribution of the model, given the priors and bistable behaviour. Therefore, there are 7 posterior distributions of the bistable CS-MA switch, each one corresponding to one parameter having a wide prior range. For each run, the Bayes factor of the model compared to the DP-MA model used in

Table 4.6 Priors used for studying the effect of priors to robustness

	VN2	N1	W
ge	5 - 10	1 - 10	1 - 100
rep	2 - 10	1 - 10	1 - 100
rep_r	0 - 5	1 - 10	0 - 10
dim	5 - 10	1 - 10	0 - 10
dim_r	0 - 5	0 - 5	0 - 10
deg	1 - 5	0 - 10	0 - 100
deg_dim	0 - 0.1	0 - 0.5	0 - 1

Figure 4.14 is calculated, using Algorithm 5. The posterior distribution for DP-MA remains the same every time. Figure 4.17C shows the Bayes factors calculated for the DP-MA model against each run of the CS-MA. We can see from Figure 4.17C that when the priors for ge are much larger, the Bayes factor increases. The evidence for choosing DP-MA over CS-MA changes from substantial to strong, as defined in Table 2.5, by using a larger prior for parameter ge. This change is attributed to the fact the ge is constrained to be low. If the prior range is much larger, it is evident that the ratio of the volumes of the functional region to the prior will be much smaller for CS-MA.

In Section 4.6.3 we found that the DP-MA is more robust than the CS-MA model. Here I want to test whether this result still remains when the priors of the models are made wider. In order to test that, I change the prior ranges of both models and measure their robustness each time using Algorithm 5. The classes of priors used are given in Table 4.6. The robustness measures calculated, which corresponds to the fraction of the volume of the functional region over the volume of the prior, are shown in Figure 4.18A.

We find that the robustness measure changed as the priors of the models changed. When both models have very narrow or when both models have wide priors then their robustness measures are very similar. When both models have vary narrow priors the Bayes factor is equal to 1.32 and when the priors are wide the Bayes factor is equal to 1.06. In both these cases the Bayes factor is less than 3.2, and therefore is considered not significant (Kass & Raftery 1995). When the priors for both models are narrow, the Bayes factor is equal to 2.25, which is still considered not significant. Most notably, when the priors for the CS-MA model are very narrow and the priors of the DP-MA model are narrow, the Bayes' factor is at 9.14. The Bayes factor is now greater than 3.2, but less than 10, and is thus considered substantial (Kass &

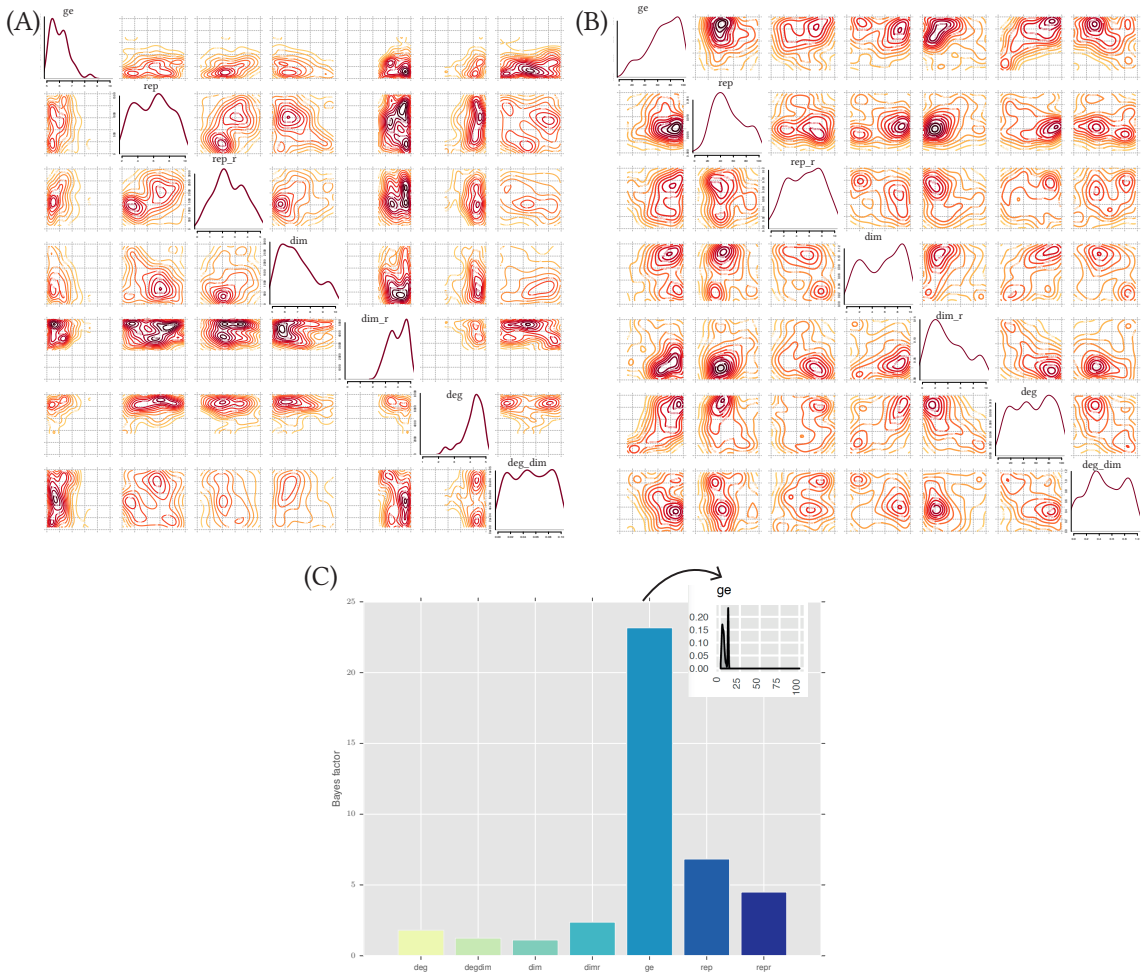

Figure 4.17 The volume of the priors has an effect on the posterior distribution obtained. (A) The CS-MA model with VN2 priors and (B) the CS-MA with W priors. (C) The increase in robustness seen is due to the gene expression parameter prior being wide while the functional region remains constrained.

Raftery 1995). These results are summarised in Table 4.7.

It is evident that the robustness measure depends on the prior volume. It is therefore useful to think of the Bayes factor in terms of the difference in the volume of the priors of the models that are being compared. I carry out this analysis for the above priors and the results are shown in Figure 4.18B. Here we see that even though the prior difference is within the same order of magnitude, the Bayes factor increases significantly. This point corresponds to the case where the priors of CS-MA are very narrow and the priors of DP-MA are narrow, and that is where we see

Table 4.7 Bayes factors of the DP-MA against the CS-MA model using different volumes of priors

Prior volume		Bayes factor ($\frac{p(B\|DP-MA}{p(B\|CS-MA)}$)
CS-MA	DP-MA	
VN2	VN2	1.32
VN2	N1	9.14
N1	N1	2.25
W	W	1.06

the Bayes factor between the two models is maximised.

These results highlight the importance of choosing the ranges of priors for the models under consideration carefully. A balance has to be struck between restricting the prior ranges too much, where interesting behaviour in the model can be missed, and making the prior range too wide, where it becomes uninformative.

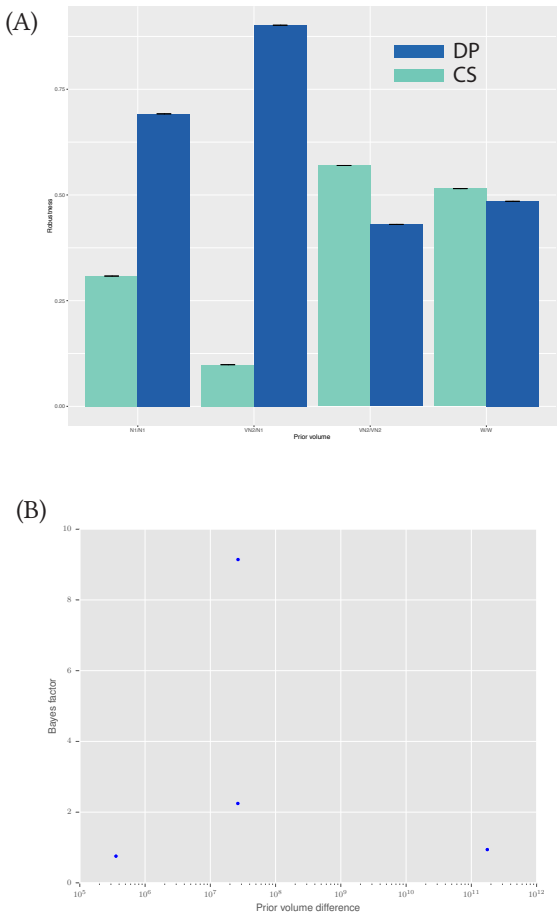

Figure 4.18 Changing the priors in both models affects the robustness measure. (A) Using different prior ranges for the CS-MA and the DP-MA models yields different robustness for each. (B) The Bayes factor as a function of the prior volume difference.

4.7 Discussion

Here I developed a novel framework, StabilityFinder, that can be used to infer parameter values that can produce a desired system multistable behaviour. The novelty in the framework I developed over existing methodology is that complex models can be analyzed assuming either deterministic or stochastic dynamics. I have used StabilityFinder to uncover the design principles of a bistable, a tristable and a quadristable switch. I found key parameters that are important in determining the number of steady states a system is capable of. This is important in the design of novel synthetic switches, where the genetic parts chosen, with their corresponding reaction rates can have an effect on the stability of the system. A bistable, a tristable or a quadristable switch could each be used for different functions within a synthetic system.

Being able to *in silico* determine the stability a given system will aid in the design of novel synthetic circuits. In the future, by selecting the system components accordingly during sequence design, the parameter values can be selected *in vivo*. For example, the parameter value corresponding to the translation initiation rate can be chosen by selecting the appropriate RBS sequence which given a nucleotide sequence will produce the desired rate (Salis, Mirsky, & Voigt 2009). Another method to tweak the parameter values *in vivo* is to select the promoter to have the strength corresponding to the levels of gene expression and repression desired. Activity of each promoter can be measured and standardised (Kelly et al. 2009) making this process possible. For a system requiring more than one promoter, these can be efficiently selected from a promoter library using a genetic algorithm created by Wu, Lee, & Chen (2011). These standardised interchangeable components with known sequence and activity constitute the ultimate goal of synthetic biology.

Nevertheless, it is important to note that the work carried out here using StabilityFinder predicts the stability of the toggle switch in isolation. Recent body of work has shown that modules like the one studied here are not independent of downstream processes (Del Vecchio, Ninfa, & Sontag 2008; Ventura et al. 2010; Jiang et al. 2011; Lyons et al. 2014). Lyons et al. (2014) showed that adding a downstream load to the genetic toggle switch can render it monostable. In order for multi-module systems to be successful, the effect of downstream loads to the system under study will have to be considered. The system including additional loads should be studied using StabilityFinder in order to determine the stability it is capable of. Therefore extrapolation of the conclusions of StabilityFinder for a given module studied in isolation would not be justified when the module is part of a larger system of modules

working in tandem.

The methodology used here can only be used to study the presence of a given stability and not its absence. If the algorithm is not converging it cannot be concluded that the given model is not capable of the desired stability under these priors. For example, the mass action switches were found to be both bistable and tristable when stochastic effects were taken into account. Using deterministic dynamics the algorithm did not converge using priors within the ranges used in this work. Nevertheless this does not permit the conclusion of absence of tristability in the deterministic classic or double positive mass action switch models. StabilityFinder only permits the interpretation of models that have converged to a given stability.

StabilityFinder can also be used to study the topology of more complex multistable switches that exist in natural biological systems such as developmental pathways. I limited this framework to the objective behaviour of a given number of stable steady states. This could be extended to examine systems with a given switching rate or systems robust to a particular set of perturbations, both of which could be of great importance for building more complex genetic circuits.

Importantly I find that the prior distributions used during such an analysis greatly affect the robustness observed. More generally, the assumptions made when building a model can have a significant effect on the predictions made. This is consistent with current understanding (Babtie, Kirk, & Stumpf 2014) and highlights the importance of the combination of experimental work and systems modelling, in order to understand the rules of thumb for abstraction in model based design of synthetic biological systems.

4.8 Summary

In this chapter I discussed the algorithm I developed and demonstrated how it can identify the parameter regions necessary for a model to achieve a given number of stable steady states. I used it to uncover the underlying principles that govern the stability of a given switch.

I first tested StabilityFinder on a known switch and then proceeded to apply it to more complex models. I uncovered the design principles that make the Lu switch bistable, tristable or quadristable. I extended the Lu models to a three-node switch and showed how it can achieve 6 steady states.

Furthermore, I built two novel models of the toggle switch which do not use the QSSA and showed that the QSSA cannot be justified in these models. Using these

models I studied the effect positive autoregulation has on the robustness of a model. I also studied the effect the priors have on the posteriors and on the robustness of a model. Finally, using stochastic modelling I showed that these switch models are capable of both bistable and tristable behaviour. In the next chapter I study the genetic toggle switch in the lab and fit the toggle switch models used here to experimental data.

5 Bayesian model fitting for flow cytometry data

5.1 Introduction

In this chapter I aim to fit the toggle switch model to experimental data. This chapter is organised as follows: In the first section I provide an overview of the framework developed to fit models to flow cytometry data (ABC-Flow). In the subsequent section I test ABC-Flow on simulated flow cytometry data. Next I use flow cytometry to study the toggle switch experimentally and examine the concentrations of the inducers and the time needed to flip the switch. Finally, I use ABC-Flow to fit a computational model to the experimental data acquired.

5.2 Contributions to this Chapter

The R code used to pre-process the flow cytometry data was provided by Alex J. Fedorec. The R code to fit the Hill function to the flow cytometry concentration assays was adapted from code provided by David T. Gonzales.

5.3 Flow cytometry

Flow cytometry detects the fluorescent intensity levels in individual cells. It can also provide physical information about the size and granularity of a cell via the forward and side scattering respectively. An overview of flow cytometry is shown in Figure 5.1. A laser excites the fluorochrome present in the bacterial cells. The fluorochromes emit a signal that is detected by channels in the optics. The signals are then all collected and analysed. A sample typically consists of single cell measurements of 10^4-10^5 cells. Flow cytometry is a powerful tool for synthetic biology

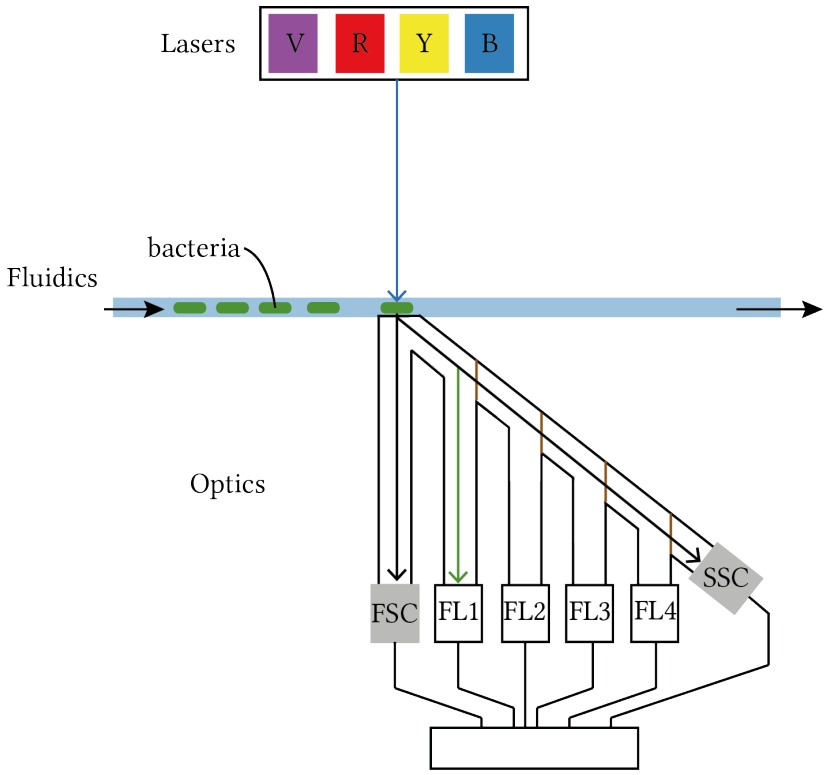

Figure 5.1 Flow cytometry. A laser excites the fluorescent proteins present in each cell. The cytometer has up to 4 lasers, violet (V), red (R), yellow (Y) and blue (B). The detectors in the optics, FL1-4 pick up the signals. The cytometer also picks up size and granularity information via the forward scatter (FSC) and side scatter (SSC) detectors. Diagram adapted from (*What is Flow Cytometry* n.d.)

as it can measure multiple parameters in single cells, and process up to 35,000 cells sec^{-1} (*Attune NxT Acoustic Focusing Cytometer* 2015).

5.4 Flow cytometry and model fitting

Computational modelling is well known to aid the understanding of complex systems by fitting experimental data and providing further insights and testable predictions. Experimental data is used to fit the model parameters and then the model can provide further understanding of the system and aid in the design of further experiments. Flow cytometry is used in synthetic biology for BioBrick characterisation (Kelly et al. 2009), enzyme screening (Choi et al. 2014) and industrial bioprocesses (Díaz et al. 2010) among others.

Flow cytometry data presents a challenge to computational modelling as the fluorescence intensity per cell is measured rather than number of proteins. The problem with measuring fluorescence intensity is that it is a relative and not an absolute measurement. This makes the inference of parameter values challenging (Rosenfeld et al. 2006). Absolute measurements would increase the predictive power of computational models (Bower, McClintock, & Fong 2010; Cooling et al. 2010), but this type of biological data cannot be directly measured (Kelwick et al. 2014). The fluorescence intensity values can vary between experiments due to instrument settings so they can only be used in relative terms within the same experiment. Efforts have been made to alleviate this problem by standardizing experimental methods (Kelly et al. 2009), or using bead-based calibration to convert the arbitrary units of fluorescent proteins to MEFLs (Molecules of equivalent fluoresceine) (Beal et al. 2016).

Another approach to the problem is converting the model output of GFP $cell^{-1}$ s^{-1} to relative fluorescence intensity. This approach was first developed by Lillacci & Khammash (2013). The converted model output can then be compared to the data output from the flow cytometer. The fluorescence intensity measurements acquired via flow cytometry are treated as a sample from the distribution of the fluorescence present in the cell (Lillacci & Khammash 2013). This means that the flow cytometry fluorescence distribution at each time point can be compared to the model fluorescence distribution. Here I expand the method developed by Lillacci & Khammash (2013) in order to be able to apply it to flow cytometry data including two fluorescent proteins simultaneously. This new framework, ABC-Flow, can be used to fit stochastic models to flow cytometry data involving multiple species like the genetic toggle switch, but could be applied to any synthetic biology system.

5.5 ABC-Flow algorithm development

The algorithm used in ABC-Flow is based on the same ABC algorithm as ABC-SysBio and Stability Finder described in Algorithms 2 and 4 respectively. ABC-Flow uses the same fundamental ABC SMC algorithm but has been adapted to be used for flow cytometry data, which required two main adaptations: Firstly, the output of the simulation module, in number of proteins, has to be converted to fluorescence intensity in order to compare it to flow cytometry data. Secondly, the distance function, measuring how close the simulated data is from the experimental data, had to be adapted in order to compare the distance between distributions of values rather than point values. This is because flow cytometry data typically involves

measurements from a large number of individual cells. The algorithm of ABC-Flow is outlined in Algorithm 6 and illustrated in Figure 5.2. The modified modules of the ABC algorithm are outlined in the sections that follow.

ABC-Flow uses stochastic dynamics to simulate the model under consideration. Gene regulation is known to exhibit stochastic dynamics (Elowitz 2002) due to the often low number of protein molecules involved. This can cause genetically identical cells to exhibit different phenotype and behaviour (Weinberger et al. 2005). Therefore, ABC-Flow uses stochastic dynamics to account for this variability observed in single cell behaviour. Just as the same genetic code can produce different phenotypes in different individual cells, the same parameter values in the model will be able to produce a different behaviour due to the added intrinsic noise to the system. The assumption is made that the cells are in identical conditions and have the same genetic code, thus extrinsic noise is not taken into consideration here.

All models are simulated stochastically using the Gillespie algorithm (Gillespie 1977). ABC-Flow simulations are implemented on GPUs. The user provides an SBML model file and an input file to specify the information needed to run ABC-Flow, such as the final epsilon threshold and the priors to the parameters. The user must also provide a data file containing the flow cytometry data to which the model will be fitted. The data files used here were generated from .fcs files, which is the standard output of flow cytometers, using the R bioconductor packages flowCore (Ellis et al. 2016b). ABC-Flow is available as a Python package, and can be downloaded from https://github.com/ucl-cssb/ABC-Flow.git.

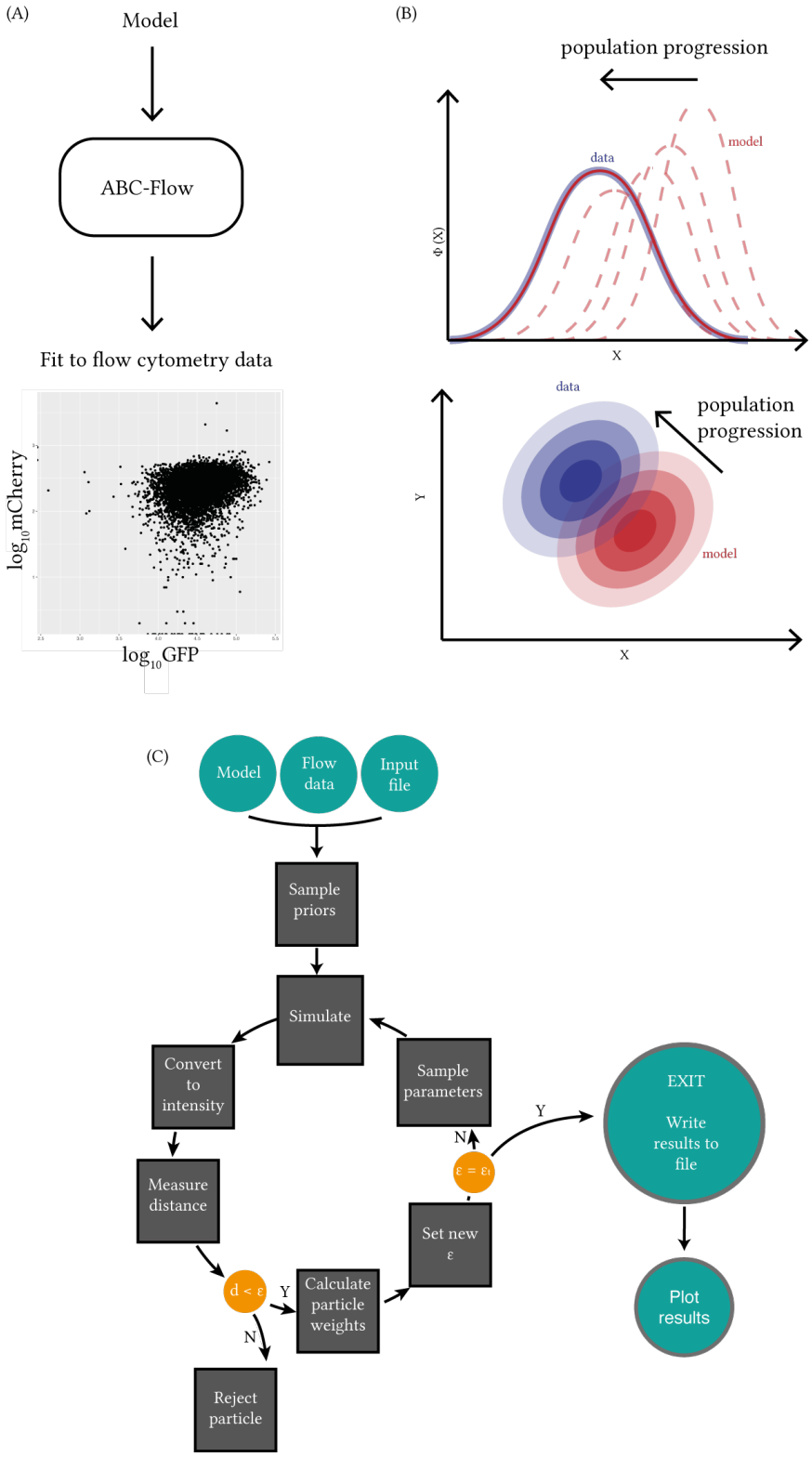

Figure 5.2 Overview of ABC-Flow. (A) ABC-Flow is used to fit models to experimental flow cytometry data. (B) The algorithm can be applied to 1D and 2D flow data. (C) ABC-Flow uses Approximate Bayesian Computation.

Algorithm 6 ABC-Flow

1: Initialise ε
2: population p ← 1
3: beta ← each stochastic trajectory
4: **if** p = 1 **then**
5: Sample particles (θ) from priors
6: **else**
7: Sample particles from previous population
8: Perturb each particle θ* using perturbation kernel K_t to obtain perturbed particle θ**
9: **end if**
10: Simulate model using the Gillespie algorithm.
11: Convert signal to intensity:
12: **for** each particle **do**
13: **for** each beta **do**
14: **for** each time point **do**
15: **for** each fluorescent protein **do**
16: Intensity ~ $\mathrm{N}\left(\text{signal}\times\mu, \sqrt{(signal \times \sigma^2)}\right)$
17: **end for**
18: **end for**
19: **end for**
20: **end for**
21: Measure distance to data
22: Reject particles if d > ε.
23: Calculate weight for each accepted θ
24: $w_t^{(i)} = \begin{cases} 1, & \text{if } p = 0 \\ \frac{\pi(\theta_t^{(i)})}{\sum_{j=1}^{N} w_{t-1}^{(j)} K_t(\theta_{t-1}^{(j)}, \theta_t^{(i)})}, & \text{if } p \geq 0. \end{cases}$
25: Normalise weights
26: Repeat steps 3 - 15 until $\varepsilon \leq \varepsilon_T$

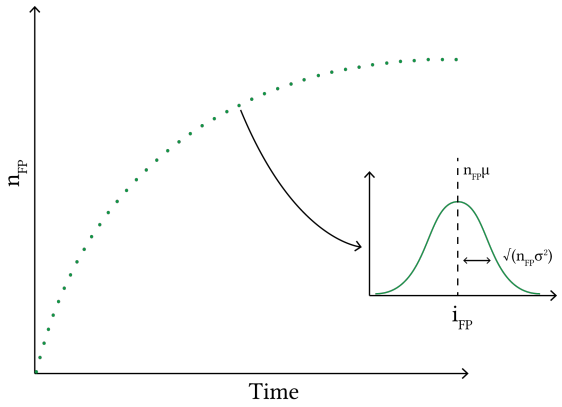

Figure 5.3 Converting the number of fluorescent proteins (n_{FP}) to the intensity (i_{FP}) is done by drawing from a normal distribution, as shown in Equation 5.1.

5.5.1 Intensity Calculation

The units of the result of the stochastic simulations is in number of molecules. On the other hand, flow cytometry data units are in the form of fluorescence intensity. For ABC-Flow, the simulation results are converted to intensity in order to be able to compare the data to the simulations. In order to do this two additional parameters are defined, intensity μ and intensity σ, for each fluorescent protein used. To convert the number of fluorescent proteins to intensity, random samples are drawn from a normal distribution

$$X \sim N\Big(n_{FP}\mu, \sqrt{(n_{FP}\sigma^2)}\Big), \tag{5.1}$$

where n_{FP} is the number of fluorescent proteins. These parameters are estimated from the data along with the model parameters. An illustration of the intensity calculation is shown in Figure 5.3. The intensity conversion also includes the addition of background signal. This represents the intensity signal detected by flow cytometers when no fluorescent proteins are present (Lillacci & Khammash 2013). The level of background fluorescence is determined by using controls during the flow cytometry experiment, and is added to the intensity signal of each fluorescent protein in the model.

5.5.2 Distance Calculation

In order to compare the flow cytometry data to the model generated data, I developed a distance measure. This distance measure should be able to determine whether two datasets are sufficiently close to each other to be able to be assume that they have been drawn from the same distribution. The measure should also give an estimate of how different the two data sets are, and thus get larger as two data sets are drawn from increasingly different distributions.

In order to select an appropriate distance metric, I tested three methods using simulated distributions. First, the calculation of the kernel distance between the simulated and experimental data was considered. As can be seen in the results shown in Section 5.5.2.1 the distance value did not monotonically decrease with increasingly similar distributions therefore it was not considered a good method for this application. I then implemented the Kolmogorov-Smirnov (KS) distance, as used in Lillacci & Khammash (2013). The KS test is a non-parametric statistic test that determines whether two data sets were drawn from the same underlying distributions by computing the largest distance between the empirical distribution functions of the two datasets. This was shown to work well when comparing simulated data to flow cytometry data (Lillacci & Khammash 2013). Nevertheless, the KS test does not scale well for multidimensional distributions. This is because there is no unique way to order the data points to calculate the largest distance. As is discussed in Section 5.5.2.2, the KS test did not perform well on two-dimensional distributions. Finally, I tested the Wald-Wolfowitz test for two-dimensional distributions. This was found to work well during the theoretical tests. Therefore, I implemented the Kolmogorov-Smirnov distance for one-dimensional datasets and the Wald-Wolfowitz for two-dimensional datasets in ABC-Flow. Each distance metric is described in more detail in the following sections.

5.5.2.1 Kernel distance

In order to measure the distance between the flow cytometry data and the fitted model, Algorithm 7 was developed. The algorithm consists of defining a grid from the minimum to the maximum value of the data. A gaussian kernel was then fit to the flow and simulated data. The distance between the two kernels is given by:

$$d = \sum_{i=xmin}^{xmax} (fD_i - fS_i)^2,$$

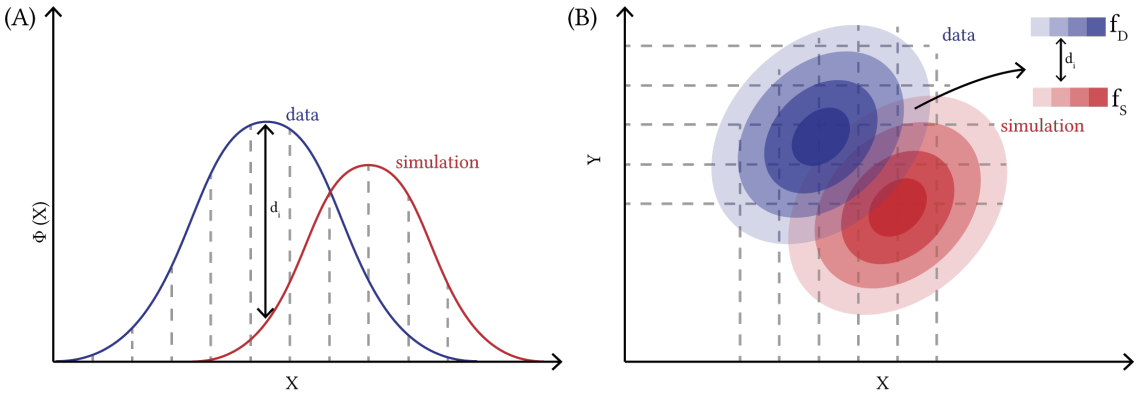

Figure 5.4 Calculating the distance between two distributions using the kernel distance in (A) 1D and (B) 2D.

Algorithm 7 1D Kernel distance calculation

1: xx ← min(data):max(data):ngrid
2: kD = kernel density estimation(data)
3: kS = kernel density estimation(simulations)
4: fD = kD(xx)
5: fS = kS(xx)
6: $d = \sum(fD - fS)^2$

where fD_i is the kernel of the flow data at each value of x and fS_i the kernel of the simulated data. An illustration of the distance calculation is shown in Figure 5.4.

In order to test this distance metric samples were drawn from two uniform distributions with varying mean and standard deviation. Algorithm 7 was then used to calculate the distance between the different distributions. First, Algorithm 7 was tested by drawing samples from two distributions with an increasingly different mean. This is done to determine the dynamical range of the distance calculation.

From Figure 5.5 we see that the distance value decreases with increasing mean difference of the two distributions. As the difference between the means increases, the distance value reaches a peak when the difference is at 3. From that point, as the mean difference increases, distance values decrease until they reach a plateau at epsilon = 0.38 in the 1D case and epsilon = 0.14 in the 2D case. Next I tested the distance calculation by comparing bimodal distributions. Two bimodal distributions are generated with increasingly different mean, in 1D and 2D.

Similar to the normal distribution, for the bimodal distributions shown in Figure 5.6 we find that the distance values do not increase linearly. There are two peaks in the distances distribution, one at mean difference = 3 and one at mean

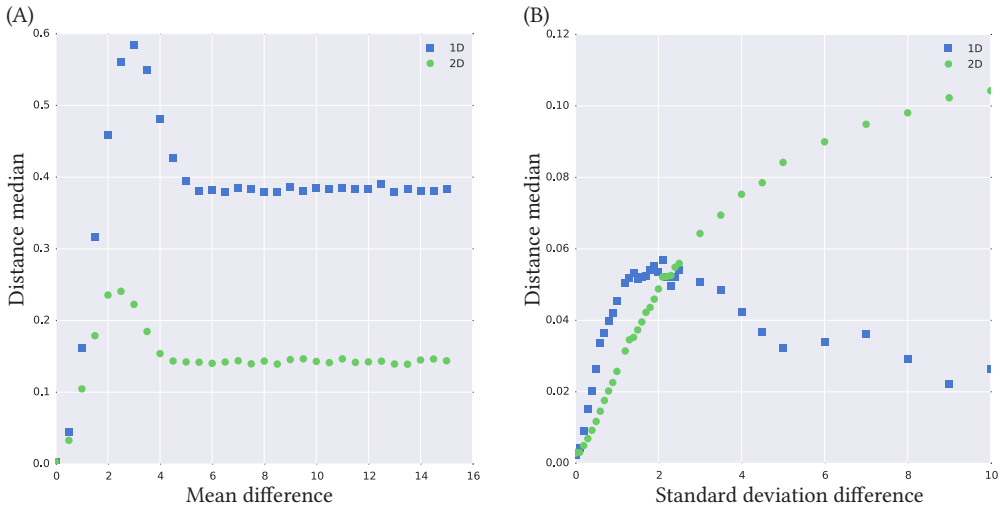

Figure 5.5 (A) The range by which distance varies as the difference between the means of the distributions increases. (B) The median of the distance distributions varies by a small amount with increasing difference in the standard deviation of the distributions.

difference = 6. The distance values then decline until they reach a plateau. The distance values do not have a large range of values, for either the 1D or 2D cases. We also find that the difference in the distance values between the 1D and 2D cases is not constant. Finally, I studied how these distance functions perform when comparing a bimodal with a normal distribution. A bimodal distribution is generated and a series of normal distributions with increasing mean, in 1D and 2D. From Figure 5.7 we find that the distance calculation is the lowest when the mean of the normal distribution corresponds to the μ of one of the two peaks in the bimodal distribution and the highest when there is no overlap between the distributions.

From Figures 5.5-5.7 I conclude that Algorithm 7 is not a good measure for distance to be used in ABC-Flow. If Algorithm 7 was used in order to minimize the distance between two distributions that start off with very different means, the distance between the two distributions will not be sufficiently minimized. This stems from the fact that ABC-Flow works by iteratively making the epsilon threshold smaller. As can be seen in Figure 5.5, if the two distributions have a large difference in the means, (>6) it would not be possible to overcome the peak that is created when the mean difference is at 3. Distance values increase before the decrease again,

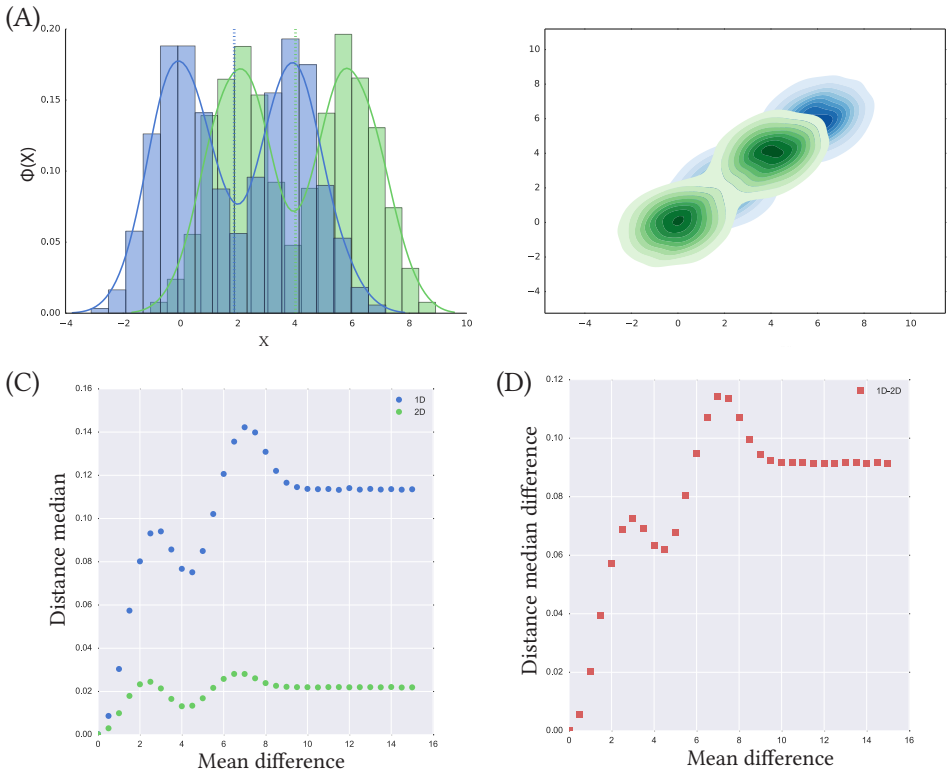

Figure 5.6 Comparing the 1D and 2D distances between bimodal distributions. (A) and (B) show samples of the bimodal distributions compared in 1D and 2D respectively with a mean difference of 4 between simulations and data. (C) The range by which the distance median varies as the difference between the mean of the distributions increases. (D) The difference between the distances calculated in 1D and 2D is not constant.

which will be a problem in ABC-Flow. Therefore a different distance calculation was developed.

5.5.2.2 Kolmogorov-Smirnov distance

In order to avoid the problems that arose from the distance calculation described in Section 5.5.2.1 I implemented a different distance calculation for ABC-Flow. I used a Python implementation of the Kolmogorov-Smirnov two sample test for the 1D case (Kolmogorov 1933). The KS distance between two distributions is equal to the largest distance between the empirical distribution functions of the two samples, as shown in Equation 5.2.

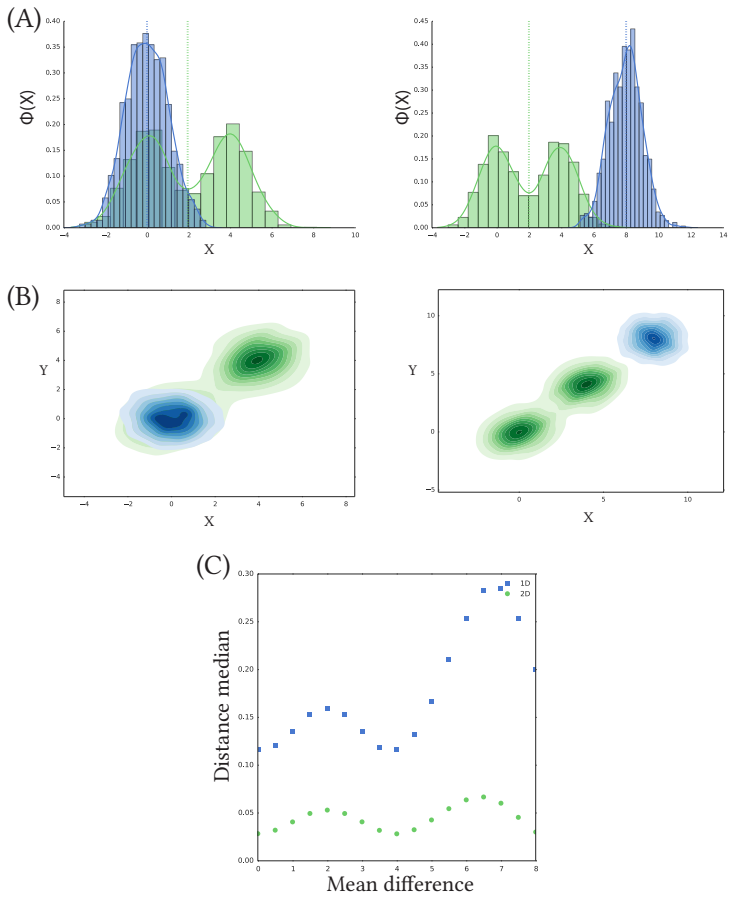

Figure 5.7 Comparing a multimodal to a normal distribution, in 1D and 2D. (A, B) The mean of the normal distribution is varied from equal to the mean of the first peak of the bimodal distribution to beyond the range of the bimodal distribution. (C) Distance median and variance are at the lowest when the mean of the normal distribution is equal to the mean of one of the peaks of the bimodal distribution.

$$D_{n,n'} = \sup_x |F_{1,n}(x) - F_{2,n'}(x)|. \tag{5.2}$$

For the 2D case the distance was calculated by using the 2D Kolmogorov-Smirnov two sample test. The algorithm was developed by Fasano & Franceschini (1987) and the Python implementation developed by Major (2016).

This distance calculation was tested to determine whether it is an appropriate distance function to use in ABC-Flow. Two datasets were drawn from normal distributions with increasingly different means. The KS test was then used to calculate the distance between the data sets. This was carried out in 1D and 2D. The results

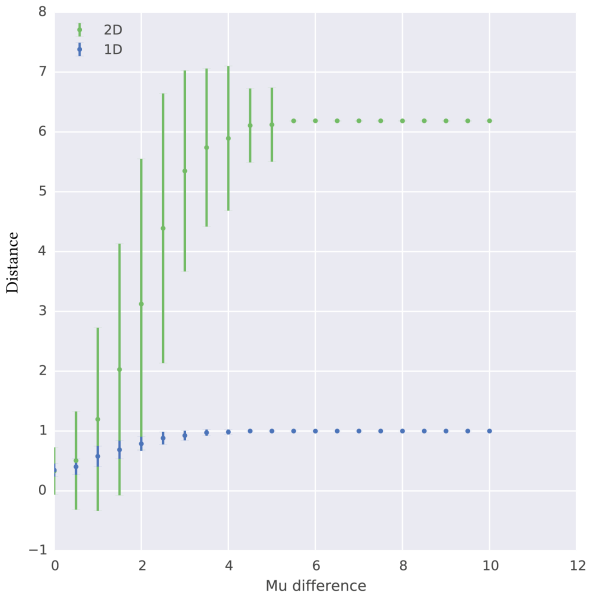

Figure 5.8 The Kolmogorov-Smirnov distance function was tested in 1D (blue) and 2D (green). Two data sets were generated with increasing mean difference, and the Kolmogorov-Smirnov two-sample test was applied to compute the distance between the two.

are shown in Figure 5.8. The distances of the 1D Kolmogorov-Smirnov calculation increased with increasing mean difference until it reached a plateau when the two distributions were very different. This distance function is therefore preferable over the kernel distance tested above. As the epsilon threshold is lowered at each iteration the difference between the two data sets decreases. Therefore the 1D KS statistic was used in ABC-Flow.

This test did not scale well in two dimensions. The variability in the calculation of the distance between data sets originating from distributions with known distance is large relative to the range of values the calculation can take. The KS is problematic in higher dimensions as there are $2^d - 1$ ways of ordering the data points and defining a cumulative distribution function, where d is the number of dimensions (Lopes, Reid, & Hobson 2007). To alleviate the above shortcomings of the multi-dimensional generalisation of the Kolmogorov-Smirnov test, a different distance calculation was used for the 2D case. The Kolmogorov-Smirnov test for the 1D case was used in ABC-Flow as it performed well in the testing shown here.

5.5.2.3 Wald-Wolfowitz distance

For the 2D case the distance was calculated by using the multivariate Wald-Wolfowitz test (Friedman & Rafsky 1979). This is a generalisation of the Wald-Wolfowitz test proposed by (Wald & Wolfowitz 1940), a non-parametric test to determine whether two data sets were drawn from the same distribution. This test works by computing the minimum spanning tree of the pooled samples. Any edge whose nodes originated from different samples are removed, and the number of *runs* (R) is then defined by the number of disjointed subtrees (Friedman & Rafsky 1979). If the number of *runs* is small, then the null hypothesis that the two samples originated from the same distribution cannot be rejected. The quantity W for two samples of length m and n is given by

$$W = \frac{R - 2\frac{mn}{N} - 1}{\sqrt{\frac{2mn(2mn-N)}{N^2(N-1)}}}, \tag{5.3}$$

where $N = m+n$ and R is the number of *runs*. A Python implementation of the multivariate Wald-Wolfowitz test by Monaco (2014) was used here. This is a variation of the Wald-Wolfowitz test that can be efficiently applied to larger data sets.

I tested this distance calculation in a similar way as Section 5.5.2.1. First, the two data sets are drawn from increasingly different distributions, and the distance between them calculated. As shown in Figure 5.9D, the distance is 0 when the two datasets are drawn from the same distribution. The distance calculation reached a plateau at distance = 140 when the mean difference was 4 or larger. The 1D distance is also shown in Figure 5.9C in order to compare the two calculations, but the 1D distance was computed using the Kolmogorov-Smirnov distance described in Section 5.5.2.2.

To further study the distance calculation used in ABC-Flow, two normal distributions were simulated, with $\mu = 0$ and $\sigma = 1$ and distance between them calculated using the Kolmogorov-Smirnov test in the 1D case and the Wald-Wolfowitz test in the 2D case. Doing this multiple times, the expected variation in distance values for identical distributions can be calculated. This represents the variation that can be expected when measuring distance in ABC-Flow. As can be seen in Figure 5.10, the range of distance values obtained in the 1D case is small. For the 2D case, the distance values obtained vary more than in the 1D case, but it is still small relative to the range of values that the Wald-Wolfowitz test can take shown in Figure 5.9.

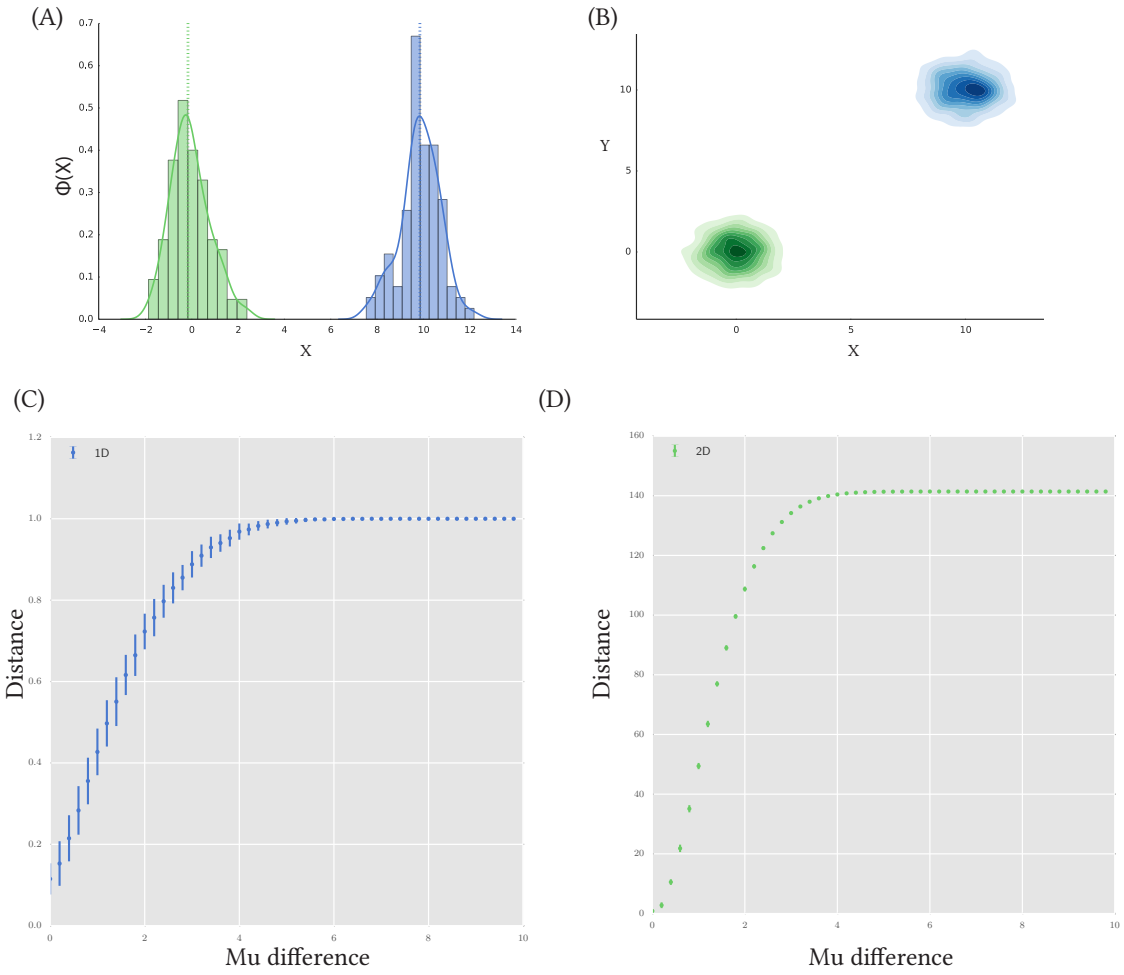

Figure 5.9 The distance calculation for data sets drawn from increasingly different distributions. Two examples are shown of distributions compared in (A) 1D and (B) 2D. (C) As the difference between the means of the two 1D distributions increases, the KS distance calculation increases until it plateaus at 1. (D) In the 2D case the distance plateaus at 140.

Using the Wald-Wolfowitz test the distance value increased with increasing distance between the distributions with relatively small variability between repeats. Since the 2D Wald-Wolfowitz test performed well in the test carried out above, it was implemented in ABC-Flow as the distance function for the 2D calculations.

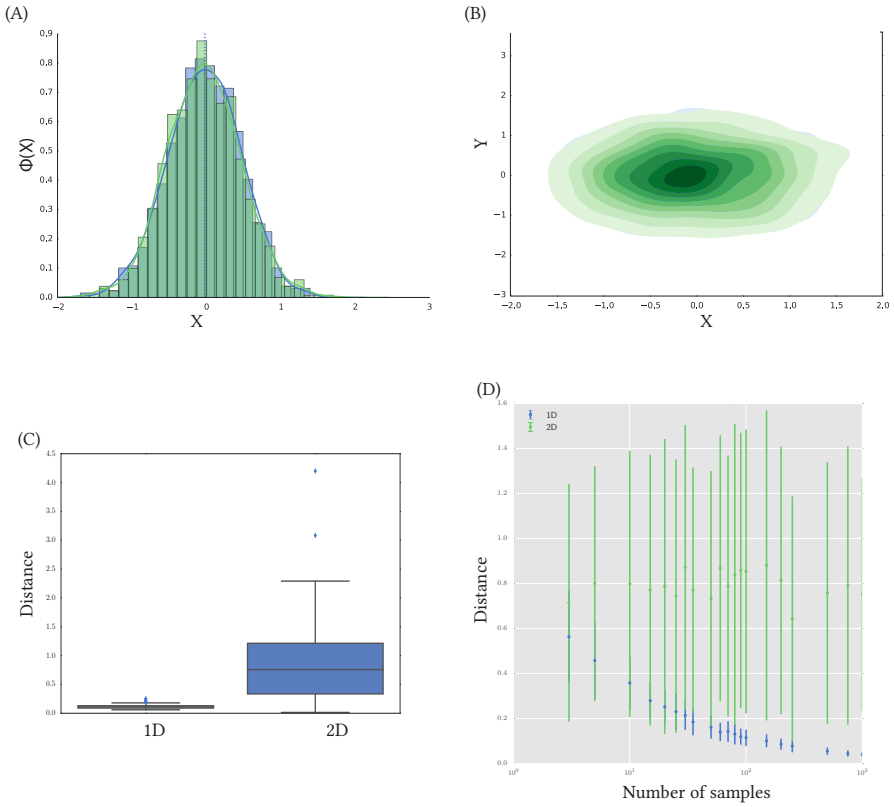

Figure 5.10 The distance between two data sets drawn from the same distribution are compared using (A) the Kolmogorov-Smirnov in 1D and (B) the Wald-Wolfowitz distance in 2D. (C) The distance is calculated for 1000 data sets. A larger variation of values is found for the 2D distance calculation, but still small relative to the overall range of values that the 2D distance can take. (D) As the number of samples in the datasets increase the distance calculation becomes more accurate in the 1D case. It has no effect on the 2D case.

5.6 ABC-Flow model fitting to simulated data

In this section I apply ABC-Flow to simulated data, where the parameter values used to produce the data are known. This analysis will serve as a verification test for ABC-Flow.

5.6.1 Toggle switch model development

The model used to produce the simulated data is an extension of the Gardner, Cantor, & Collins (2000) switch. The model consists of two mutually repressing transcription factors. The model used here has additional parameters allowing for gene expression to be leaky as well as include repression from an external stimulus. In order to produce the data set the model was simulated stochastically using the Gillespie algorithm (Gillespie 1977). The model used is defined by the following hazards:

$$h_1 = u \tag{5.4}$$

$$h_2 = \frac{a_1 l_1}{1 + l_1 + v^{\beta}} \tag{5.5}$$

$$h_3 = (1 + \alpha)v \tag{5.6}$$

$$h_4 = \frac{a_2 l_2}{1 + l_2 + u^{\gamma}}, \tag{5.7}$$

where u and v are the two proteins in the system, a_1 and a_2 represent the effective gene expression of u and v respectively, β and γ represent the cooperativity of u and v respectively and l_1 and l_2 represent the leakiness of the promoters for each species. Parameter α represents the addition of an inducer molecule that binds to repressor v, effectively increasing its degradation. This has been used previously to successfully represent the addition of the IPTG repressor to the system (Lillacci & Khammash 2013).

In addition, the system included a fluorescence intensity model. Each fluorescent molecule was assumed to emit a fluorescent signal and the signal was assumed to be normally distributed around a mean μ and standard deviation σ. The fluorescent signal emitted by each molecule was described using Equation 5.1. There was also a background fluorescence signal added to each calculation. This represents the intensity signal detected by flow cytometers when no fluorescent proteins are present (Lillacci & Khammash 2013). The background fluorescence can be measured using control samples (Lillacci & Khammash 2013), but in the case of the simulated data it was set to 0.01 (a.u.).

Table 5.1 The priors used for the 1D and 2D ABC-Flow model fitting to simulated data

Parameters			
	Units	1D	2D
a_1	molecules h^{-1}	80 - 120	80 - 120
β	molecules^{-1} h^{-1}	1 - 4	1 - 4
l_1	h^{-1}	750 - 850	750 - 850
a_2	molecules h^{-1}	80 - 120	80 - 120
γ	molecules^{-1} h^{-1}	1 - 5	1 - 4
l_2	h^{-1}	750 - 850	750 - 850
α	h^{-1} μM^{-1}	1 - 4	1 - 4
Species			
u		9 - 11	9 - 11
v		90 - 110	90 - 110
Intensity parameters			
μ fp_u	AU	0 - 2	0 - 2
μ fp_v	AU		0 - 2
σ fp_u	AU	4 - 6	4 - 6
σ fp_v	AU		4 - 6

5.6.2 Parameter inference for simulated flow cytometry data

The model given in Equations 5.4-5.7 was simulated in order to obtain the time course for the two fluoscent molecules in the system, u and v. The time course data, the model and the prior distributions shown in Table 5.1 were supplied to ABC-Flow. ABC-Flow was then used to infer the parameters that gave rise to the data using one (1D) or both (2D) fluorescent molecule time courses to fit to the data. We can therefore determine whether there is an added benefit on fitting both fluorescent proteins to the data. In order to compare the 1D and 2D fits, equivalent epsilon distance values were used. These were determined by comparing the distance value obtained when measuring the distance between two 1D distributions with the distance value obtained when measuring the distance between two 2D distributions using the appropriate distance measure for each one. This was done using the theoretical distance values shown in Figure 5.9 for the 1D and 2D case, for distributions where the difference in the means is 0.2 or less. Therefore they were set to 0.27 for the 1D fit and 2.82 for the 2D fit.

The fit resulting from the 1D fit is shown in Figure 5.11A, together with Q-Q

Table 5.2 The parameters inferred from simulated data

Parameters	True value	Posterior 95% credible region					
		1D			2D		
		0.025	Median	0.975	0.025	Median	0.975
a_1	100	83.507	96.586	114.485	80.064	93.539	104.940
β	2	1.873	2.003	2.167	1.926	2.063	2.131
l_1	800	755.923	807.885	836.913	761.972	791.912	840.872
a_2	100	80.162	98.570	108.148	83.362	93.726	109.974
γ	2	1.195	1.642	2.163	1.679	1.93	2.233
l_2	800	752.757	801.021	838.058	758.759	803.549	845.507
α	1	0.495	1.055	1.507	0.607	0.932	1.359
$\mu\ fp_u$	1.5	1.126	1.749	2.542	1.762	2.164	2.546
$\mu\ fp_v$	1.5				0.826	1.067	1.413
$\sigma\ fp_u$	4	3.086	3.786	5.712	3.028	4.053	4.121
$\sigma\ fp_v$	4				4.005	4.211	4.396

plots to assess the fit to the data. A Q-Qplot is a plot where the quantiles of two distributions are plotted against each other. If the two datasets were drawn from the identical distributions then the points will lie on the x = y line (Wilk & Gnanadesikan 1968). The fit resulting from the 2D fit is shown in Figure 5.12, while the posterior distributions obtained from both fits are shown in Figure 5.13 and Table 5.2. Comparing the 1D and 2D fits, we find that both fits inferred the parameters necessary to produce the simulated data. From the posteriors we find that the most identifiable parameters in this model are parameters β and γ, the parameters representing the cooperativity of the repressors. We find that the 2D fit resulted in a better inference of the parameter values. This is most prominent in parameters β, γ and α as can be seen in Table 5.2. Parameters l_1 and l_2, the parameters representing the leakiness of the promoters, remained unidentifiable for both fits. For a system involving two fluorescent proteins, like the toggle switch, it would therefore be beneficial to use a two-dimensional fit to the data. The 2D fit performed better in the simulated test used here, with priors centred closely around known parameter values and clean data. This will become more evident when ABC-Flow is used on real experimental data where less information is known for the priors. Therefore, a system involving two fluorescent proteins should be fit using the 2D fit in ABC-Flow. The 1D fit should be used in cases where there is only one fluorescent protein of interest in the system.

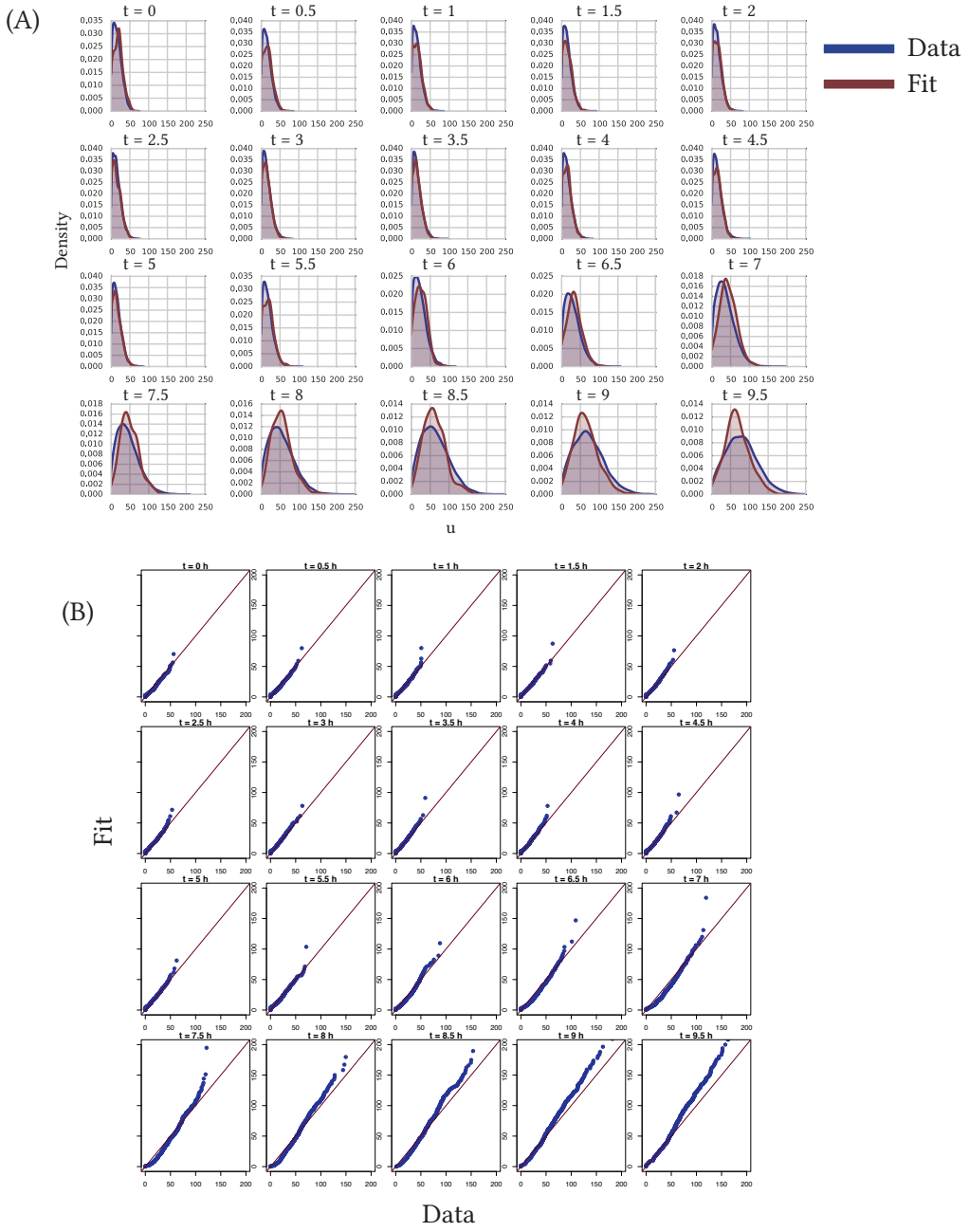

Figure 5.11 (A) Time course produced by the 1D ABC-Flow fit (shown in blue) to data (shown in red) produced by simulating the same model. (B) Q-Qplot of each time point fit. The quantile of the two distributions are plotted against each other. If the distributions are identical, the points would lie on the x = y line, shown in red.

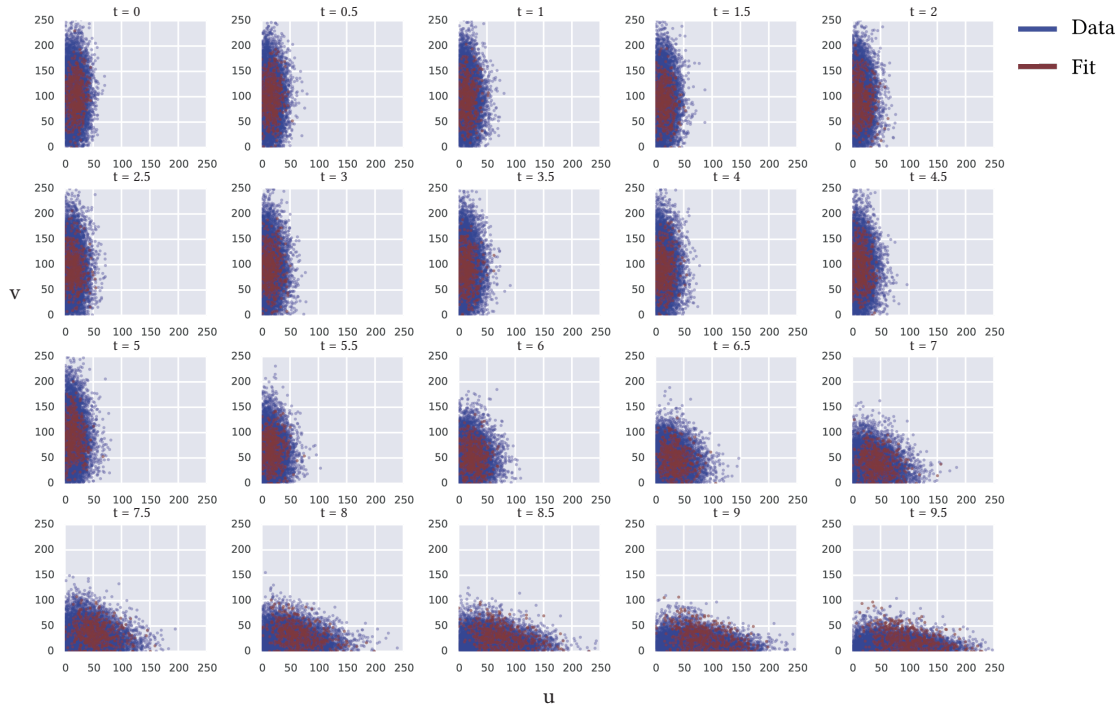

Figure 5.12 2D ABC-Flow fit (shown in blue) to data (shown in red) produced by simulating the same model.

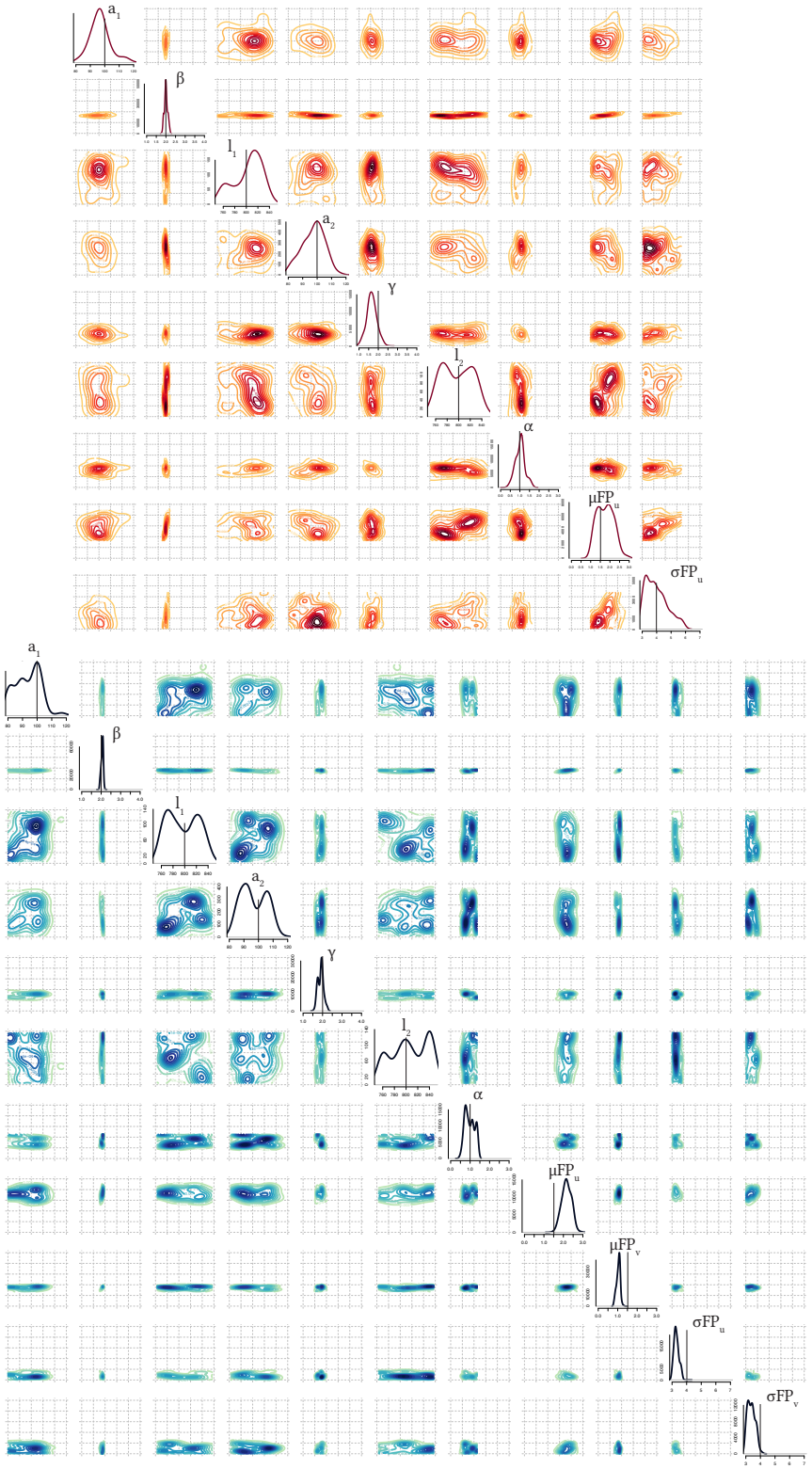

Figure 5.13 Posterior distributions of inferred parameters from (A) 1D and (B) 2D simulated data. The parameters used to produce the simulated data set were well inferred in both cases. The parameter values used to produce the data are marked on the marginal posterior distributions as a black line.

These results demonstrate that ABC-Flow can successfully fit a computational model to flow cytometry data. It can identify the parameter values necessary to produce the observed behaviour. ABC-Flow can now be confidently applied to real flow cytometry data of the genetic toggle switch. This allows the inference of the parameter values of the toggle switch model producing the data observed.

5.7 Toggle switch data collection

In this section I describe the collection of experimental data on the genetic toggle switch. Using flow cytometry and the necessary inducers to flip the switch I study the dynamics over time as well as over different inducer concentrations.

5.7.1 Circuit overview

The toggle switch plasmid I used here was provided by Litcofsky et al. (2012). All the switch components were contained in one plasmid, pKDL071. An overview of the plasmid is shown in Figure 5.14A. The circuit consists of two promoters, P_{trc2} and $P_{LtetO-1}$ (Lutz & Bujard 1997). P_{trc2} is a LacI repressible promoter. $P_{LtetO-1}$ is also a TetR repressible promoter, as shown in Figure 5.14B. mCherry (Shaner et al. 2004) and GFP (Shimomura, Johnson, & Saiga 1962) are fluorescent proteins, under the control of the same promoters as the repressors, and thus reflect the levels of TetR and LacI in the system. The plasmid contains kanamycin antibiotic resistance and is high copy (ColE1 origin of replication).

This system is capable of two states, GFP high/mCherry low and GFP low/mCherry high. When IPTG is added to the system, it represses the repression of TetR and mCherry and thus the cells transition to the mCherry high state. When aTc is added to the system, it represses the repression of LacI and GFP and thus the cells transition to the GFP high state. If no inducer is added to the system it will randomly go to the GFP high or mCherry high states.

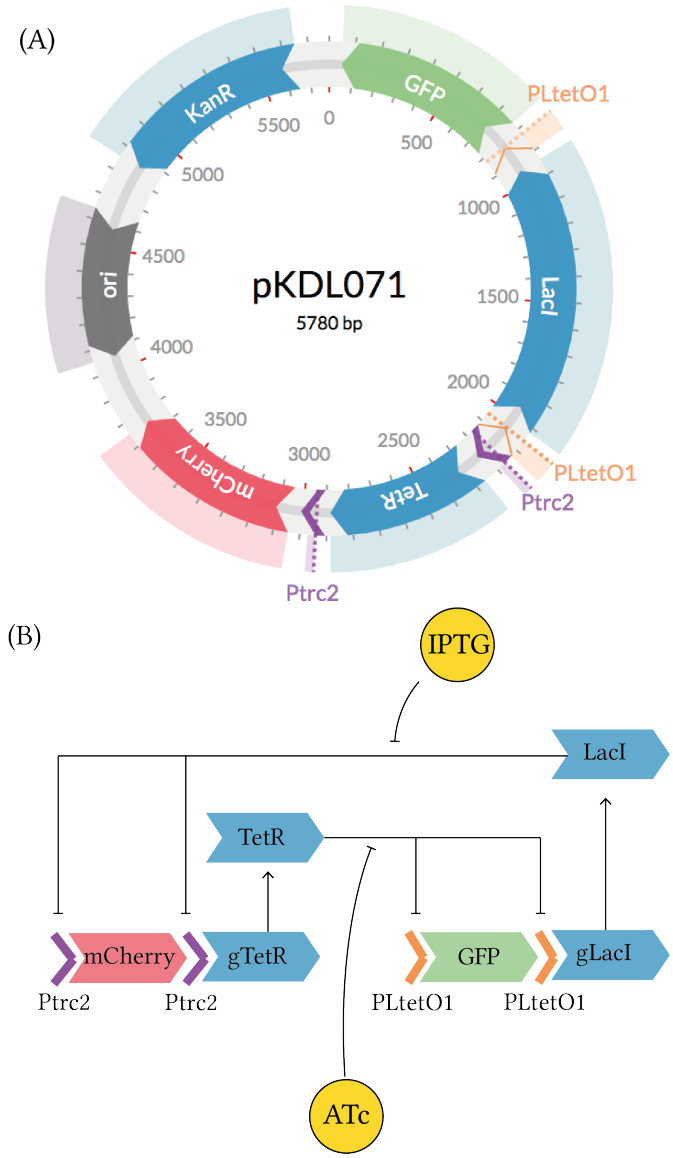

Figure 5.14 The genetic toggle switch circuit used in this chapter. (A) The plasmid map of pKDL071, the plasmid containing the genetic toggle switch used in Litcofsky et al. (2012) (B) The interactions between each element of the circuit.

5.7.2 Methods

The toggle switch plasmid was provided by the James J Collins lab in the form of a stab culture in *E. coli* K-12 MG1655.

5.7.2.1 *Escherichia coli* culturing conditions

Lysogeny broth (LB) was made by diluting LB in deionized water to a concentration of $25\,\mathrm{g\,L^{-1}}$ and subsequently autoclaved at 121 °C for 15 minutes. LB agar plates were made by adding bacteriological agar to the above solution to a concentration of $45\,\mathrm{mg\,mL^{-1}}$ before autoclaving. The solution was then cooled down to 55 °C using a water bath. If antibiotic was required it was added to the correct concentration to the cooled solution. The solution was then aliquoted to petri dishes and left to solidify at room temperature. The plates were stored in the fridge for up to 1 month.

Overnight cultures were made by picking a single colony from a static culture on an agar plate. Each colony was placed in 15 mL Falcon tubes (Fisher Scientific, MA, U.S.A) with 5 mL LB with kanamycin antibiotic at a concentration of $50\,\mathrm{\mu g\,mL^{-1}}$. The tubes were then screwed loosely and taped securely in order to allow for aeration. The falcon tubes were put in an incubator at 37 °C with orbital shaking at 200 rpm for 12-16 hours.

5.7.2.2 Glycerol stock preparation

To preserve the transformed cultures long-term glycerol stocks were made. 5 mL LB and Kanamycin overnight cultures were made as described in Section 5.7.2.1. The cultures were kept on ice and 70 % glycerol was added to the cultures in a ratio of glycerol to culture of 1:7. These were aliquoted into cryovials and transferred to a −80 °C freezer for long-term storage.

5.7.2.3 Revival

For subsequent revival of the frozen cultures, a 1.5 mL eppendorf tube was removed from the −80 °C freezer and put on ice. A small amount was streaked onto an agar plate containing LB and kanamycin. The plates were stored in an incubator at 37 °C overnight. Then the plates were sealed using parafilm and stored at 4 °C for up to two weeks.

5.7.2.4 Plasmid construction

Plasmids were constructed via PCR cloning. PCR primers were chosen to add restriction enzyme sites on the 5' and 3' where needed. Following PCR amplification, the amplified DNA was purified using the Qiagen PCR cleanup kit (Qiagen, Crawley, U.K). Double digests were carried out and the desired fragment isolated via gel extraction. The relevant fragments were subsequently ligated. Following construction, each plasmid was isolated using the QIAprep Spin Miniprep Kit (Qiagen, Crawley, U.K). Plasmid concentration was determined using the Thermo Scientific NanoDrop 1000 Spectrophotometer (Fisher Scientific, MA, U.S.A).

5.7.2.5 Polymerase Chain Reaction

In order to amplify DNA and add the restriction enzyme sites required, a Polymerase Chain Reaction (PCR) reaction was carried out with mutagenic primers. A list of primers can be found in Appendix B. Q5® DNA Polymerase (NEB, MA, U.S.A) was used with its associated buffer, dNTPs and Q5® enhancer, as specified in Table 5.3. PCR reactions were run in a T100™ thermal cycler (Bio-Rad Laboratories, Inc., UK) as per the Q5® recommendations, and as outlined in Tables 5.3 and 5.4.

Table 5.3 PCR recipe

Reagent	Final concentration	50 µL reaction
Q5® buffer 5X	1X	10 µL
dNTPs	200 mM each	1 µL
Forward primer	0.5 µM	2.5 µL
Reverse primer	0.5 µM	2.5 µL
Template DNA	2 µg/50 µL	-
Q5® DNA polymerase	0.02 U μL^{-1}	0.5 µL
Q5® enhancer	1X	10 mL
H_2O	-	to 50 µL

Table 5.4 Thermocycling conditions

Step	Cycles	Temperature	Time
Initiation	1	98 °C	30 s
Denaturation		98 °C	10 s
Annealing	30	55 °C -72 °C	20 s
Extension		72 °C	2 min
Final extension	1	72 °C	30 s/kb
Hold	1	4 °C	∞

5.7.2.6 Digestion

All enzymes, buffers and Bovine Serum Albumin (BSA) were supplied by NEB. Digestion controls were carried out by adding H_2O instead of DNA in the digestion reaction. Additionally, during agarose gel electrophoresis uncut plasmid was run alongside the digested plasmid as a further control.

1 µg digests were set up by mixing the plasmid with 0.5 µL of each restriction enzyme as per the recipe shown in Table 5.5. The reactions were placed in an incubator at 37 °C for 4 hours. Finally, the solutions were analysed using agarose gel electrophoresis (Section 5.7.2.7).

Table 5.5 Digestion recipe

Reagent	Volume
PstI	0.5 µL
HindIII	0.5 µL
NEB Buffer 2.1	2 µL
BSA	0.2 µL
DNA	1 µg
H_2O	to 20 µL

5.7.2.7 Agarose gel electrophoresis

To make a 0.8% agarose gel, 0.4 g agarose were diluted in 50 mL 1X TAE buffer. It was further dissolved by microwaving for 1-3 minutes. The solution was left to cool for 5 minutes and then 1.5 µL GelRed™ (Biotium, Fremont, CA) was added. Gel trays were prepared by putting the well comb in place and taping the ends shut. The solution wast then poured into the prepared gel trays and left to solidify for 20-30 minutes at room temperature.

Agarose gel electrophoresis was carried out by placing the poured gels into the gel tanks. The tank was then flooded with 1X TAE buffer. The DNA was prepared to be analysed by adding 4 µL loading dye to 20 µL sample. A negative control was used with H_2O instead of sample. The DNA ladder of choice was prepared by adding 1 µL H_2O and 1 µL dye to 2 µL ladder. The agarose gel was ran at 90 V between 45 - 60 minutes.

To purify the fragments from the agarose gel, a sterile razor blade was used to cut out the desired fragment. This was placed in a clean eppendorf tube. The DNA was isolated from the gel using the QIAquick Gel Extraction Kit.

5.7.2.8 Ligation

A ratio of 3:1 of insert to recipient plasmid was used, 1 µL T4® DNA ligase (NEB, MA, U.S.A) and 2 µL ligase buffer. Molecular biology grade H_2O was added to make the reaction up to 20 µL. The controls used for each ligation reaction, are shown in Table 5.6. Control 1 is used to detect competent cell viability, control 2 background due to uncut vector, control 3 re-circularization and control 4 contamination.

The ligation reactions were placed at 4 °C for 12 hours. The reactions were then placed at 65 °C for 10 minutes to heat inactivate the T4 DNA ligase enzyme. A transformation was then carried out as per Section 5.7.2.9.

Table 5.6 Ligation controls

	Control 1	Control 2	Control 3	Control 4
Vector	Uncut	✓	✓	✗
Insert	✗	✗	✗	✓
Buffer	✓	✓	✓	✓
H_2O	✓	✓	✓	✓
Ligase	✗	✗	✓	✓

5.7.2.9 Transformation

Thermocompetent *E.coli* DH5α was transformed with the constructed plasmids. Each ligation reaction was added to 50 µL of thawed competent cells. The cells were subsequently kept on ice for 30 minutes, then placed at a 42 °C water bath for 45 s. The cells were then placed back on ice for 15 minutes. Then 500 µL of Super Optimal broth with Catabolite repression (SOC) was added to each ligation and placed in a 37 °C shaking incubator for 3 hours. 500 µL and 50 µL were subsequently pipet-

ted of each ligation onto petri dishes with LB agar and the appropriate antibiotic. The plates were incubated at 37 °C for 12-16 hours. Two controls were used for the transfection protocol, a positive control with no antibiotic in the LB agar and non-transfected cells and a negative control of non-transformed cells and LB agar with antibiotic. These ensure that the cells are viable and not contaminated respectively.

Individual colonies were then selected from each transfection and grew each separately in 5 mL LB medium for 12-16 hours at 37 °C, 200 rpm. Glycerol stocks were then prepared from each culture, as per Section 5.7.2.2.

5.7.2.10 Colony PCR

In order to determine if the fragment was successfully inserted into the vector DNA plasmid, diagnostic colony PCR was then carried out. Primers were designed that amplified the multiple cloning site of the vector DNA plasmid. These can be found in Appendix B. A PCR master mix was made for the number of colonies to be amplified, 32, with an added 10% to account for pipetting error. GoTaq® Flexi DNA polymerase (Promega Corp., WI, U.S.A.) was used with its associated buffer, dNTPs and $MgCl_2$and H_2O. The recipe for the master mix is shown in Table 5.7.

Table 5.7 Colony PCR master mix recipe

Reagent	Final concentration	Master mix
GoTaq® green Flexi buffer	1X	141 µL
dNTPs	200 mM each	14.1 µL
Forward primer	0.5 µM	1.4 µL
Reverse primer	0.5 µM	1.4 µL
GoTaq® Flexi polymerase	0.02 U μL^{-1}	3.5 µL
$MgCl_2$	1X	42.2 µL
H_2O	-	465 µL

19 µL were then added from the master mix to each PCR tube. Each of the colonies was then lifted from the transformation from the agar plate using a 20 µL pipette tip and added it to a PCR mix by mixing. The pipette tip was subsequently used to make a scratch into a clean agar plate, and labelled. A PCR was then carried out according to GoTaq® Flexi polymerase recommendations, and as shown in Table 5.8.

Table 5.8 Thermocycling conditions for colony PCR

Step	Cycles	Temperature	Time
Cell lysis	1	95 °C	10 minutes
Denaturation		95 °C	30 s
Annealing	35	55 °C -72 °C	1 minute
Extension		72 °C	30 s/kb
Final extension	1	72 °C	5 min
Hold	1	4 °C	∞

Finally a diagnostic agarose gel electrophoresis was carried out as outlined in Section 5.7.2.7.

5.7.2.11 Sequencing

In order to confirm plasmid identity, all plasmids were sequenced using Source Bioscience, Cambridge UK. 10 µL of each plasmid DNA were submitted at a minimum of 100 ng µL^{-1} as per the requirements. Primer sequences were also submitted and manufactured by Source Bioscience. Primers can be found in Appendix B.

5.7.2.12 Inducers

Anhydrotetracycline (aTc) solution was made by diluting aTc from Cayman Chemical Company in 100 % ethanol to a concentration of 1 mg mL^{-1}. Isopropyl-beta-D-thiogalactopyranoside (IPTG) solution was made by dissolving IPTG in deionized water to a concentration of 1 M. The solution was sterilised by passing the solution through a 0.22 µm syringe filter. Both inducers were stored in 1 mL aliquots at −20 °C.

5.7.2.13 Growth rate measurement

Plate reader analysis was carried out in order to measure the growth of *E.coli* over time. Overnight cultures were made using the method shown in Section 5.7.2.1. Overnight cultures were then diluted by a 1:1000 ratio into a 5 mL LB + kanamycin solution. The diluted cultures were grown at 37 °C with shaking at 200rpm for 1 hour. These cultures were then further diluted by a 1:100 ratio. 200 µl aliquots of the dilutions were then transferred to a clear flat bottom, black-walled 96-well plate. Wells with only LB and kanamycin were also added in order to be used as blanks. The plate was then sealed using a gas permeable membrane and placed in a BMG

FLUOstat OPTIMA plate reader to measure absorbance. The plate reader was set to a constant 37 °C, with 30 seconds orbital shaking at 150 rpm and 4 mm shaking width every ten minutes. Absorbance was measured at 540 nm. Data was exported as a CSV file and analysed using Python.

5.7.2.14 Flow cytometry

Flow cytometry experiments were carried out in order to get fluorescence levels in single cells. Flow cytometry data was exported as FCS files and analysed using the R bioconductor packages flowCore (Ellis et al. 2016b), flowViz (Ellis et al. 2016a) and Ggplot2 (Wickham 2009). Prior to analysis the raw data was processed to remove any debris or instrument noise detected. The data was also processed to removed any doublets, which occurs when more than one bacterial cell passes through the detector at a time. This will skew the data by including datapoints with double the fluorescent intensity that the rest of the population. The pre-processing was done by using the side scattering data. The height and the area of the sample forward scattering distribution is recorded during an experiment. The cells that lie in the diagonal where the area equals the height are single bacterial cells. If the area of the signal exceeds the height it is indicative of a doublet, or cluster of cells, and is removed from the data. This pre-processing was carried out using autoGate, developed by Fedorec (2016).

5.7.2.15 Concentration assays

Concentration assays were carried out in order to determine the concentration of each inducer (aTc and IPTG) at which the switch changes state. Separate overnight cultures were prepared as per Section 5.7.2.1 with added IPTG at a concentration of 1 mM or added aTc at a concentration of 100 ng mL^{-1} (Litcofsky et al. 2012). The cultures were then diluted by 1:1000 into fresh LB medium with varying concentrations of the opposite inducer than what the cells were grown in overnight. The concentrations used are shown in Table 5.9. For each concentration, triplicate cultures were made.

The cultures were placed in an incubator at 37 °C, 200rpm for 5 hours. The cultures were then placed in a centrifuge and spun at 13,000rpm for 5 minutes. The supernatant was discarded and replaced it with 1 mL PBS solution. The BD LSRFortessa^{TM} cell analyzer (Becton, Dickinson and Company) was used at the St. Mary's Flow Cytometry Core Facility at Imperial College London for flow cytometry analysis. GFP was excited using the 488 nm laser and detected using the

Table 5.9 Concentrations used for flow cytometry assay

aTc (ng/ml)	IPTG (M)
0.05	1e-7
0.06	6e-7
0.07	1e-6
0.08	6e-6
0.09	1e-5
0.1	1e-3
1.0	0.1

533/30 filter. mCherry was excited using the 561 nm laser and detected using the 620/10 filter. Data was obtained at n=10000 events per experiment.

5.7.2.16 Time course assays

Time course assays were carried out to measure the time it takes for the switch to flip to each state. Separate overnight cultures of pKDL071 were prepared as per Section 5.7.2.1 with added IPTG at a concentration of 1 mM or added aTc at a concentration of 100 ng mL^{-1} (Litcofsky et al. 2012). Overnight cultures of pSEVA281G and pSEVA281C were also made. The cultures were then diluted by a ratio of 1:1000 into fresh LB medium. Separate cultures for each time point were made, in triplicate. For cultures grown overnight in IPTG, aTc was added at a concentration of 100 ng mL^{-1} and for cultures grown overnight in aTc, IPTG was added at a concentration of 1 mM. All cultures were placed at 37 °C, 200rpm incubator. At 30 minutes, 1 hour and then every hour up to 6 hours flow cytometry was carried out for the corresponding cultures. Triplicates for each induction were removed from the incubator and placed in a centrifuge at 13, 000rpm for 10 minutes. The supernatant was discarded and replaced with 1 mL PBS solution. These cultures were then analysed in an Attune^{TM} NxT Flow Cytometer (Thermo Fisher Scientific) at University College London. GFP was excited using the 488 nm laser and detected using the 533/30 filter. mCherry was excited using the 561 nm laser and detected using the 620/10 filter. Data was obtained at n=10000 events per experiment. pSEVA281G and pSEVA281C cultures were used to set the laser voltages and pKDL071 cultures to detect the bacteria population.

5.7.3 Results

5.7.3.1 pKDL071 plasmid alteration

The pKDL071 plasmid contains all the elements of the switch. The two states of the switch are LacI high and TetR high. These are detected by using the fluorescent proteins that are controlled by the same promoters, and thus mirror the levels of LacI and TetR. The concentration of LacI can be estimated by GFP intensity and TetR concentration by mCherry intensity. In order to detect GFP and mCherry levels within each cell simultaneously, flow cytometry can be used. The lasers needed to excite GFP and mCherry are 488 nm blue and 561 nm yellow respectively. Since the yellow laser was not available for use in the BD Accuri^{TM} C6 or the BD LSRII^{TM} (Becton, Dickinson and Company) flow cytometers available, an alternative construct had to be made in order to be able to detect the levels of both sides of the switch.

In order to alter the switch construct to be able to detect both sides, the mCherry gene was swapped for the YFP gene. The yellow fluorescent protein is excited by the blue laser and could thus be detected using the equipment available. The YFP gene was available from BioBrick registry of standard biological parts as BBa_K592101. PCR cloning was used to introduce the flanking sequences of EcoRV and KasI restriction enzymes in the 5' and 3' ends respectively. The primers used are given in Appendix B . A double digest was performed on plasmids pKDL071 and BBa_K592101, as well as positive and negative controls. Following gel extraction and ligation, the pKDL071-YFP plasmid was complete. The plasmid map is shown in Figure 5.15.

GFP and YFP have overlapping emission spectra, which have to be compensated during flow cytometry data acquisition (Shapiro 1941). This is because the signal from GFP can be detected at the YFP detector and vice versa. Due to the high level of compensation needed to be carried out and the relatively dim signal given by the bacteria used here, the different stages of the switch, ON and OFF, could not be resolved (data not shown). In order to be able to acquire toggle switch flow cytometry data, an alternative facility was found that was able to detect GFP and mCherry fluorescence.

5.7.3.2 Control plasmids construction

I constructed two plasmids in order to use them for the flow cytometry mCherry-/GFP experiments. The first plasmid, pSEVA281G contains the promoter $P_{LtetO-1}$ and GFP and the other, pSEVA281C contains the promoter P_{trc2} and mCherry from PKDL071, shown in Figure 5.16. These two plasmids were used to determine the

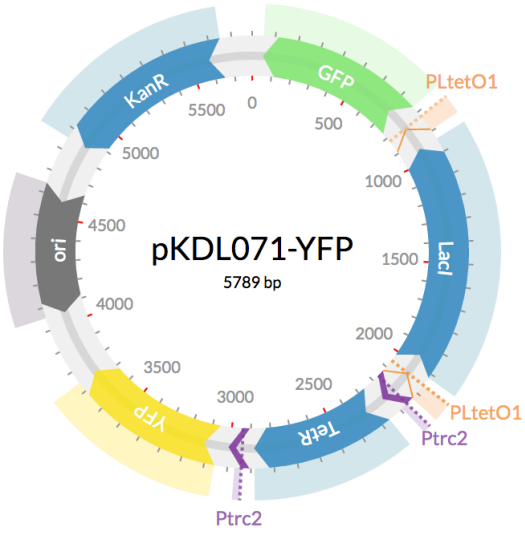

Figure 5.15 pKDL071-YFP plasmid map.

appropriate voltages for the lasers that excite GFP and mCherry.

pSEVA281G was constructed by digesting pKDL071 and pSEVA281 using the protocol outlined in Section 5.7.2.6. pSEVA281 (supplied by Esteban Martinez García) is a plasmid backbone containing kanamycin resistance, a high copy origin of replication (pUC) and a multiple cloning site. The digested fragments were isolated using gel purification (Section 5.7.2.7) and then the isolated fragments were ligated (Section 5.7.2.8). *Escherichia coli* DH5α was then transformed with each plasmid (Section 5.7.2.9).

pSEVA281C was constructed via PCR cloning. PCR was carried out using the pKDL071 plasmid as a template DNA using the protocol outlined in Section 5.7.2.5. Primers were chosen so that P_{trc2} and mCherry were copied and a HindIII restriction enzyme recognition sequence added to the fragment. The rest of the cloning procedure followed as per plasmid pSEVA281G.

5.7.3.3 Growth rate investigation

I carried out a growth rate analysis to determine whether the aTc or IPTG added to pKDL071 or pSEVA281G *E. coli* cultures affected the growth of the bacteria. Cultures were grown without any inducer overnight as described in Section 5.7.2.13. Assays for the cultures were ran with and without added inducers. As can be seen in Figure 5.17, there is no difference between the conditions. The addition of either aTc or IPTG does not affect the growth rate of *E. coli* K-12 MG1655. Additionally,

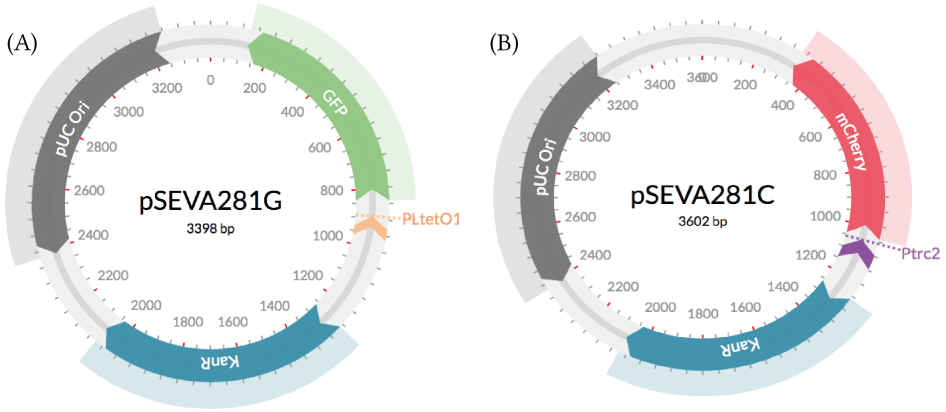

Figure 5.16 The plasmids used to calibrate GFP and mCherry fluorescence. (A) pSEVA281G plasmid map (B) pSEVA281C plasmid map.

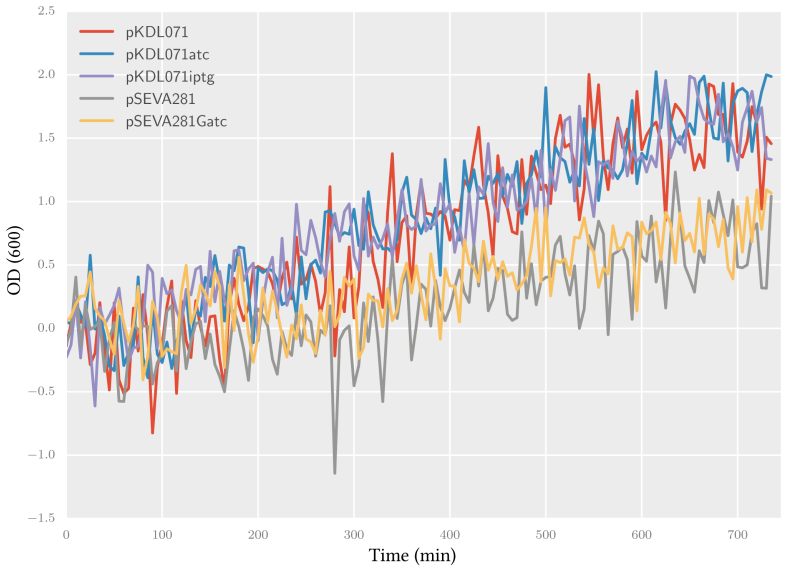

Figure 5.17 Growth rate analysis of *E. coli* K-12 MG1655 pKDL071 and *E. coli* DH5α pSEVA281G cultures with and without inducers. The inducers do not affect the growth of the bacteria.

aTc does not affect the growth rate of *E. coli* DH5α. Since the addition of aTc flips the switch to the GFP high state, and IPTG to the mCherry high state, we can also conclude that the growth rate of the chassis is not affected by which side of the switch is in the high state. The growth rate of *E. coli* DH5α was consistently lower than that of *E. coli* K-12 MG1655.

5.7.3.4 Toggle switch concentration assays

I carried out a concentration assay using flow cytometry, as described in Section 5.7.2.15. As can be seen in Figure 5.18A, during aTc induction the switch flips to a GFP high state when aTc concentration is at 0.09 ng mL^{-1} or higher. We observe a bimodal distribution at concentrations 0.07 ng mL^{-1} and 0.08 ng mL^{-1} close to the instantaneous switching point (at 0.09 ng mL^{-1}) where noise creates two simultaneous states. Therefore we observe part of the population switched to the GFP high state. In the case of IPTG induction (Figure 5.18B) we find that the switch flips to the mCherry high state when the concentration of IPTG is higher or equal to 0.001M. A decrease in GFP fluorescence is also observed. We do not observe a bimodal distribution in this case. The Hill functions for repression and activation were used to obtain the characterisation curves of the two inductions, aTc and IPTG and are given below.

$$F_{act} = P_{min} + (P_{max} - P_{min})\frac{\left(\frac{[I]}{Kd}\right)^n}{1+\left(\frac{[I]}{Kd}\right)^n}, \tag{5.8}$$

$$F_{rep} = P_{max} + (P_{min} - P_{max})\frac{(1)}{1+\left(\frac{[I]}{Kd}\right)^n} \tag{5.9}$$

where F is the median fluorescent unit and [I] is the concentration of inducer. P_{min} and P_{max} are the minimum and maximum fluorescence respectively, and Kd and n are the dissociation constant, and Hill coefficient respectively. I fit the Hill function by using the nonlinear least squares estimation in the R statistical environment (R Core Team 2008). The inferred values of the Hill function parameters P_{min}, P_{max}, Kd, and n are given in Table 5.10.

For the case of the aTc induction we observe a sharp switch between the GFP low to the GFP high state, as well as between the mCherry high to the mCherry low states, as can be seen in the characterisation curves in Figure 5.19B and D. This sharp switch made the fitting of the Hill function challenging. The parameters producing the best fit of the Hill function found are given in Table 5.10. The cooperativity parameter n is very high in this model, in order to be able to fit the data collected.

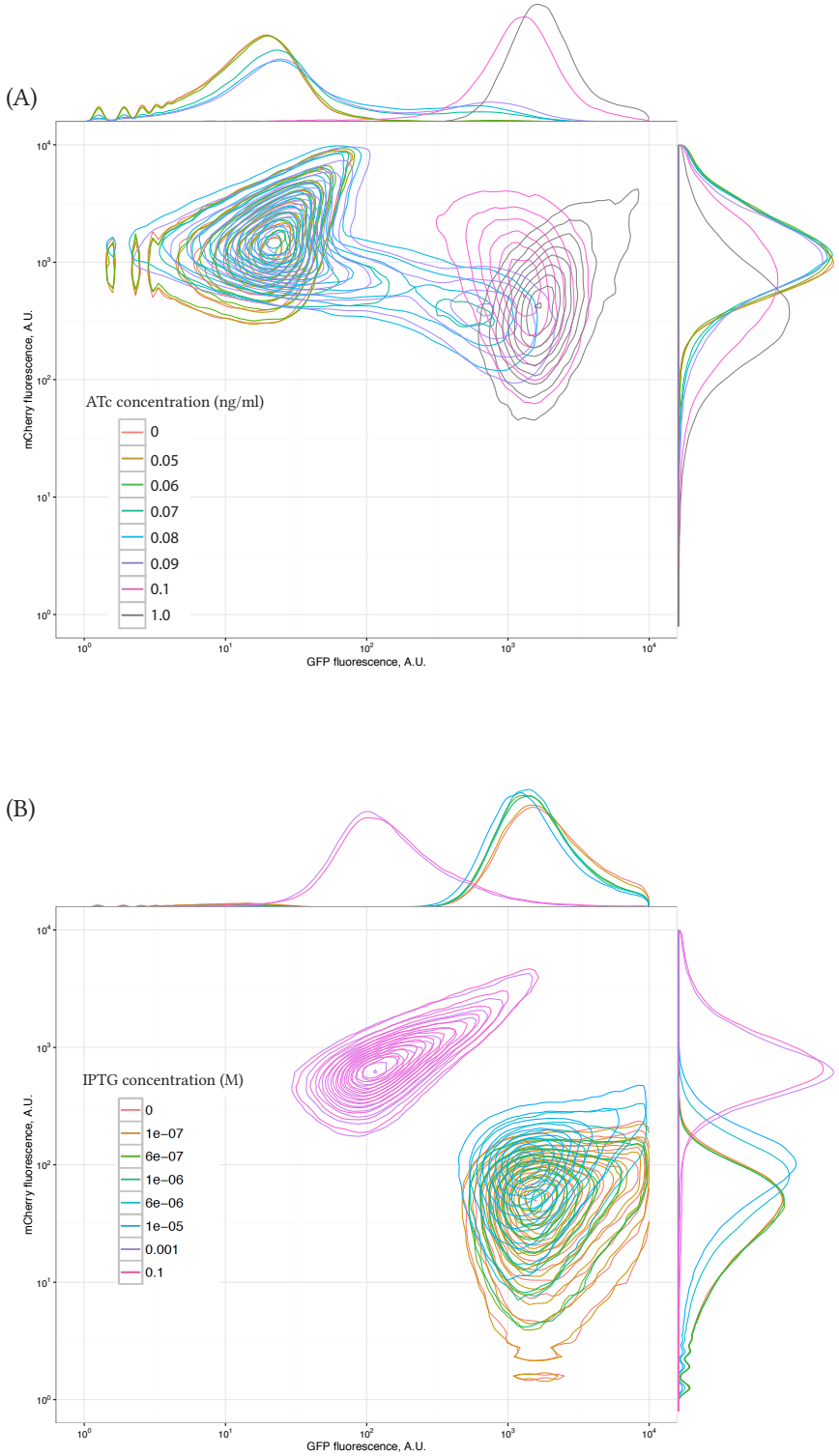

Figure 5.18 (A) aTc induction at various concentrations (B) IPTG induction at various concentrations.

Table 5.10 Inferred values from the Hill equations in aTc and IPTG inductions

Parameters	aTc induction		IPTG induction	
	GFP	mCherry	GFP	mCherry
P_{min}	18.3	330.45	139.69	7.3
P_{max}	1541.3	974.46	1392.822	687.3
kd	0.097	0.09917	0.000019	0.00012
n	56.7	135.006	2.59	0.98
fold change	84.1	2.94	9.97	94.15

This could be caused by discontinuous switching, which would mean that the Hill function is not appropriate for modelling its behaviour.

During IPTG induction we observe an increase in mCherry fluorescence, as seen in Figure 5.20. The parameters obtained via the nonlinear least squares estimation are given in Table 5.10. There is a 94.5 fold increase in mCherry fluorescence. We also observe a decrease in GFP fluorescence with increasing IPTG concentrations.

Figures 5.19 and 5.20 demonstrate that the genetic toggle switch present on the pKDL071 plasmid is capable of behaving like a switch. By adding the appropriate inducers at increasing concentrations I observed the switch flipping between its two states, GFP high/mCherry low and GFP low/mCherry high. I observed a bigger fold increase in fluorescence in mCherry during IPTG inductions compared to GFP during aTc inductions. Both inductions resulted in a large overall change in fluorescence for the two fluorescent proteins GFP and mCherry.

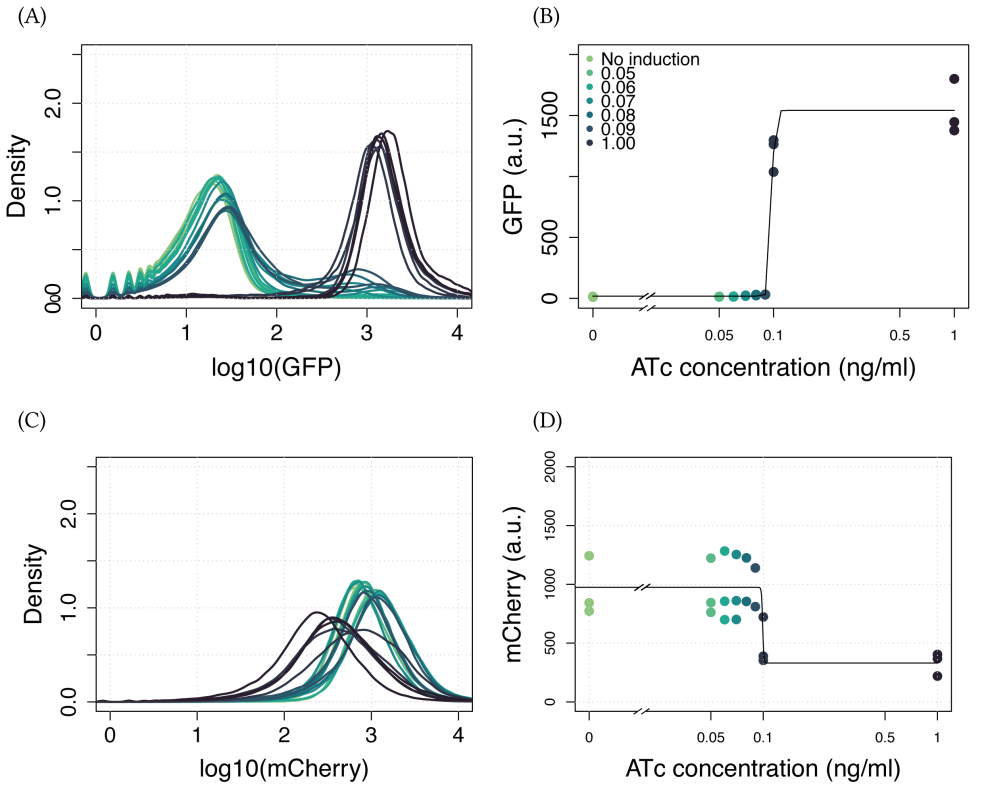

Figure 5.19 (A) Flow cytometry density plots of the logged GFP fluorescence obtained for each aTc induction. (B) There is an 84.1 fold increase in GFP fluorescence with increasing aTc concentration. (C) Flow cytometry density plots of the logged mCherry fluorescence obtained for each aTc induction. (D) The medians of the flow cytometry densities of the triplicates of aTc induction. We observe a decrease in mCherry fluorescence.

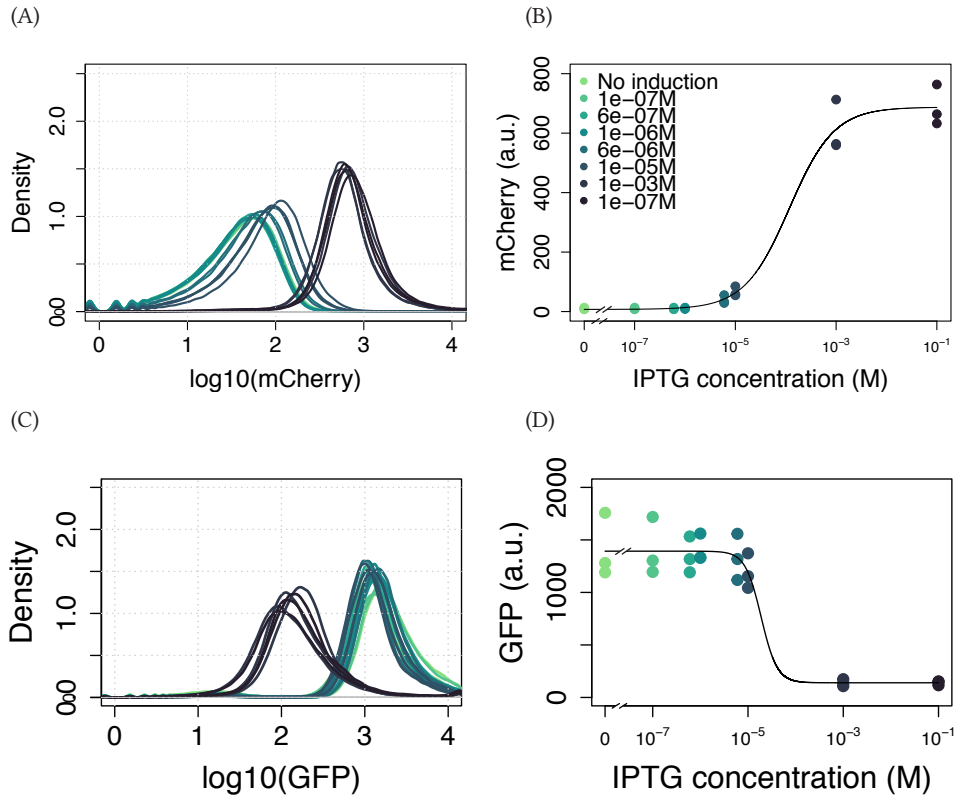

Figure 5.20 (A) Flow cytometry density plots of the logged mCherry fluorescence obtained for each IPTG induction. (B) There is a 94.5 fold increase in mCherry fluorescence with increasing IPTG concentration. (C) Flow cytometry density plots of the logged GFP fluorescence obtained for each IPTG induction. (D) The medians of the flow cytometry densities of the triplicates of IPTG induction. We observe a decrease in GFP fluorescence.

5.7.3.5 Toggle switch time course assay

I further analysed the pKDL071 toggle switch by investigating the time it takes for it to switch from one high state to the other. To do that I used the method outlined in Section 5.7.2.16. I obtained separate time courses for the IPTG and aTc inductions.

As can be seen in Figure 5.21 pKDL071 aTc induction begins switching 1 hour after induction. Complete induction is seen at 6 hours. During the IPTG induction (Figure 5.22) we see a bimodal distribution at 4 hours, and induction is complete at 6 hours. We observe that during aTc induction there is an increase in GFP fluorescence and a decrease in mCherry fluorescence, in the case of IPTG induction the increase in mCherry fluorescence is not as prominent. A decrease in GFP fluorescence is observed during IPTG induction.

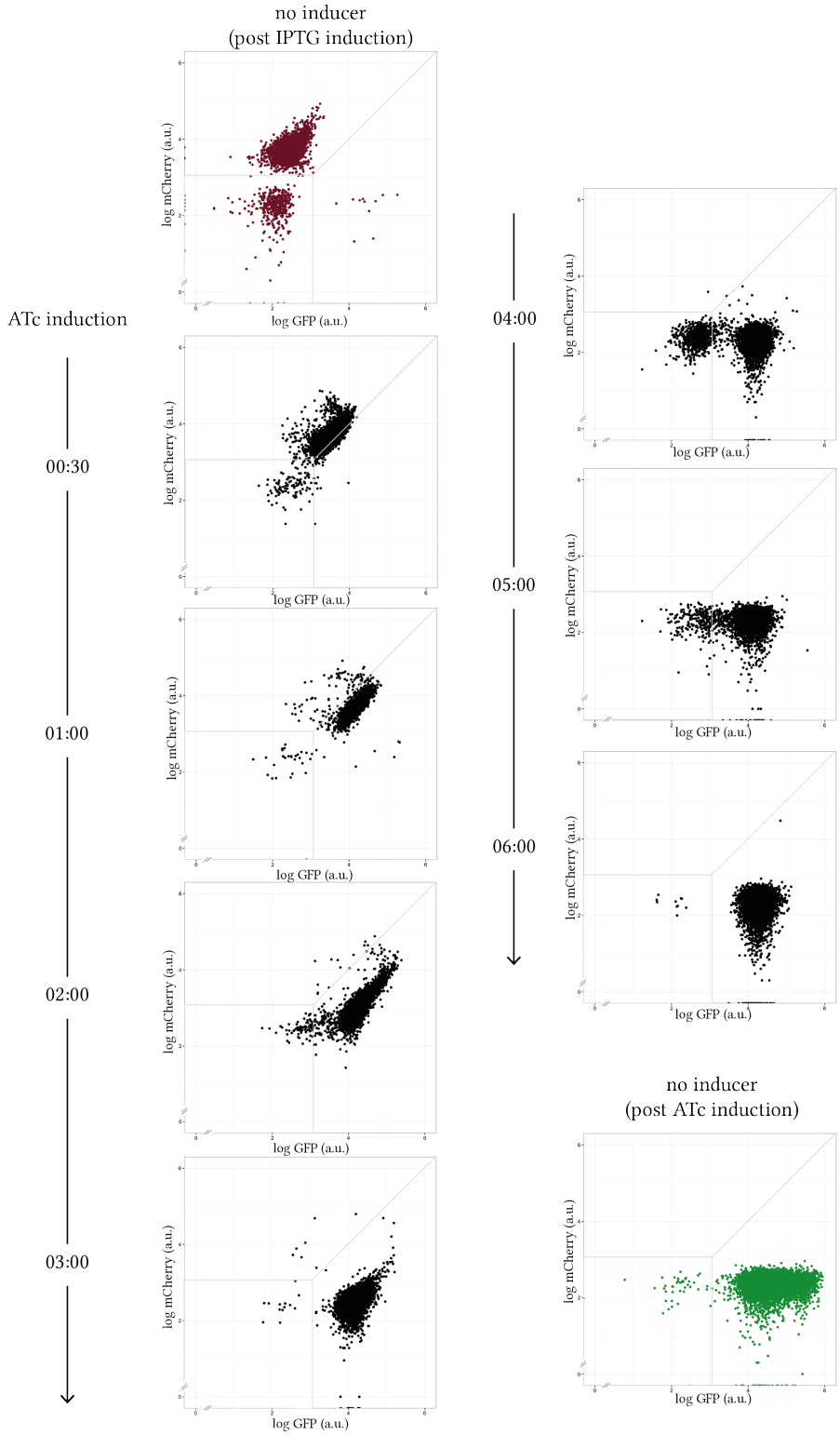

Figure 5.21 aTc induction of pKDL071 over time (in hours)

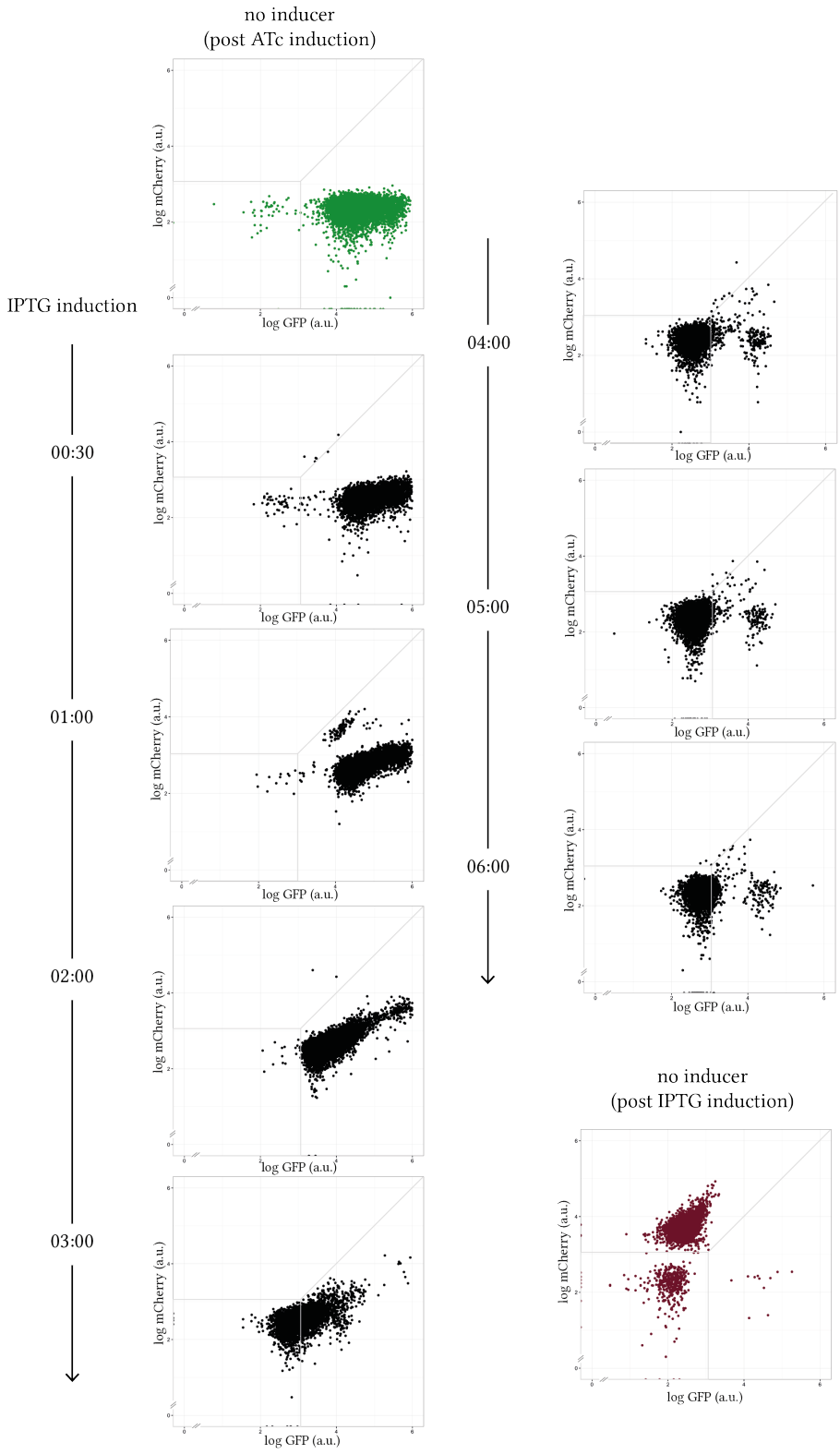

Figure 5.22 IPTG induction of pKDL071 over time (in hours)

In the next section I use ABC-Flow to fit a computational model to the time course data obtained. Prior to fitting a model to it, I process the data by removing the unresponsive populations. This ensures that the model is fitted only to the data from cells that respond to the inducers. As seen in Figure 5.21, during the aTc induction there is an unresponsive population of cells where GFP and mCherry fluorescence are both less that 10^3. This population is excluded from further analysis of the data. During the IPTG induction there is a population of cells that does not respond to the addition of IPTG by switching from GFP high to mCherry high. This population of cells was also excluded from further analysis.

5.8 ABC-Flow parameter inference for experimental data

In this section I apply ABC-Flow to the experimental flow cytometry data collected in Section 5.7.3.5. The data set is comprised of time course data of the Litcofsky et al. (2012) toggle switch. The two states of the switch are represented by the levels of GFP and mCherry intensity in each bacterial cell. Using aTc inducer, each cell transitions from a GFP low/mCherry high state to a GFP high/mCherry low state and using IPTG each cell transitions from a GFP high/mCherry low state to an GFP low/mCherry high state.

5.8.1 Toggle switch model developed to fit to flow cytometry data

The model used to fit the toggle switch time course assays was developed using the Shea-Ackers formalism which represents the probability of a given promoter expressing (Ackers, Johnson, & Shea 1982), as shown in Figure 5.23. The Shea-Ackers formalism is described in Section 2.2.2.2. The model represents the two promoters, P_{trc2} and $P_{Ltet\text{-}O}$ expressing mCherry and GFP respectively. The switch present in plasmid pKDL071 has been simplified to only take into account two genes, one for GFP and one for mCherry and it does not include LacI and TetR. Therefore in the model GFP represses the expression of mCherry and vice versa.

In order to take into account the stochastic dynamics of the system, the Gillespie algorithm is used in ABC-Flow, and thus the toggle switch model is described by the following hazards:

$$h_1 = KD_u(1 + KI_u)GFP \tag{5.10}$$

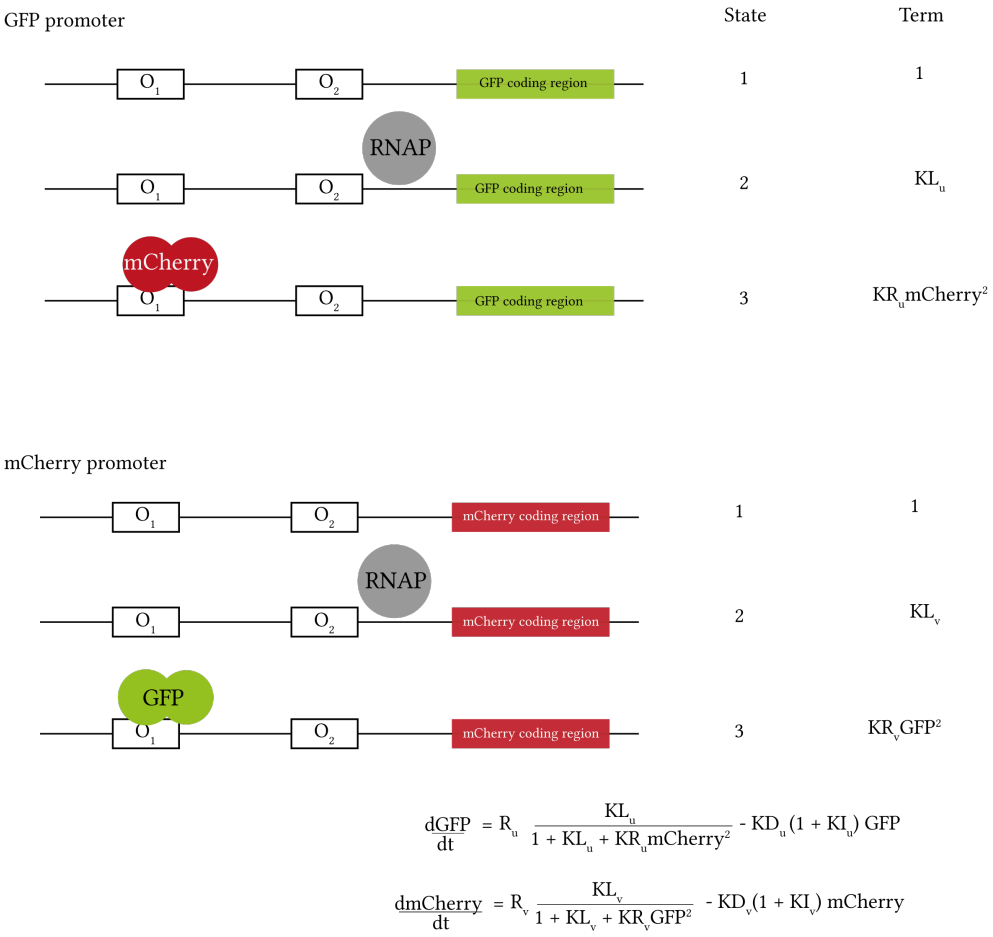

Figure 5.23 pKDL071 switch model using the Shea-Ackers formalism

$$h_2 = 60\frac{R_u KL_u}{1 + KL_u + KR_u mCherry^2} \tag{5.11}$$

$$h_3 = KD_v(1 + KI_v)mCherry \tag{5.12}$$

$$h_4 = 60\frac{R_v KL_v}{1 + KL_v + KR_v GFP^2}, \tag{5.13}$$

where GFP and mCherry represent the two fluorescent proteins in the system. The cooperativity of the repressors (GFP and mCherry) is assumed in the model as well as the dimerisation of the repressors. Parameters KI_u and KI_v increase the degradation of one of the species, and simulate the addition of a repressor, IPTG and aTc respectively. When using this model to fit to the post-aTc induction time course data, KI_u was set to 0. KI_v was set to 0, until t = 0.01h where the prior distribution was used. This was done to simulate the addition of the inducer. When using this model to fit the post-IPTG induction time course, KI_v was set to 0 and KI_u sampled

from the prior after t = 0.01h.

The two production hazards, $h2$ and $h4$ are multiplied by 60 to reflect the copy number of the toggle switch plasmid in each cell. The plasmid containing the toggle switch used here, pKDL071, contains the ColE1 origin of replication, and thus 50-70 copies of the plasmid are present in each cell (Milo et al. 2010). The priors used in ABC-Flow for this model are given in Table 5.11. All priors given assume a uniform distribution. The values of the prior were chosen in agreement with (Lillacci & Khammash 2013) and in reference to http://bionumbers.hms.harvard.edu/, the database of useful biological numbers (Milo et al. 2010).

The model also included a fluorescence intensity component. The fluorescent signal emitted by each individual fluorescent molecule was assumed to be normally distributed around a mean μ and standard deviation σ, as defined in Equation 5.1. In order to account for background fluorescence I used the OFF state of each side of the switch. Overnight inductions of the plasmid pKDL071, with aTc or IPTG, resulted in the switch fully flipped to each state mCherry low/GFP high and mCherry high/GFP low respectively. The fluorescence levels of the fluorescent proteins in the low state were assumed to be the background fluorescence detected by the flow cytometer. This was assumed to be a sample from a normal distribution with $\mu = 100$ and $\sigma^2 = 50$ that was added to the signal at each time point. The populations were allowed to progress until the reduction in epsilon started to plateau and the acceptance rate became very low. This indicates that the fit will not improve significantly with subsequent populations.

Table 5.11 The priors used for the 1D and 2D ABC-Flow model fitting to flow cytometry data

Parameters				
Description	Symbol	Units	aTc induction	IPTG induction
IPTG-induced mCherry degradation rate	KI_u	h^{-1} μM^{-1}		1 - 100
GFP transcription rate	R_u	molecules h^{-1}	1 - 50	1 - 50
GFP promoter leakiness	KL_u	molecules h^{-1}	1 - 50	1 - 50
GFP degradation	KD_u	h^{-1}	0.001 - 0.1	0.001 - 0.1
mCherry-induced GFP repression rate	KR_u	molecules^{-1} h^{-1}	0.016 - 1.2	0.016 - 1.2
mCherry transcription rate	R_v	molecules h^{-1}	1 - 50	1 - 50
mCherry promoter leakiness	KL_v	molecules h^{-1}	1 - 50	1 - 50
GFP-induced mCherry repression rate	KR_v	molecules^{-1} h^{-1}	0.016 - 1.2	0.016 - 1.2
mCherry degradation	KD_v	h^{-1}	0.1 - 2	0.1 - 2
aTc-induced GFP degradation rate	KI_v	h^{-1} μM^{-1}	1 - 100	
Species				
GFP		mM	0 - 1	100 - 1000
mCherry		mM	100 - 1000	0 - 1
Intensity parameters				
Mean of fluorescence of single GFP molecule	μ_{GFP}	AU	5 - 200	5 - 200
Mean of fluorescence of single mCherry molecule	$\mu_{mCherry}$	AU	5 - 200	5 - 200
Standard deviation of fluorescence of single GFP molecule	σ_{GFP}	AU	5 - 200	5 - 200
Standard deviation of fluorescence of single mCherry molecule	$\sigma_{mCherry}$	AU	5 - 200	5 - 200

5.8.2 Model fitting to the genetic toggle switch post aTc induction

The priors shown in Table 5.11, the hazard functions defining the model and the flow cytometry time course data were supplied to ABC-Flow. ABC-Flow was used to infer the parameter values that could produce the best fit to the experimental data. The resulting simulated time course data and posterior distributions are given in Figures 5.24 and 5.28 respectively. The model was also simulated stochastically without converting the number of molecules to fluorescence intensity in order to confirm that the model behaves like a switch. This is shown in Figure 5.24B. Following aTc induction, the number of GFP molecules increases and the number of mCherry molecules decreases.

The inferred parameters, and the 95% credible intervals, are given in Table 5.12. In order to validate these results I compare the 95% confidence intervals of the inferred parameters to values reported in the literature. First, the half-lives of the two proteins in the system, GFP and mCherry, are estimated. The half-life of a protein is given by

$$t_{\frac{1}{2}} = \frac{ln(2)}{k}$$

where k is the decay rate of the protein. The half-life of GFP was estimated to be (7.788, 43.322) hours which is consistent with the >24 hours half life that has been previously reported for the variant of GFP used here, GFPmut3b (Andersen et al. 1998). The half-life for mCherry was estimated to be (9.242, 31.507) hours which is consistent with the long half-life of mCherry, reported to be more than 24 hours (Shaner et al. 2004). We find that the inferred parameter for GFP gene expression, R_u is (1.055, 1.979) molecules h^{-1}, whereas the parameter for gene expression of mCherry (R_v) was inferred to (8.970, 42.611) molecules h^{-1}. These values correspond to the promoter strengths of $P_{LtetO-1}$ and P_{trc2} respectively. This is in agreement with characterisation data on the strength of these two promoters, which report that there is a difference in strengths, P_{trc2} being a stronger promoter than $P_{LtetO-1}$ (Litcofsky et al. 2012). By simulating the system using the median inferred values of the system, we find that when the switch is in the GFP high state, there are approximately 600 GFP molecules in the cell. Further, we find that the values of the intensity parameters μ and σ, which represent the settings on the flow cytometer, to be inferred better for GFP than for mCherry.

The inferred parameters can be used to study the switch system present in the pKDL071 plasmid. Here I examined the effect that the values of the most well in-

Table 5.12 The inferred parameter values of the toggle switch post-aTc induction

Parameter	Units	0.025	Median	0.975
R_u	molecules h^{-1}	1.055	1.298	1.979
KL_u	molecules h^{-1}	19.606	40.466	49.138
KD_u	h^{-1}	0.016	0.0834	0.089
KR_u	molecules^{-1} h^{-1}	0.006	0.013	0.093
R_v	molecules h^{-1}	8.970	31.057	42.611
KL_v	molecules h^{-1}	5.807	35.396	38.333
KR_v	molecules^{-1} h^{-1}	0.095	0.386	0.947
KD_v	h^{-1}	0.022	0.043	0.075
KI_v	h^{-1} μM^{-1}	21.690	54.214	89.937
μ_{GFP}	AU	53.424	76.229	84.055
$\mu_{mCherry}$	AU	91.329	108.691	176.058
σ_{GFP}	AU	154.373	193.198	196.525
$\sigma_{mCherry}$	AU	68.178	115.581	139.816

ferred parameters, R_u and KR_u as well as the initial conditions have on the behaviour of the system. The results are shown in Figure 5.26. First, I studied the effect of the initial conditions of the dominant protein, in this case mCherry has on the system. In order to do that I simulated the model using the median values from the posterior distribution, and increased the initial condition value of mCherry. We do not find this to have had an effect on the final state of the system, as GFP reaches a similar value at 6 hours as the one produced using inferred initial condition values (shown in Figure 5.26B). On the other hand, decreasing the initial condition values of GFP to 10 molecules, destabilises the switch, as shown in Figure 5.26B. The switch happens at a later time point, and it does not take place for all stochastic trajectories.

The value of R_u represents the strength of the promoter driving GFP expression. It is therefore important to understand the effect that the promoter strength has on the behaviour of the system. In order to do that, I simulated the model with values that exceeded the 95% credible region of the values inferred from ABC-Flow. The results are shown in Figure 5.26C and D. We find that if the promoter strength is lower than the range of the 95% credible region of the inferred value then the system stops behaving like a switch. If the value is much higher, GFP reaches a much higher value after 6 hours. Further, I examined the effect that the strength of KR_u, the parameter representing the mCherry-induced GFP repression rate, has to the behaviour of the system. We find that KR_u affects the ability of the system to behave like a switch. If KR_u is set to 0.1, the switch occurs at a later time point, and

does not take place for all stochastic trajectories. If KR_u is set to much higher than the 95% credible region, the system does not switch. These findings are important in the understanding the system under study as they allow the above predictions to be made about the behaviour of the system.

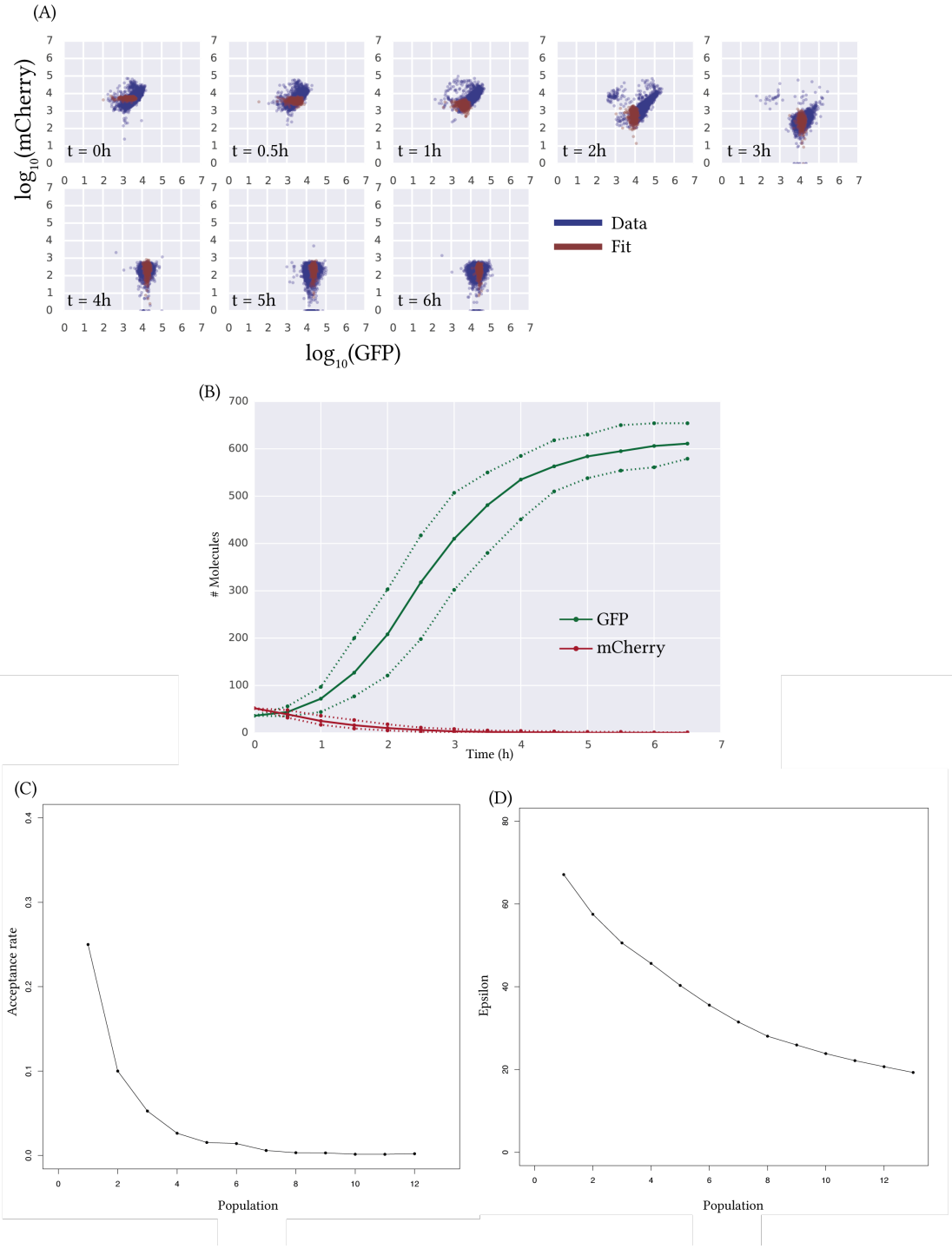

Figure 5.24 (A) The post-aTc induction flow cytometry time course data (blue) and the resulting model fit from ABC-Flow (red). (B) The model simulation using parameters sampled from the posterior distribution shows that this model has a high and low state for GFP. A solid line depicts the median value of the trajectories and the dotted line depict 0.025 and 0.975 quartiles. (C) The particle acceptance rate from ABC-Flow (D) The progression of epsilon threshold values at each population.

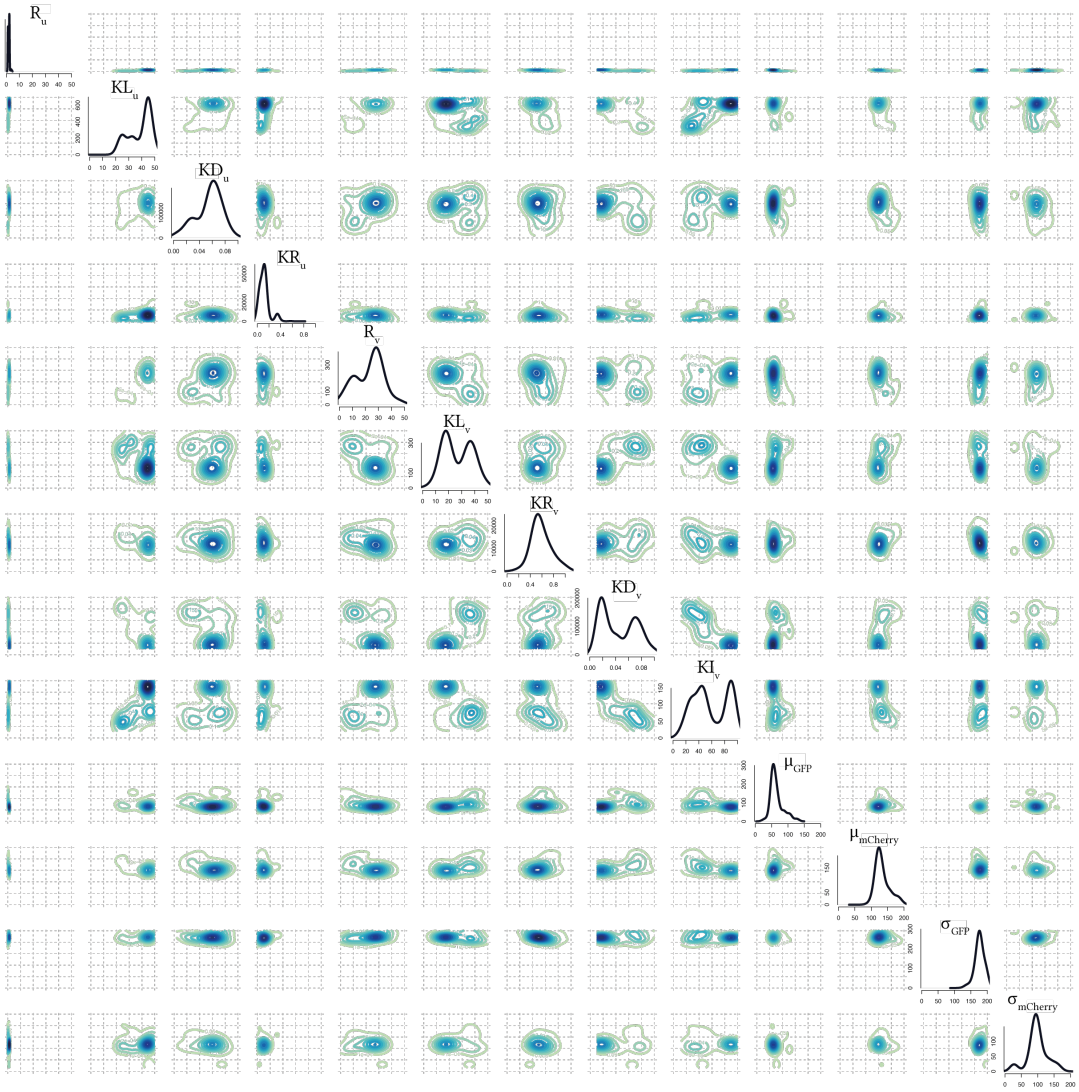

Figure 5.25 The posterior distributions of the 13 parameters fitted to post-aTc induction time course data using ABC-Flow. We find that the parameters for GFP expression (R_u) and repression (KR_u) are the most identifiable.

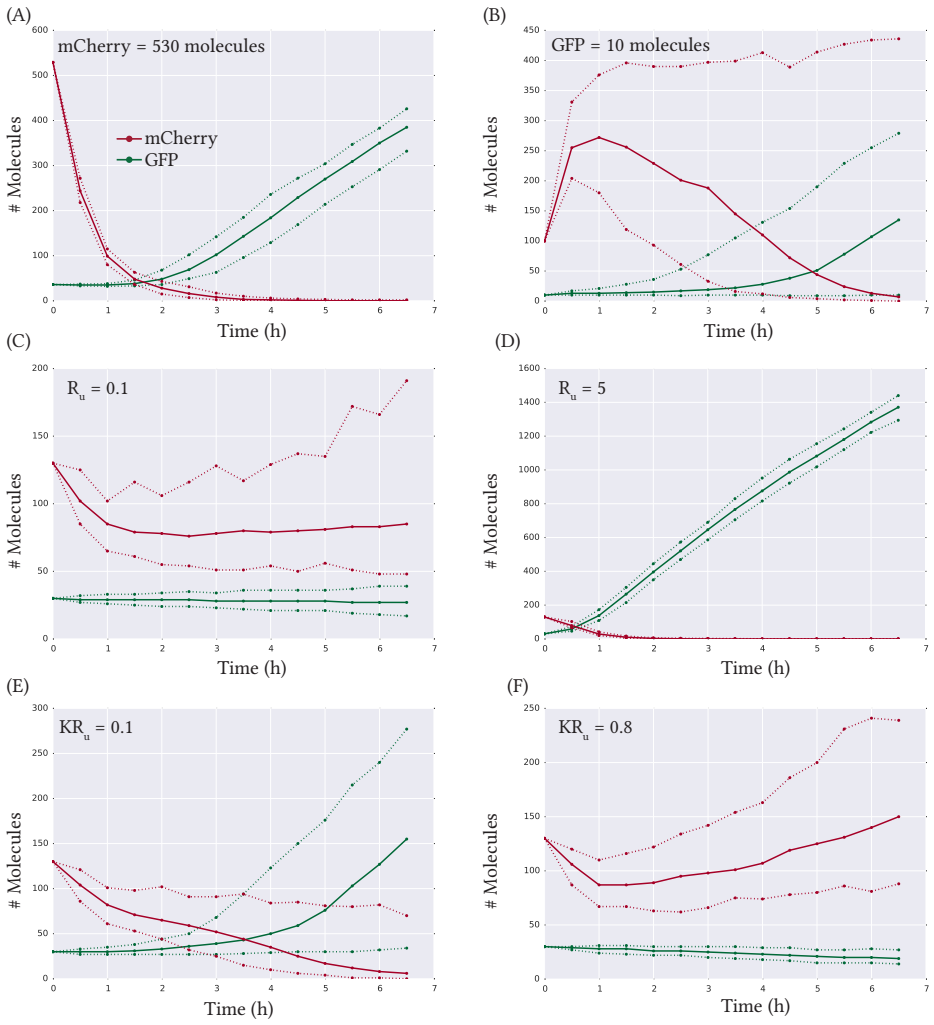

Figure 5.26 (A) Increasing the initial condition value of mCherry does not affect the state of the system. (B) When the initial condition value of GFP is decreased, the system no longer exhibits a high and low state for GFP and mCherry. (C) Decreasing parameter R_u to a value outside the 95% credible region stops the model from behaving like a switch. (D) Increasing the value of R_u, the rate representing the transcription rate of GFP, causes GFP to reach higher levels. (E) Increasing the value of KR_u, the parameter representing the mCherry-induced GFP repression rate causes the switch to happen at a later time point and not all stochastic trajectories switch state. (F) If the value of KR_u is set to larger than the 95% credible region, the system does not switch states. For all plots, a solid line depicts the median value of the trajectories and the dotted line depict 0.025 and 0.975 quartiles.

5.8.3 Model fitting to the genetic toggle switch post IPTG induction

ABC-Flow was further used to fit the experimental time course obtained from the toggle switch IPTG induction. The prior densities used are given in Table 5.11, the hazard functions of the model as well as the flow cytometry data were provided to ABC-Flow. The resulting time course of the model fitted to the experimental flow cytometry data is shown in Figure 5.27 and the median and 95% confidence intervals of the inferred parameters are shown in Table 5.13.

The median values of the marginal posterior distribution of each parameter were used to simulate the model without the conversion to fluorescent intensity in order to confirm that the model behaved like a switch. As can be seen in Figure 5.27B, the model does not behave like a switch within the timeframe given from the experimental data (0 - 6 hours). We find a rapid decay of GFP without an increase in mCherry fluorescence as would be expected. This could be attributed to the experimental time course obtained. As shown in Figure 5.27A, over a period of 6 hours post induction there is a decrease in GFP fluorescence. mCherry can be seen increasing after two hours post induction but then not maintaining that high level. Over the 6 hours, there is no overall increase in mCherry fluorescence. This time course is challenging to fit using the model used here as it does not behave like a switch as expected. The epsilon progression of the fit for the IPTG induction of the switch further confirms that the fit to the data is not as good as expected. Epsilon reduction levels off at a high epsilon value compared to the aTc induction, while the acceptance rate is very low. This indicates that continuing with the fit of the above model to the IPTG induction data will not produce a better fit.

The posterior density obtained from by ABC-Flow is given in Figure 5.28. Parameter R_u, representing GFP expression, was inferred to be (1.057, 9.707) molecules h^{-1}, within the 95% credible region of the posterior. This is in good agreement with the inferred values obtained for the post-aTc induction. The half-life values obtained for GFP and mCherry were found to be (8.35, 46.20) and (7.61, 77.016) respectively, which are in good agreement with reported literature values (Shaner et al. 2004; Andersen et al. 1998).

The two fits obtained from the post-aTc and post-IPTG induction represent the different sides of the switch from the same genetic system and should thus be considered together. Comparing the fits we find a good agreement for the inferred parameters R_u and KD_u. On the other hand we find that the inferred values for KR_u and KD_v do not agree. This could be caused by the fact that the switch is not perfectly symmetric on the two sides, as can be seen from the experimental data. The

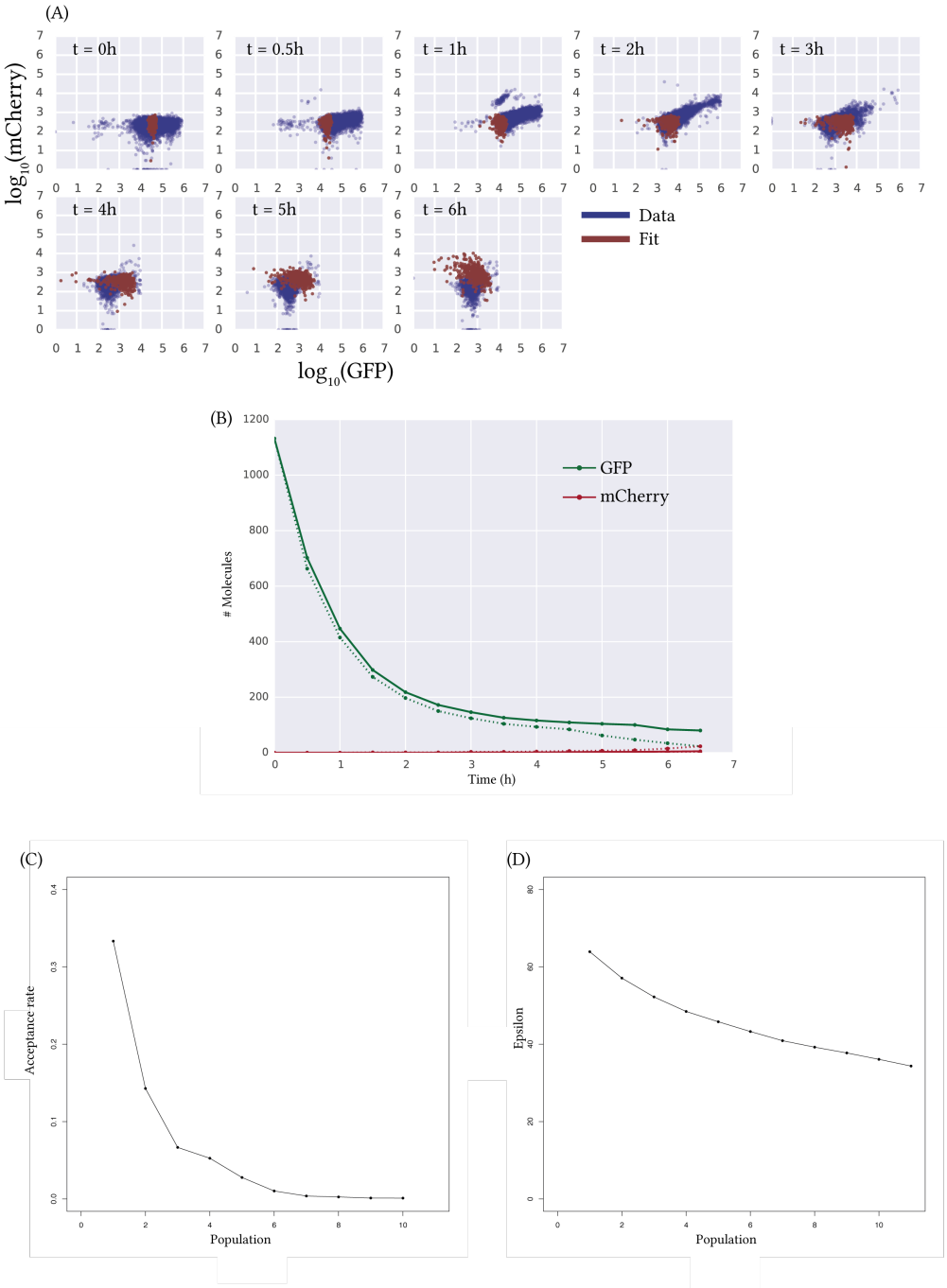

Figure 5.27 (A) The time course data obtained of the post-IPTG induced toggle switch is shown in blue and the resulting fit from ABC-Flow is shown in red. (B) The model was simulated by using parameter sampled from the posterior distribution. The resulting model did not behave like a switch. (C) The particle acceptance rate from ABC-Flow (D) The progression of epsilon threshold values at each population.

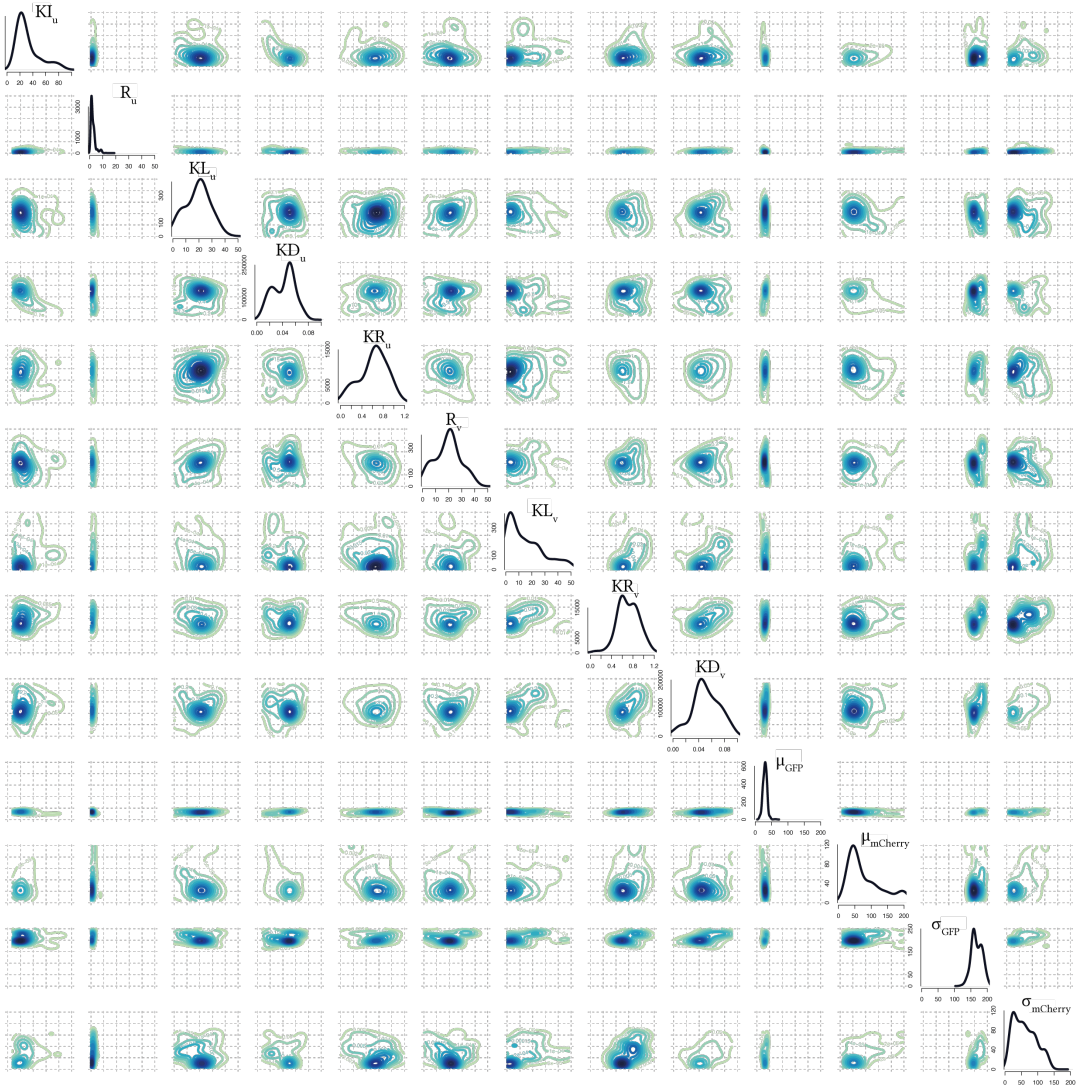

Figure 5.28 The posterior distribution obtained from ABC-Flow for the post-IPTG time course data. The parameter for GFP expression (R_u) was found to be the most well inferred.

Table 5.13 The inferred parameter values of the toggle switch post-IPTG induction

Parameter	Units	0.025	Median	0.975
KI_u	$h^{-1}\mu M^{-1}$	11.767	33.094	59.814
R_u	molecules h^{-1}	1.057	1.818	9.707
KL_u	molecules h^{-1}	1.161	4.986	28.258
KD_u	h^{-1}	0.015	0.031	0.083
KR_u	molecules^{-1} h^{-1}	0.146	0.839	1.183
R_v	molecules h^{-1}	1.119	3.223	21.715
KL_v	molecules h^{-1}	1.396	11.515	33.369
KR_v	molecules^{-1} h^{-1}	0.461	0.758	1.136
KD_v	h^{-1}	0.009	0.061	0.091
μ_{GFP}	AU	19.371	34.634	46.499
$\mu_{mCherry}$	AU	17.082	34.943	138.208
σ_{GFP}	AU	151.169	179.357	197.939
$\sigma_{mCherry}$	AU	9.907	52.085	89.492

system responds to aTc faster than to IPTG. This difference in the data could lead to different values inferred for the underlying parameters. In the future this could be incorporated in to ABC-Flow to improve the accuracy of the inference.

In this section I used ABC-Flow to fit a toggle switch model to experimental flow cytometry data. Both sides of the switch were examined, aTc induction which flips the switch from mCherry high/GFP low to mCherry low/GFP high and IPTG induction, which flips the switch from GFP high/mCherry low to GFP low/mCherry high. The model was successfully fit to the data set obtained from the aTc induction of the switch but less so to the data obtained from the IPTG induction of the switch.

5.9 Discussion

In this Chapter I characterised the genetic toggle switch experimentally. First I studied the effect of the two inducers aTc and IPTG on the growth rate of the selected chassis *E. coli* K-12 MG1655. I find that there is no detrimental effect to the bacterium by the inducers. I further characterised the switch by determining the minimum inducer concentration necessary to change the state of the switch. I find that for aTc induction, a minimum of 0.09 ng mL^{-1} is required to cause the switch to go to a GFP high state. For IPTG induction I find that a minimum of 0.001 M is required to flip the switch to an mCherry high state. This information is critical for using this switch in other applications. Both sides of the switch are very sensitive

to inducer concentrations, as the concentrations required to observe a change in fluorescence are very small. Further, I found that the switch is instantaneous for both GFP and mCherry, with no intermediate fluorescence levels observed.

Furthermore I found that this toggle switch, pKDL071, is faster to respond to a change in aTc concentration that to a change in IPTG concentration. For IPTG induction we observe a change in fluorescence after 3-4 hours of induction. For aTc induction we can see a difference within an hour of induction. This result is in agreement with Litcofsky et al. (2012). This difference in response times must be taken into account when using the pKDL071 switch for other applications. This difference could be attributed to maturation times of the fluorescent proteins. Macdonald, Chen, & Mueller (2012) found that mCherry half-maturation time is 150 mins, whereas the GFP variant used here, GFPmut3b has been especially mutated for fast action (Cormack, Valdivia, & Falkow 1996). Cormack, Valdivia, & Falkow (1996) found that whereas wild type GFP is detectable 1-2 hours after induction, GFPmut3b is detectable 8 minutes after induction. This difference could account for the different response times observed.

Here I also developed a Bayesian framework, ABC-Flow, that is used to fit stochastic models to flow cytometry data. Fitting computational models to flow cytometry data can be challenging; fluorescence intensity is measured in arbitrary units and there can be a big variability between experiments depending on instrument settings. This poses a challenge for model fitting as the fluorescence intensity emitted by each individual fluorophore molecule cannot be reliably estimated (Kelwick et al. 2014). ABC-Flow converts the number of molecules obtained via simulations to fluorescence intensity in order to overcome current limitations in fitting computational models to flow cytometry data. The novelty of ABC-Flow is that it can be used on two-dimensional flow cytometry data. Unlike Lillacci & Khammash (2013), it can be used to infer the parameter values of systems involving one or two fluorescent molecules. This makes it ideal to be used on the genetic toggle switch, whose behaviour is reflected by the levels of two fluorescent proteins, GFP and mCherry.

I have used ABC-Flow to fit the toggle switch model to simulated flow cytometry data in one and two dimensions. This demonstrated the effectiveness of ABC-Flow in parameter identifiability of intensity data. Further, I used ABC-Flow to fit a stochastic computational model to flow cytometry time course data obtained by inducing the genetic toggle switch to its two states. This was done using both sides of the switch, GFP high/mCherry low to GFP low/mCherry high using IPTG and vice versa using aTc. The model parameters were inferred from the data obtained via aTc

induction, representing the flip from GFP low/mCherry high to GFP high/mCherry low. The inferred parameter values were used to make predictions on the behaviour of the toggle switch system under different initial conditions, promoter and repression strengths. The parameters were not inferred as well for the data obtained via IPTG induction. This could be attributed to the experimental data obtained. Post-IPTG induction we observed a decrease in GFP but the increase in mCherry was not as prominent. This result could be improved by a repetition of the time course experiment, which was not carried out due to time constraints.

The model fits to experimental data presented here could be improved by a number of ways in the future. Firstly, the toggle switch model used here was a simplified version of the system. The model consisted of two proteins, GFP and mCherry and used the Shea-Ackers formalism. A more flexible and detailed model, like one built using mass action kinetics and including LacI and TetR, as well as mRNA and the maturation of GFP could lead to more accurate inference. It would also allow the testing of more aspects of the system via simulations using the inferred parameters. In addition, the model fits would be improved if the flow cytometry data was further pre-processed by calibration using commercially available calibration beads. Computational methods like FlowCal, developed by Tabor et al. (2009) can be used to convert fluorescence arbitrary units (a.u.) to MEFs (molecules of equivalent fluorophore). This can account for instrument gain settings as well as day to day instrument variability. Future improvements on ABC-Flow could also include the simultaneous fitting of the toggle switch model to both time courses, post-aTc and post-IPTG induction. Both of these time courses are obtained from the same genetic system, and the accurate characterisation of said system would have to include both functions. This would allow us to obtain parameter estimates for components that can respond to both inducers. ABC-Flow could be also be further developed to be able to fit computational models to more fluorophores simultaneously. This would enable the effective characterisation of more complex systems.

The framework developed here can aid the advance of the understanding of genetic systems. ABC-Flow can be used to characterise a system, and infer the parameter values that give rise to the experimental flow cytometry data. This will allow the accurate parameterisation of computational models describing the system, a known bottleneck in quantitative model building (Le Novère 2015). A parameterised model can be used to further the understanding of the system and make testable predictions for its behaviour. It can be used to study how the fluxes and concentrations of the species are influenced by the parameter values (Li et al. 2010),

such as transcription or translation rate. This allows for the prediction of the response of the system under different stresses *in silico*. The different parts of the system under study, working together to produce the response observed, can be disentangled, understood and studied separately without the need for numerous laborious and time consuming experiments.

5.10 Summary

In this chapter I developed ABC-Flow, a Bayesian framework used to fit computational models to flow cytometry data. I tested the method using simulated data. I summarised the experiments carried out for the analysis of the genetic toggle switch. I used the pKDL071 plasmid and characterised its switching behaviour over various inducer concentrations and over time. I found the concentration of each inducer necessary to flip the switch as well as the time it takes for the change to be observed. The time course experiments were used as input to ABC-Flow in order to fit a computational model to the data and infer the parameter values. In the next Chapter I outline an experimental design to construct more robust genetic toggle switches.

6 Designing new switches

6.1 Introduction

In the previous Chapters I studied the effect that adding positive feedback loops to the genetic toggle switch has on the robustness of the system. I found that adding two positive feedback loops to the simple toggle switch can increase its parametric robustness. The next step in this analysis would be to test these predictions experimentally. Therefore, in this Chapter I provide the experimental design for the construction of the genetic toggle switch with single and double positive autoregulation. The constructed switches could then be compared to the simple Litcofsky et al. (2012) toggle switch experimentally. Their robustness could be tested by varying the experimental conditions, like temperature and pH, and measuring the response of the switch.

Structurally, this Chapter is organised as follows: First I provide an overview of the cloning plan, by listing the relevant BioBrick parts used and their interactions. Then I outline the experimental design for producing these switches.

6.2 Cloning overview

The Litcofsky et al. (2012) toggle switch plasmid, pKDL071, used in Chapter 5 is modified to construct three new switches. Two switches will have single positive autoregulation, one on each gene, and one switch will have positive autoregulation on both genes. An overview of the cloning stages to be carried out is shown in Figure 6.1. The three stages required for the cloning plan to be completed are outlined in the sections below.

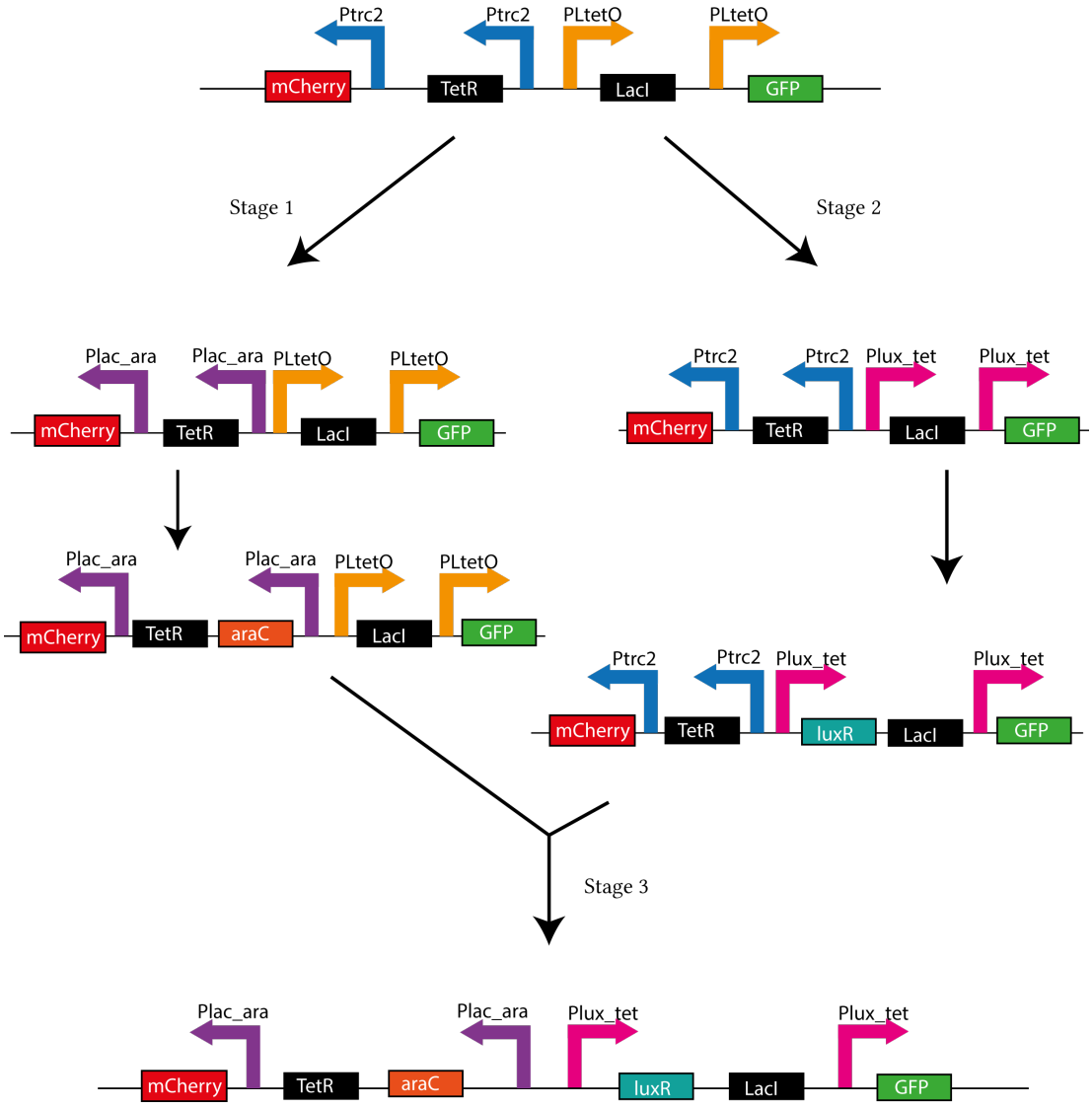

Figure 6.1 An overview of the cloning plan to produce three new switches, two with single positive autoregulation and one with double positive autoregulation.

6.2.1 Resulting switches

The three switches shown in Figure 6.2 will be constructed through this cloning process. The first switch, on plasmid pKDL071-plac/ara-araC is a toggle switch with positive autoregulation on the TetR/mCherry side of the switch. The second plasmid, pKDL071-pLuxTet-luxR consists of a toggle switch with positive autoregulation on the LacI/GFP side of the switch. Finally, the switch with positive autoregulation on both sides of the switch is on the pKLD0713a plasmid. The plasmid maps and a schematic of their components' interactions are shown in Figure 6.2.

6.3 Experimental design

The construction of the three switches shown in Figure 6.2 is broken down in three stages, one for the construction of each switch. In this section I will outline the necessary cloning steps that need to be carried out in order to construct each switch. The detailed methods that will have to be used for each cloning step are described in Section 5.7.2. All primer sequences have been designed and are given in Appendix B. Following the construction of each plasmid outlined below, competent *E.coli* cells will be transformed following the method outlined in Section 5.7.2.9.

6.3.1 Stage 1 - Construction of pKDL071-plac/ara-araC

In order to construct plasmid pKDL071-plac/ara-araC with single positive autoregulation on the mCherry/TetR side, the P_{trc2} promoter will be swapped for the $P_{lac\_ara\text{-}1}$ and AraC added upstream of TetR. $P_{lac\_ara\text{-}1}$ is activated by arabinose (AraC) and repressed by LacI (Lutz & Bujard 1997), and is thus ideal. The $P_{lac\_ara\text{-}1}$ promoter is present in the pJS167 plasmid, which is provided by Jeff Hasty (Addgene plasmid # 48881) (Stricker et al. 2008). This promoter is also present in the BioBrick registry of standard biological parts as BBa_K1713000.

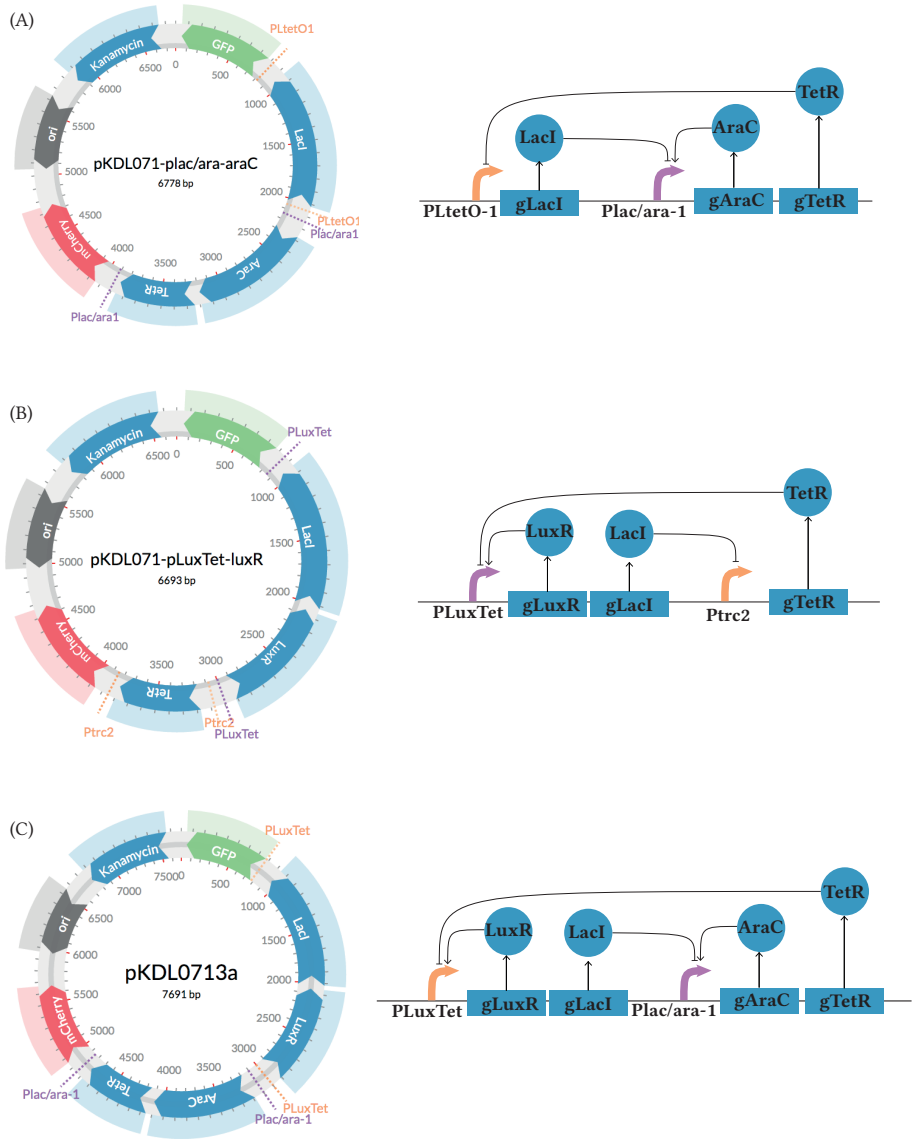

Figure 6.2 The plasmid maps of three new switches to be constructed. The first two switches, (A, B) have a single positive autoregulation on each side of the switch respectively. (C) The switch with double positive autoregulation.

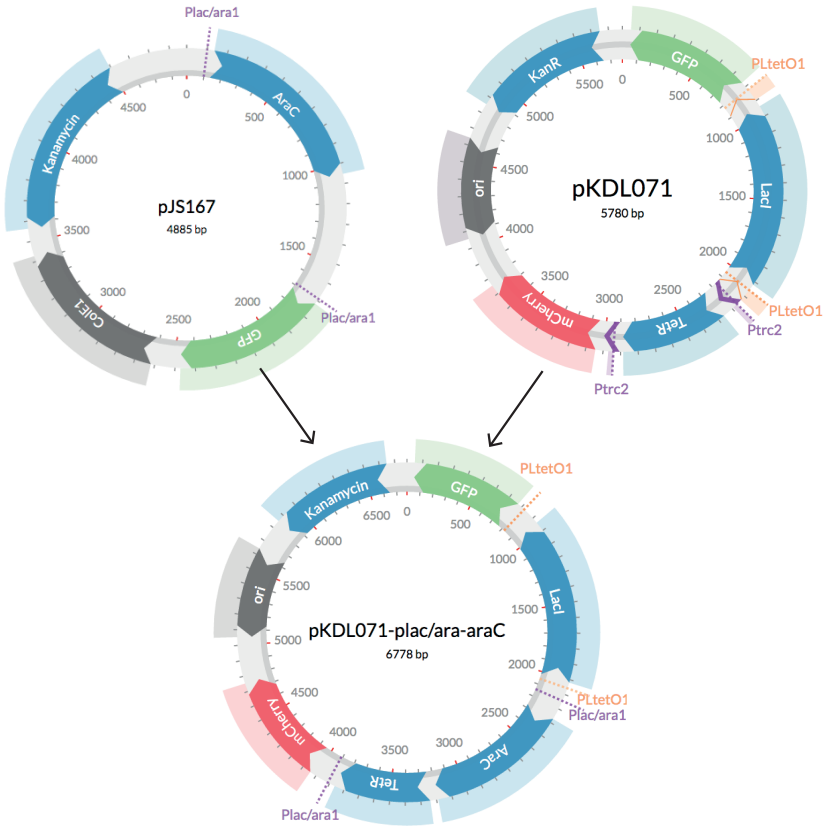

Figure 6.3 Stage 1 cloning procedure. The pKDL071-plac/ara-araC plasmid is constructed via PCR cloning from the pJS167 and pKDL071 plasmids.

Using PCR cloning, the $P_{lac\_ara\text{-}1}$ promoter will be cloned from pJS167, with added XmaI and KasI restriction enzyme sequences on each end. Both pKDL071 and the PCR product will be digested with XmaI and KasI restriction enzymes. After gel electrophoresis and gel extraction, the two will be subsequently ligated. The resulting plasmid should have the $P_{lac\_ara\text{-}1}$ promoter instead of the P_{trc2} promoter upstream of mCherry.

Following that, PCR cloning will be used to clone $P_{lac\_ara\text{-}1}$ and AraC from the pJS167 plasmid. EagI and SalI flanking sequences will be added via PCR on the 5' and 3' ends. Plasmid pKDL071 and the PCR product will be digested using EagI and SalI. Following gel electrophoresis and gel extraction, the two products will be ligated in order to complete the pKDL071-plac/ara-araC plasmid. The detailed methods for each cloning technique mentioned here can be found in Section 5.7.2.

6.3.2 Stage 2 - Construction of pKDL071-pluxtet-luxR

In order to construct the plasmid pKDL071-pluxtet-luxR, the $P_{Lux/tet}$ promoter is necessary. The $P_{Lux/tet}$ promoter is present in the BioBrick registry of standard biological parts as BBa_K934024. $P_{Lux/tet}$ is a hybrid promoter activated by LuxR and repressed by TetR. This promoter will be added in exchange of $P_{LtetO\text{-}1}$ to the pKDL071 plasmid. The LuxR gene will also be added upstream of LacI, in order to construct a switch with positive autoregulation on the LacI/GFP side.

First, the $P_{Lux/tet}$ promoter is synthesised as the reverse complement of BBa_K934024 with added flanking sequences of EcoO1091 and SphI on the 5' side and AclI and Eag1 at the 3' side. These are added to aid with further cloning steps. The sequence synthesised is given below:

5'- TTGGGACCTGCATGCTAATCTCTATCACTGATAGGGATAATACGAGTATCTC
TATCACTGATAGGGAGTAAACCTGTACGATCCTACAGGTAACGTTCGGCCG -3'

The pLux/tet and pKDL071 plasmids will subsequently be digested with SphI and AclI. Following gel extraction and ligation, the $P_{Lux/tet}$ promoter will be added upstream of GFP, replacing the $P_{LtetO\text{-}1}$ promoter. Then, the plux/tet and pKDL071 plasmids will be digested with EcoO109I and EagI restriction enzymes. Following gel extraction and digestion, the $P_{Lux/tet}$ promoter will be added upstream to LacI, replacing the $P_{LtetO\text{-}1}$ promoter.

The final stage of constructing the pKDL071-pluxtet-luxR plasmid consists of PCR cloning of the pTD103aiiA(Cm) plasmid with added BsGI flanking sequences

at both ends. The pTD103aiiA(Cm) is provided by Jeff Hasty (Addgene plasmid # 48886) (Prindle et al. 2012). The plasmid constructed in the previous step and the PCR product will be digested with BsGI restriction enzyme. Following gel extraction and ligation, the pKDL071-pluxtet-luxR should be complete. The ligated products will be transformed into thermocompetent *E.coli*.

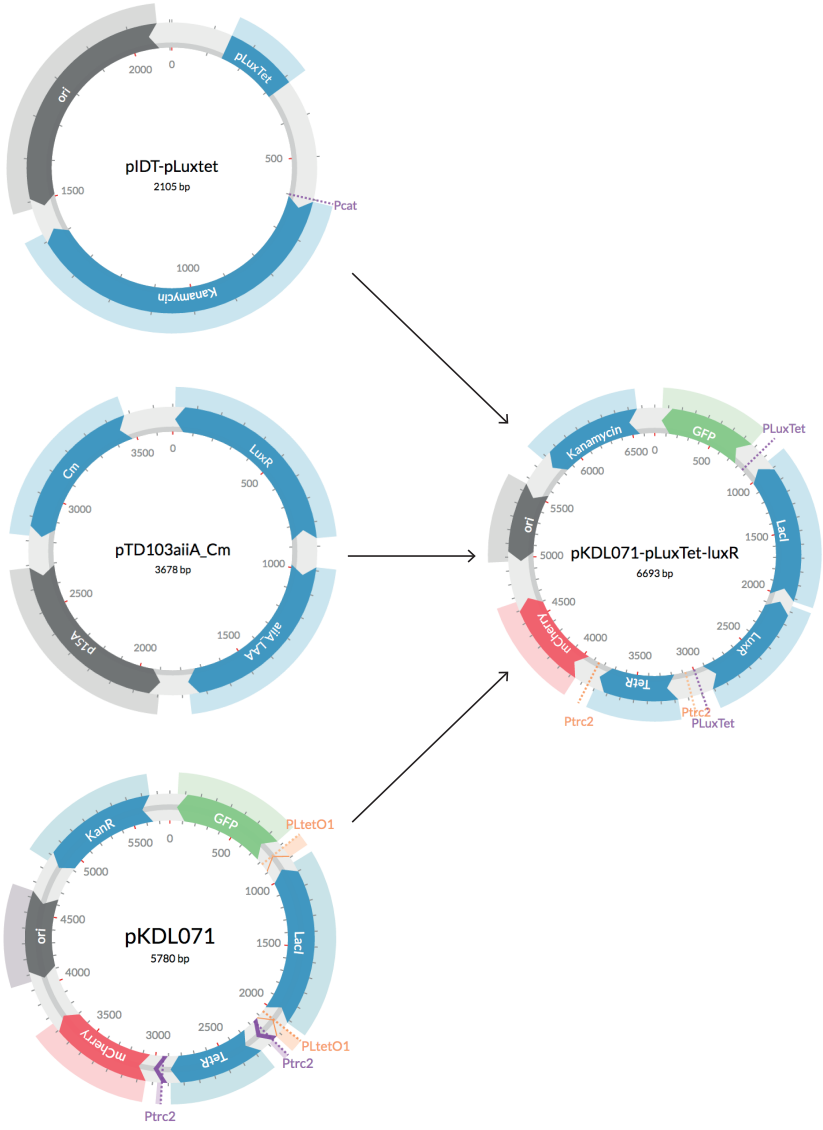

Figure 6.4 Stage 2 cloning procedure. The pKDL071-pluxtet-luxR plasmid is constructed via PCR cloning from the synthesised $P_{Lux/tet}$ promoter, pKDL071 and pTD103aiiA(Cm) plasmids.

6.3.3 Stage 3 - Construction of pKDL0713a

The final construction stage requires the complete pKDL071-plac/ara-araC plasmid, as well as the synthesised $P_{Lux/tet}$ promoter and pTD103aiiA(Cm) plasmid used in Stage 2. The plux/tet plasmid and pKDL071-plac/ara-araC will be digested with SphI and AclI restriction enzymes. This will be followed by gel extraction to isolate the fragments of interest. These will then be ligated to result in the modified pKDL071-plac/ara-araC plasmid, (pKDL071-plac/ara-araC-pluxtetA) with $P_{Lux/tet}$upstream of GFP instead of $P_{LtetO-1}$.

Then, the plux/tet plasmid and the plasmid created above (pKDL071-plac/ara-araC-pluxtetA) will be digested with EcoO109I and EagI. Following gel extraction and ligation, the $P_{Lux/tet}$ promoter will be added upstream of LacI instead of $P_{LtetO-1}$ to make a new plasmid, pKDL071-plac/ara-araC-pluxtet. Subsequently, the PCR product produced above of pTD103aiiA_Cm with BsGI flanking sequences and pKDL071-plac/ara-araC-pluxtet will be digested using BsGI. The fragments of interest will then be extracted following gel electrophoresis of the digested products and ligated. The ligates should be screened for the correct orientation of the insert.

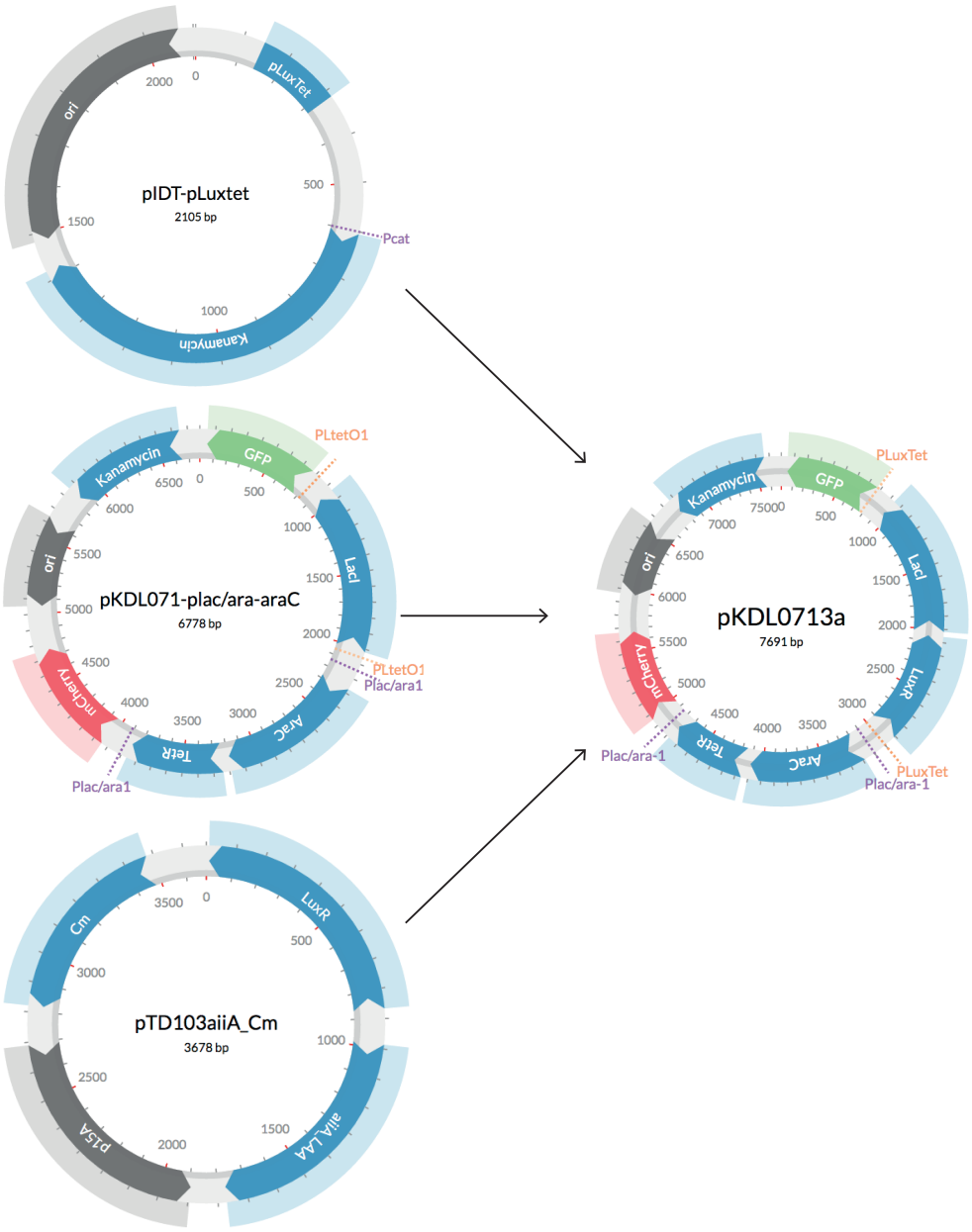

Figure 6.5 Stage 3 cloning procedure. The pKDL0713a plasmid is constructed via PCR cloning from the synthesised $P_{Lux/tet}$ promoter, pKDL071-plac/ara-araC and pTD103aiiA(Cm) plasmids.

6.4 Discussion

The switches described in this Chapter will enable the testing of the predictions on system robustness made in this thesis. Once the new toggle switches are constructed, they will be able to be characterised using ABC-Flow. This will allow the inference of the parameter values in the models describing these systems and ultimately a better understanding of the effect that adding positive autoregulation has to the switch.

I started implementing the experimental plan outlined above. In Stage 1 all the PCRs and digestions described were completed successfully (data not shown). Transformation of the ligated products into thermocompetent *E.coli* was carried out, but the transformation was not successful. In Stage 2 the $P_{Lux/tet}$ promoter was synthesised. Synthesis was carried out by Integrated DNA Technologies, Inc. (Leuven, Belgium, http://eu.idtdna.com/CodonOpt). *E.coli* DH5α was transformed with the synthesised plasmid. The method is outlined in Section 5.7.2.9. All subsequent PCRs and digestions described in Section 6.3.2 were carried out successfully. Due to time constraints, the rest of the experimental plan was not carried out.

6.5 Summary

In this Chapter I designed the experimental protocol to be followed in order to construct three novel switches. These switches can be used in the future in synthetic biology applications. The execution of the experimental protocol described has not been completed to date but constitutes the future directions of this project.

7 Conclusions

Synthetic biology aims at using engineering principles for the construction of new biological systems. A parallel is drawn between the design and construction process in engineering and synthetic biology. Emphasis is put on modularity and standardisation of the parts in play as well as on the separation of design and construction (Agapakis & Silver 2009). Ideally, synthetic biology would have a toolkit of interchangeable parts that can be chosen for each application. Like nuts and bolts whose functions and features are well known and characterised, the synthetic biologist aims to have an equivalent toolkit of fully characterised promoters and genes that can be selected to produce the system of choice. Nevertheless, biology does not conform with this idealised scenario. The cell is a noisy environment with a large number of unknowns and biological parts exhibit crosstalk, making the system unpredictable. This only highlights the need for the use of better computational tools for the understanding of a given biological system. Better tools are needed not only for the design of new synthetic systems but also for the better understanding of existing systems.

Here I have addressed both of these issues by applying Bayesian statistics to synthetic biology problems. An existing package for Bayesian model selection was used, as well as two new computational tools developed. The first tool that was constructed, StabilityFinder, is used for the design of synthetic systems and the second, ABC-Flow, is used for the inference of the parameters of existing biological systems. I applied the above tools to the understanding of a commonly found motif, the genetic toggle switch. The genetic toggle switch is an essential part of the synthetic biology toolkit and understanding its features is of great importance for the success of future applications of this motif. The methods developed for this simple system can be expanded to larger and more complex systems.

The first synthetic system design problem that was approached here was the design of a robust toggle switch. Bayesian model selection was used to determine that the addition of feedback loops to the classic design of the toggle switch can

increase its parametric robustness and improve the system's ability to realise the set of predefined design objectives. This finding can be used for the construction of a reliable synthetic toggle switch, and aid in moving this motif from the lab onto a real word application.

The first tool developed here, StabilityFinder, uses Bayesian statistics to identify the parameter values that give rise to the desired stability for a given model and can be used to design novel synthetic switches. StabilityFinder was used to gain further insights into the stability the genetic toggle switch is capable of. It was shown that the genetic toggle switch is capable of multistable behaviour, and the design principles behind each behaviour were uncovered. This insight can be used to construct new synthetic switches that can behave in the desired way. The successful construction of switches with more than two possible steady states can extend the number of applications the switch can be used for. StabilityFinder was also used to study the genetic toggle switch under different modelling abstractions. It was shown that the QSSA cannot always be justified in the study of system behaviour. More general mass action models were used to study the switch model using deterministic and stochastic dynamics and it was found that it is capable of multistationarity.

Robustness of the bistable toggle switch was examined using StabilityFinder and it was found that the addition of double positive feedback loops increases the parametric robustness of the system. This result complements the first conclusion drawn in this thesis. Both methods look at the ability of this model to behave like a switch, but define the behaviour in different ways. Using ABC-SysBio a switch-like behaviour was defined as three design objectives that needed to be fulfilled, with a predefined time to reach each state and level of protein. The models were ranked for their ability to fit this very specific behaviour. StabilityFinder defined a switch-like behaviour as two clusters of steady state values of the two proteins in the system within a predefined time frame. It does not automatically rank the models under consideration, and the robustness analysis is applied after-the-fact. The two methods agree that the addition of positive feedback loops increases the parametric robustness of the switch, thus strengthening the argument.

The toggle switch was also studied experimentally. The sensitivity of the switch to both inducers was investigated. The switch was also observed switching states over time for both sides of the switch. The problem of parameter inference for flow cytometry data was addressed by developing ABC-Flow. ABC-Flow is used to fit stochastic computational models to data obtained from flow cytometry. It was shown that it can be used to infer the parameter values that give rise to the observed

experimental data collected here. A computational model of the toggle switch was fit to one and two dimensional data. This enables the parameterisation of quantitative models using 2D flow cytometry data. This can provide further insights into the system under study that could not be otherwise obtained. The behavioural properties owed to different sub-parts can be untangled and further our understanding of the underlying effects at play in a given system.

The next step will be to move towards the testing of the predictions made here experimentally. The realisation of the switches with added positive feedback - using the construction strategy described here - would enable the testing of system robustness. Further, using the design principles of multistable switches predicted here, a switch with three or four states could be constructed in the lab. This will expand the toolkit of modules that can be used for synthetic biology applications.

Another important step from here would be to move towards the integration of multiple devices. For synthetic biology to move into real clinical and industrial applications, the systems we can design and build will have to become more complex and reliable. This will require the successful interplay between multiple devices like the one studied here. In the future multiple switch modules can be combined to create more complex system behaviours. Switches can be combined to work in tandem with other kinds of modules like actuators and oscillators to perform complex functions in the cell. This will not be a trivial process due to retroactivity and crosstalk between devices (Del Vecchio, Ninfa, & Sontag 2008) and thus further testing will be required.

In this thesis I studied the genetic toggle switch computationally and experimentally. I developed two computational tools that can be used for the study of genetic systems in systems and synthetic biology. I used them to uncover important aspects of the toggle switch system, a known regulatory motif in natural and synthetic systems. The work presented here advances our understanding of the design of novel switches as well as of an existing synthetic genetic toggle switch. These approaches are a necessary first step in transforming synthetic biology into a a true engineering discipline.

Bibliography

Ackers, G. K., Johnson, A. D., & Shea, M. A. (1982). 'Quantitative model for gene regulation by lambda phage repressor.' *Proceedings of the National Academy of Sciences of the United States of America* **79**(4), 1129–1133.

Agapakis, C. M. & Silver, P. A. (2009). 'Synthetic biology: exploring and exploiting genetic modularity through the design of novel biological networks'. *Molecular BioSystems* **5**(7), 704.

Alon, U. (2007). *An Introduction To The Systems Biology*. Chapman & Hall/CRC.

Andersen, J. B., Sternberg, C., Poulsen, L. K., Bjorn, S. P., Givskov, M., & Molin, S. (1998). 'New unstable variants of green fluorescent protein for studies of transient gene expression in bacteria.' *Applied and Environmental Microbiology* **64**(6), 2240–2246.

Andrianantoandro, E., Basu, S., Karig, D. K., & Weiss, R. (2006). 'Synthetic biology: new engineering rules for an emerging discipline.' *Molecular systems biology* **2**(1), 1–14.

Atkinson, M., Savageau, M., Myers, J., & Ninfa, A. J. (2003). 'Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in Escherichia coli'. *Cell* **113**(5), 597–607.

Attune NxT Acoustic Focusing Cytometer (2015). CO016625.

Babtie, A. C., Kirk, P., & Stumpf, M. P. H. (2014). 'Topological sensitivity analysis for systems biology.' *Proceedings of the National Academy of Sciences of the United States of America* **111**(52), 18507–18512.

Banaji, M. & Craciun, G. (2010). 'Graph-theoretic criteria for injectivity and unique equilibria in general chemical reaction systems'. *Advances in Applied Mathematics* **44**(2), 168–184.

Barkai, N. & Leibler, S. (1997). 'Robustness in simple biochemical networks.' *Nature* **387**(6636), 913–917.

Barnes, C. P., Silk, D., Sheng, X., & Stumpf, M. P. H. (2011). 'Bayesian design of synthetic biological systems.' *Proceedings of the National Academy of Sciences of the United States of America* **108**(37), 15190–15195.

Barnes, C. P., Silk, D., & Stumpf, M. P. H. (2011). 'Bayesian design strategies for synthetic biology.' *Interface Focus* **1**(6), 895–908.

Basu, S., Mehreja, R., Thiberge, S., Chen, M.-T., & Weiss, R. (2004). 'Spatiotemporal control of gene expression with pulse-generating networks.' *Proceedings of the National Academy of Sciences of the United States of America* **101**(17), 6355–6360.

Batt, G., Yordanov, B., Weiss, R., & Belta, C. (2007). 'Robustness analysis and tuning of synthetic gene networks.' *Bioinformatics (Oxford, England)* **23**(18), 2415–2422.

Beal, J., Haddock-Angelli, T., Gershater, M., de Mora, K., Lizarazo, M., Hollenhorst, J., Rettberg, R., & iGEM Interlab Study Contributors (2016). 'Reproducibility of Fluorescent Expression from Engineered Biological Constructs in E. coli.' *PLoS ONE* **11**(3), e0150182.

Biancalani, T. & Assaf, M. (2015). 'Noise Can Induce Bimodality in Positive Transcriptional Feedback Loops Without Bistability'. *Physical review letters* **327**(5969), 1142–1145.

Bower, A. G., McClintock, M. K., & Fong, S. S. (2010). 'Synthetic biology: a foundation for multi-scale molecular biology.' *Bioengineered Bugs* **1**(5), 309–312.

Brandman, O., Ferrell, J. E., Li, R., & Meyer, T. (2005). 'Interlinked fast and slow positive feedback loops drive reliable cell decisions.' *Science* **310**(5747), 496–498.

Chen, B.-S., Chang, C.-H., & Lee, H.-C. (2009). 'Robust synthetic biology design: stochastic game theory approach.' *Bioinformatics (Oxford, England)* **25**(14), 1822–1830.

Cherry, J. L. & Adler, F. R. (2000). 'How to make a biological switch.' *Journal of Theoretical Biology* **203**(2), 117–133.

Choi, S.-L., Rha, E., Lee, S. J., Kim, H., Kwon, K., Jeong, Y.-S., Rhee, Y. H., Song, J. J., Kim, H.-S., & Lee, S.-G. (2014). 'Toward a generalized and high-throughput enzyme screening system based on artificial genetic circuits.' *ACS Synthetic Biology* **3**(3), 163–171.

Cinquin, O. & Demongeot, J. (2005). 'High-dimensional switches and the modelling of cellular differentiation'. *Journal of Theoretical Biology* **233**(3), 391–411.

Clewley, R. (2012). 'Hybrid models and biological model reduction with PyDSTool.' *PLoS Computational Biology* **8**(8), e1002628–e1002628.

Conradi, C., Flockerzi, D., Raisch, J., & Stelling, J. (2007). 'Subnetwork analysis reveals dynamic features of complex (bio)chemical networks'. *PNAS* **104**(49), 19175–19180.

Cooling, M. T., Rouilly, V., Misirli, G., Lawson, J., Yu, T., Hallinan, J., & Wipat, A. (2010). 'Standard virtual biological parts: a repository of modular modeling components for synthetic biology.' *Bioinformatics (Oxford, England)* **26**(7), 925–931.

Cormack, B. P., Valdivia, R. H., & Falkow, S. (1996). 'FACS-optimized mutants of the green fluorescent protein (GFP)'. *Gene* **173**(1), 33–38.

De Jong, H. (2002). 'Modeling and simulation of genetic regulatory systems: a literature review.' *Journal of Computational Biology* **9**(1), 67–103.

Deans, T. L., Cantor, C. R., & Collins, J. J. (2007). 'A Tunable Genetic Switch Based on RNAi and Repressor Proteins for Regulating Gene Expression in Mammalian Cells'. *Cell* **130**(2), 363–372.

Del Vecchio, D., Ninfa, A. J., & Sontag, E. D. (2008). 'Modular cell biology: retroactivity and insulation.' *Molecular systems biology* **4**, 161–161.

Díaz, M., Herrero, M., García, L. A., & Quirós, C. (2010). 'Application of flow cytometry to industrial microbial bioprocesses'. *Biochemical Engineering Journal* **48**(3), 385–407.

DuCharme, B. (1999). *XML : the annotated specification / Bob DuCharme*. English. Prentice Hall PTR Upper Saddle River, NJ, xix, 339 p. : ISBN: 0130826766.

Eldar, A. & Elowitz, M. B. (2010). 'Functional roles for noise in genetic circuits'. *Nature* **467**(7312), 167–173.

Ellis, B., Gentleman, R., Hahne, F., Le Meur, N., Sarkar, D., & Jiang, M. (2016a). *flowViz: Visualization for flow cytometry*. R package version 1.36.2.

Ellis, B., Haaland, P., Hahne, F., Le Meur, N., Gopalakrishnan, N., Spidlen, J., & Jiang, M. (2016b). *flowCore: flowCore: Basic structures for flow cytometry data*. R package version 1.38.2.

Ellis, T., Wang, X., & Collins, J. J. (2009). 'Diversity-based, model-guided construction of synthetic gene networks with predicted functions.' *Nature Biotechnology* **27**(5), 465–471.

Elowitz, M. B. (2002). 'Stochastic Gene Expression in a Single Cell'. *Science* **297**(5584), 1183–1186.

Entus, R., Aufderheide, B., & Sauro, H. M. (2007). 'Design and implementation of three incoherent feed-forward motif based biological concentration sensors.' *Systems and synthetic biology* **1**(3), 119–128.

Fasano, G. & Franceschini, A. (1987). 'A multidimensional version of the Kolmogorov-Smirnov test'. *Monthly Notices of the Royal Astronomical Society* **225**(1), 155–170.

Fedorec, A. J. (2016). *autoGate*. https://github.com/ajfedorec/autoGate.git.

Feliu, E. & Wiuf, C. (2013). 'A computational method to preclude multistationarity in networks of interacting species.' *Bioinformatics (Oxford, England)* **29**(18), 2327–2334.

Ferrell Jr, J. E. (2002). 'Self-perpetuating states in signal transduction: positive feedback, double-negative feedback and bistability'. *Current opinion in cell biology* **14**(2), 140–148.

Friedland, A. E., Lu, T. K., Wang, X., Shi, D., Church, G., & Collins, J. J. (2009). 'Synthetic gene networks that count.' *Science* **324**(5931), 1199–1202.

Friedman, J. H. & Rafsky, L. C. (1979). 'Multivariate generalizations of the Wald-Wolfowitz and Smirnov two-sample tests'. *The Annals of Statistics* **7**, 697–717.

Fukunaga, K. (2013). *Introduction to Statistical Pattern Recognition*. Academic Press.

Fung, E., Wong, W. W., Suen, J. K., Bulter, T., Lee, S. G., & Liao, J. C. (2005). 'A synthetic gene–metabolic oscillator '. *Nature* **435**(7038), 118–122.

Gardner, T. S., Cantor, C. R., & Collins, J. J. (2000). 'Construction of a genetic toggle switch in Escherichia coli'. *Nature* **403**(6767), 339–342.

Ghaffarizadeh, A., Flann, N. S., & Podgorski, G. J. (2014). 'Multistable switches and their role in cellular differentiation networks.' *BMC Bioinformatics* **15**(Suppl 7), S7.

Gilks, W. R., Richardson, S., & Spiegelhalter, D. J. (1996). *Markov Chain Monte Carlo in Practice*. London: Chapman and Hall.

Gillespie, D. T. (1977). 'Exact Stochastic Simulation of Coupled Chemical-Reactions'. *Journal of Physical Chemistry* **81**(25), 2340–2361.

Gramelsberger, G. (2013). 'The simulation approach in synthetic biology.' *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences* **44**(2), 150–157.

Guantes, R. & Poyatos, J. F. (2008). 'Multistable decision switches for flexible control of epigenetic differentiation.' *PLoS Computational Biology* **4**(11), e1000235.

Hafner, M., Koeppl, H., Hasler, M., & Wagner, A. (2009). ''Glocal' Robustness Analysis and Model Discrimination for Circadian Oscillators'. *PLoS Computational Biology* **5**(10), e1000534.

Ham, T. S., Lee, S. K., Keasling, J. D., & Arkin, A. P. (2008). 'Design and Construction of a Double Inversion Recombination Switch for Heritable Sequential Genetic Memory'. *PLoS ONE* **3**(7), e2815.

Heinemann, M. & Panke, S. (2006). 'Synthetic biology–putting engineering into biology'. *Bioinformatics (Oxford, England)* **22**(22), 2790–2799.

Hill, A. V. (1910). 'The possible effects of the aggregation of the molecules of haemoglobin on its dissociation curves'. *Proceedings of the Physiological society*, 4–7.

Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., & Kummer, U. (2006). 'COPASI–a COmplex PAthway SImulator.' *Bioinformatics (Oxford, England)* **22**(24), 3067–3074.

Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., Arkin, A. P., Bornstein, B. J., Bray, D., Cornish-Bowden, A., Cuellar, A. A., Dronov, S., Gilles, E. D., Ginkel, M., Gor, V., Goryanin, I. I., Hedley, W. J., Hodgman, T. C., Hofmeyr, J.-H., Hunter, P. J., Juty, N. S., Kasberger, J. L., Kremling, A., Kummer, U., Le Novere, N., Loew, L. M., Lucio, D., Mendes, P., Minch, E., Mjolsness, E. D., Nakayama, Y., Nelson, M. R., Nielsen, P. F., Sakurada, T., Schaff, J. C., Shapiro, B. E., Shimizu, T. S., Spence, H. D., Stelling, J., Takahashi, K., Tomita, M., Wagner, J., & Wang, J. (2003). 'The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models.' *Bioinformatics (Oxford, England)* **19**(4), 524–531.

Hutchison, C. A., Chuang, R.-Y., Noskov, V. N., Assad-Garcia, N., Deerinck, T. J., Ellisman, M. H., Gill, J., Kannan, K., Karas, B. J., Ma, L., Pelletier, J. F., Qi, Z.-Q., Richter, R. A., Strychalski, E. A., Sun, L., Suzuki, Y., Tsvetanova, B., Wise, K. S., Smith, H. O., Glass, J. I., Merryman, C., Gibson, D. G., & Venter, J. C. (2016). 'Design and synthesis of a minimal bacterial genome.' *Science* **351**(6280), aad6253–aad6253.

Ingalls, B. & Iglesias, P. (2010). 'A primer on control engineering'. In: *Control theory and systems biology*. Ed. by B. Ingalls & P. Iglesias. Cambridge, MA: MIT Press. Chap. 1, 1–28.

Isaacs, F. J., Hasty, J., Cantor, C. R., & Collins, J. J. (2003). 'Prediction and measurement of an autoregulatory genetic module.' *Proceedings of the National Academy of Sciences of the United States of America* **100**(13), 7714–7719.

Jiang, P., Ventura, A. C., Sontag, E. D., Merajver, S. D., Ninfa, A. J., & Del Vecchio, D. (2011). 'Load-induced modulation of signal transduction networks.' *Science Signaling* **4**(194), ra67.

Jones, E., Oliphant, T., Peterson, P. et al. (2001). *SciPy: Open source scientific tools for Python*. [Online; accessed 2016-09-02]. URL: http://www.scipy.org/.

Kaplan, D. & Glass, L. (1995). *Understanding nonlinear dynamics*. Springer-Verlag.

Kass, R. E. & Raftery, A. E. (1995). 'Bayes Factors'. *Journal of the American Statistical Association* **90**, 773–795.

Kelly, J. R., Rubin, A. J., Davis, J. H., Ajo-Franklin, C. M., Cumbers, J., Czar, M. J., de Mora, K., Glieberman, A. L., Monie, D. D., & Endy, D. (2009). 'Measuring the activity of BioBrick promoters using an in vivo reference standard.' *Journal of Biological Engineering* **3**(1), 4–4.

Kelwick, R., MacDonald, J. T., Webb, A. J., & Freemont, P. (2014). 'Developments in the tools and methodologies of synthetic biology.' *Frontiers in bioengineering and biotechnology* **2**, 60–60.

Khalil, A. S. & Collins, J. J. (2010). 'Synthetic biology: applications come of age'. *Nature Publishing Group* **11**(5), 367–379.

Khammash, M. (2010). 'Modeling and analysis of stochastic biochemical networks'. In: *Control theory and systems biology*. Ed. by B. Ingalls & P. Iglesias. Cambridge, MA: MIT Press. Chap. 2, 29–44.

Kim, J., Bates, D. G., Postlewaite, I., Ma, L., & Iglesias, P. A. (2006). 'Robustness analysis of biochemical network models'. *Systems biology*.

Kirk, D. B. & Hwu, W.-m. W. (2010). *Programming Massively Parallel Processors*. A Hands-on Approach. Burlington: Morgan Kaufmann.

Kirkpatrick, S., C D Gelatt, J., & Vecchi, M. P. (1983). 'Optimization by Simulated Annealing'. *Science* **220**(4598), 671–680.

Kitano, H. (2007). 'Towards a theory of biological robustness'. *Molecular systems biology* **3**(1).

Kobayashi, H., Kaern, M., Araki, M., Chung, K., Gardner, T. S., Cantor, C. R., & Collins, J. J. (2004). 'Programmable cells: interfacing natural and engineered gene networks.' *Proceedings of the National Academy of Sciences of the United States of America* **101**(22), 8414–8419.

Kolmogorov, A. N. (1933). 'Sulla Determinazione Empirica di Una Legge di Distribuzione'. *Giornale dell'Istituto Italiano degli Attuari* **4**, 83–91.

Konopka, A. (2007). *Systems Biology, Principles, methods and concepts*. CRC Press.

Kramer, B. P., Viretta, A. U., Daoud-El-Baba, M., Aubel, D., Weber, W., & Fussenegger, M. (2004). 'An engineered epigenetic transgene switch in mammalian cells.' *Nature Biotechnology* **22**(7), 867–870.

Le Novère, N. (2015). 'Quantitative and logic modelling of molecular and gene networks.' *Nature Reviews Genetics* **16**(3), 146–158.

Le Novère, N., Bornstein, B., Broicher, A., Courtot, M., Donizelli, M., Dharuri, H., Li, L., Sauro, H., Schilstra, M., Shapiro, B., Snoep, J. L., & Hucka, M. (2006). 'BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems.' *Nucleic Acids Research* **34**(Database issue), D689–D691.

Levenberg, K. (1944). 'A method for the solution of certain non-linear problems in least squares'. *Quarterly of applied mathematics* **2**, 164–168.

Li, P., Dada, J. O., Jameson, D., Spasic, I., Swainston, N., Carroll, K., Dunn, W., Khan, F., Malys, N., Messiha, H. L., Simeonidis, E., Weichart, D., Winder, C., Wishart, J., Broomhead, D. S., Goble, C. A., Gaskell, S. J., Kell, D. B., Westerhoff, H. V., Mendes, P., & Paton, N. W. (2010). 'Systematic integration of experimental data and models in systems biology.' *BMC Bioinformatics* **11**, 582–582.

Liepe, J., Barnes, C., Cule, E., Erguler, K., Kirk, P., Toni, T., & Stumpf, M. P. H. (2010). 'ABC-SysBio–approximate Bayesian computation in Python with GPU support.' *Bioinformatics (Oxford, England)* **26**(14), 1797–1799.

Liepe, J., Kirk, P., Filippi, S., Toni, T., Barnes, C. P., & Stumpf, M. P. H. (2014). 'A framework for parameter estimation and model selection from experimental data in systems biology using approximate Bayesian computation.' *Nature Protocols* **9**(2), 439–456.

Liew, C. W., Rand, K. D., Simpson, R. J. Y., Yung, W. W., Mansfield, R. E., Crossley, M., Proetorius-Ibba, M., Nerlov, C., Poulsen, F. M., & Mackay, J. P. (2006). 'Molecular analysis of the interaction between the hematopoietic master transcription factors GATA-1 and PU.1.' *Journal of Biological Chemistry* **281**(38), 28296–28306.

Lillacci, G. & Khammash, M. (2013). 'The signal within the noise: efficient inference of stochastic gene regulation models using fluorescence histograms and stochastic simulations.' *Bioinformatics (Oxford, England)* **29**(18), 2311–2319.

Lipshtat, A., Loinger, A., Balaban, N. Q., & Biham, O. (2006). 'Genetic toggle switch without cooperative binding.' *Physical review letters* **96**(18), 188101.

Litcofsky, K. D., Afeyan, R. B., Krom, R. J., Khalil, A. S., & Collins, J. J. (2012). 'Iterative plug-and-play methodology for constructing and modifying synthetic gene networks.' *Nature Methods* **9**(11), 1077–1080.

Lloyd, S. P. (1982). 'Least squares quantization in PCM'. *IEEE Transactions on Information Theory* **IT-28**, 129–137.

Loinger, A., Lipshtat, A., Balaban, N. Q., & Biham, O. (2007). 'Stochastic simulations of genetic switch systems'. *Physical Review E*.

Loinger, A. & Biham, O. (2009). 'Analysis of genetic toggle switch systems encoded on plasmids.' *Physical review letters* **103**(6), 068104.

Lopes, R. H. C., Reid, I., & Hobson, P. R. (2007). 'The two-dimensional Kolmogorov-Smirnov test'. In: *International Workshop on Advanced Computing and Analysis Techniques in Physics Research*. Amsterdam, 1–12.

Lotka, A. J. (1925). *Elements of Physical Biology*. Williams and Wilkins.

Lu, M., Onuchic, J., & Ben-Jacob, E. (2014). 'Construction of an Effective Landscape for Multistate Genetic Switches'. *Physical review letters* **113**(7), 078102.

Lu, T. K., Khalil, A. S., & Collins, J. J. (2009). 'Next-generation synthetic gene networks'. *Nature Biotechnology* **27**(12), 1139–1150.

Lutz, R. & Bujard, H. (1997). 'Independent and tight regulation of transcriptional units in Escherichia coli via the LacR/O, the TetR/O and AraC/I1-I2 regulatory elements.' *Nucleic Acids Research* **25**(6), 1203–1210.

Lyons, S. M., Xu, W., Medford, J., & Prasad, A. (2014). 'Loads bias genetic and signaling switches in synthetic and natural systems.' *PLoS Computational Biology* **10**(3), e1003533–e1003533.

Ma, R., Wang, J., Hou, Z., & Liu, H. (2012). 'Small-number effects: a third stable state in a genetic bistable toggle switch'. *Physical review letters* **109**(24), 248107.

Ma, W., Trusina, A., El-Samad, H., Lim, W. A., & Tang, C. (2009). 'Defining network topologies that can achieve biochemical adaptation.' *Cell* **138**(4), 760–773.

Macdonald, P. J., Chen, Y., & Mueller, J. D. (2012). 'Chromophore maturation and fluorescence fluctuation spectroscopy of fluorescent proteins in a cell-free expression system.' *Analytical Biochemistry* **421**(1), 291–298.

Major, S. (2016). *ndtest*. https://github.com/syrte/ndtest.git.

Marjoram, P., Molitor, J., Plagnol, V., & Tavare, S. (2003). 'Markov chain Monte Carlo without likelihoods.' *Proceedings of the National Academy of Sciences of the United States of America* **100**(26), 15324–15328.

Marquardt, D. W. (1963). 'An algorithm for least-squares estimation of nonlinear parameters'. *Journal of the society for Industrial and Applied Mathematics* **11**, 431–441.

Mathematica (2016). *version 10.3*. Wolfram Research, Inc.

McKay, M. D., Beckman, R. J., & Conover, W. J. (2000). 'A comparison of three methods for selecting values of input variables in the analysis of output from a computer code'. *Technometrics* **42**(1), 55–61.

Milo, R., Jorgensen, P., Moran, U., Weber, G., & Springer, M. (2010). 'BioNumbers–the database of key numbers in molecular and cell biology.' *Nucleic Acids Research* **38**(Database issue), D750–D753.

Monaco, J. V. (2014). 'Classification and Authentication of One-dimensional Behavioral Biometrics'. In: *International Journal of Cognitive Biometrics*, 1–8.

Müller, K. M. & Arndt, K. M. (2011). 'Standardization in Synthetic Biology'. *Methods in molecular biology* **813**, 23–43.

Nielsen, A. A., Segall-Shapiro, T. H., & Voigt, C. A. (2013). 'Advances in genetic circuit design: novel biochemistries, deep part mining, and precision gene expression'. *Current Opinion in Chemical Biology* **17**(6), 878–892.

Niwa, H., Toyooka, Y., Shimosato, D., Strumpf, D., Takahashi, K., Yagi, R., & Rossant, J. (2005). 'Interaction between Oct3/4 and Cdx2 Determines Trophectoderm Differentiation'. *Cell* **123**(5), 13–13.

Onbaşoğlu, E. & Özdamar, L. (2001). 'Parallel Simulated Annealing Algorithms in Global Optimization'. *Journal of Global Optimization* **19**(1), 27–50.

Pedersen, M. G., Bersani, A. M., & Bersani, E. (2007). 'Quasi steady-state approximations in complex intracellular signal transduction networks – a word of caution'. *Journal of Mathematical Chemistry* **43**(4), 1318–1344.

Phillips, R., Kondev, J., Theriot, J., & Garcia, H. G. (2013). *Physical biology of the cell*. Garland Science.

Prill, R. J., Iglesias, P. A., & Levchenko, A. (2005). 'Dynamic properties of network motifs contribute to biological network organization.' *PLoS Biology* **3**(11), e343–e343.

Prindle, A., Samayoa, P., Razinkov, I., Danino, T., Tsimring, L. S., & Hasty, J. (2012). 'A sensing array of radically coupled genetic 'biopixels'.' *Nature* **481**(7379), 39–44.

Pritchard, J. K., Seielstad, M. T., Perez-Lezaun, A., & Feldman, M. W. (1999). 'Population growth of human Y chromosomes: A study of Y chromosome microsatellites'. *Molecular Biology and Evolution* **16**(12), 1791–1798.

Ptashne, M. (1992). *A Genetic Switch: Phage lambda and Higher Organisms*. Cell Press and Blackwell Scientific.

R Core Team (2008). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. ISBN: 3-900051-07-0. URL: www.Rproject.org.

Ro, D.-K., Paradise, E. M., Ouellet, M., Fisher, K. J., Newman, K. L., Ndungu, J. M., Ho, K. A., Eachus, R. A., Ham, T. S., Kirby, J., Chang, M. C. Y., Withers, S. T., Shiba, Y., Sarpong, R., & Keasling, J. D. (2006). 'Production of the antimalarial drug precursor artemisinic acid in engineered yeast'. *Nature* **440**(7086), 940–943.

Rosenfeld, N., Perkins, T. J., Alon, U., Elowitz, M. B., & Swain, P. S. (2006). 'A fluctuation method to quantify in vivo fluorescence data.' *Biophysj* **91**(2), 759–766.

Salis, H. M., Mirsky, E. A., & Voigt, C. A. (2009). 'Automated design of synthetic ribosome binding sites to control protein expression.' *Nature Biotechnology* **27**(10), 946–950.

Shaner, N. C., Campbell, R. E., Steinbach, P. A., Giepmans, B. N. G., Palmer, A. E., & Tsien, R. Y. (2004). 'Improved monomeric red, orange and yellow fluorescent proteins derived from Discosoma sp. red fluorescent protein.' *Nature Biotechnology* **22**(12), 1567–1572.

Shapiro, H. M. (1941). *Practical flow cytometry*. Wiley-Liss.

Shimomura, O., Johnson, F. H., & Saiga, Y. (1962). 'Extraction, purification and properties of aequorin, a bioluminescent protein from the luminous hydromedusan, Aequorea.' *Journal of Cellular and Comparative Physiology* **59**, 223–239.

Shinar, G. & Feinberg, M. (2010). 'Structural Sources of Robustness in Biochemical Reaction Networks'. *Science* **327**(5971), 1389–1391.

Silk, D., Filippi, S., & Stumpf, M. P. H. (2013). 'Optimizing threshold-schedules for sequential approximate Bayesian computation: applications to molecular systems.' *Statistical Applications in Genetics and Molecular Biology* **12**(5), 603–618.

Sisson, S. A., Fan, Y., & Tanaka, M. M. (2007). 'Sequential Monte Carlo without likelihoods'. *Proceedings of the National Academy of Sciences of the United States of America* **104**(6), 1760–1765.

Stelling, J., Sauer, U., Szallasi, Z., Doyle, F. J., & Doyle, J. (2004). 'Robustness of cellular functions'. *Cell* **118**(6), 675–685.

Strasser, M., Theis, F. J., & Marr, C. (2012). 'Stability and multiattractor dynamics of a toggle switch based on a two-stage model of stochastic gene expression.' *Biophysical Journal* **102**(1), 19–29.

Stricker, J., Cookson, S., Bennett, M. R., Mather, W. H., Tsimring, L. S., & Hasty, J. (2008). 'A fast, robust and tunable synthetic gene oscillator'. *Nature* **456**(7221), 516–519.

Strogatz, S. H. (1994). *Nonlinear dynamics and chaos*. Westview Press.

Tabor, J. J., Salis, H. M., Simpson, Z. B., Chevalier, A. A., Levskaya, A., Marcotte, E. M., Voigt, C. A., & Ellington, A. D. (2009). 'A Synthetic Genetic Edge Detection Program'. *Cell* **137**(7), 1272–1281.

Thomas, R., Thieffry, D., & Kaufman, M. (1995). 'Dynamical behaviour of biological regulatory networks–I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state.' *Bulletin of mathematical biology* **57**(2), 247–276.

Tibshirani, R., Walther, G., & Hastie, T. (2001). 'Estimating the number of clusters in a data set via the gap statistic'. *Journal of the Royal Statistical Society Series B-Statistical Methodology* **63**(2), 411–423.

Tigges, M., Marquez-Lago, T. T., Stelling, J., & Fussenegger, M. (2009). 'A tunable synthetic mammalian oscillator.' *Nature* **457**(7227), 309–312.

Toni, T. (2010). 'Approximate Bayesian computation for parameter inference and model selection in systems biology'. PhD thesis. Imperial College London.

Toni, T., Jovanovic, G., Huvet, M., Buck, M., & Stumpf, M. P. H. (2011). 'From qualitative data to quantitative models: analysis of the phage shock protein stress response in Escherichia coli.' *BMC systems biology* **5**, 69–69.

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., & Stumpf, M. P. H. (2009). 'Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems.' *Journal of the Royal Society, Interface / the Royal Society* **6**(31), 187–202.

Veening, J.-W., Smits, W. K., & Kuipers, O. P. (2008). 'Bistability, epigenetics, and bet-hedging in bacteria.' *Microbiology* **62**, 193–210.

Ventura, A. C., Jiang, P., Van Wassenhove, L., Del Vecchio, D., Merajver, S. D., & Ninfa, A. J. (2010). 'Signaling properties of a covalent modification cycle are altered by a downstream target.' *PNAS* **107**(22), 10032–10037.

Vinson, V. & Pennisi, E. (2011). 'The Allure of Synthetic Biology'. *Science* **333**(6047), 1235–1235.

Walczak, A. M., Onuchic, J. N., & Wolynes, P. G. (2005). 'Absolute rate theories of epigenetic stability'. *Proceedings of the National Academy of Sciences of the United States of America* **102**(52), 18926–18931.

Wald, A. & Wolfowitz, J. (1940). 'On a test whether two samples are from the same population'. *The Annals of Mathematical Statistics* **11**, 147–162.

Wang, J. & Yu, H.-Q. (2007). 'Biosynthesis of polyhydroxybutyrate (PHB) and extracellular polymeric substances (EPS) by Ralstonia eutropha ATCC 17699 in batch cultures.' *Applied Microbiology and Biotechnology* **75**(4), 871–878.

Warren, P. B. & ten Wolde, P. R. (2004). 'Enhancement of the Stability of Genetic Switches by Overlapping Upstream Regulatory Domains'. *Physical review letters* **92**(12), 128101.

Warren, P. B. & ten Wolde, P. R. (2005). 'Chemical models of genetic toggle switches'. *The Journal of Physical Chemistry B* **109**(14), 6812–6823.

Weinberger, L. S., Burnett, J. C., Toettcher, J. E., Arkin, A. P., & Schaffer, D. V. (2005). 'Stochastic gene expression in a lentiviral positive-feedback loop: HIV-1 Tat fluctuations drive phenotypic diversity.' *Cell* **122**(2), 169–182.

What is Flow Cytometry. [Online; Accessed 2016-08-01]. URL: http://qcri.queensu.ca/cancer_biology_genetics/queens_university_biomedical_imaging_centre/services/flow_cytometry/what_is_flow_cytometry_.

Wickham, H. (2009). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN: 978-0-387-98140-6. URL: http://ggplot2.org.

Wilk, M. B. & Gnanadesikan, R. (1968). 'Probability plotting methods for the analysis of data.' *Biometrika* **55**(1), 1–17.

Wilkinson, D. J. (2006). *Stochastic Modelling for Systems Biology*. CRC Press.

Wood, G. R., Alexander, D., & Bulger, D. W. (2002). 'Approximation of the distribution of convergence times for stochastic global optimisation'. *Journal of Global Optimization*.

Woods, M. L., Leon, M., Perez-Carrasco, R., & Barnes, C. P. (2016). 'A Statistical Approach Reveals Designs for the Most Robust Stochastic Gene Oscillators.' *ACS Synthetic Biology* **5**(6), 459–470.

Wu, C.-H., Lee, H.-C., & Chen, B.-S. (2011). 'Robust synthetic gene network design via library-based search method'. *Bioinformatics (Oxford, England)* **27**(19), 2700–2706.

Youssef, S. A. M., Paik, J. K., Kim, Y. S., Kim, M. S., & Cheng, F. (2013). 'Probabilistic Selection of Ship-Ship Collision Scenarios'. In: *ASME 2013 32nd International Conference on Ocean, Offshore and Arctic Engineering*. ASME, V02AT02A027–V02AT02A027.

Zamora-Sillero, E., Hafner, M., Ibig, A., Stelling, J., & Wagner, A. (2011). 'Efficient characterization of high-dimensional parameter spaces for systems biology.' *BMC systems biology* **5**, 142.

Zheng, Y. & Sriram, G. (2010). 'Mathematical modeling: bridging the gap between concept and realization in synthetic biology.' *Journal of Biomedicine and Biotechnology* **2010**, 541609–541609.

Zhou, Y., Liepe, J., Sheng, X., Stumpf, M. P. H., & Barnes, C. (2011). 'GPU accelerated biochemical network simulation.' *Bioinformatics (Oxford, England)* **27**(6), 874–876.

A Biochemical kinetic models

A.1 Ordinary differential equations

A.1.1 Standard toggle switch with inducers

$$
\begin{aligned}
\frac{\mathrm{d}([\mathrm{A}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= +V_{\mathrm{cell}}\cdot(\mathrm{ge}\cdot[\mathrm{gA}]) - V_{\mathrm{cell}}\cdot(\mathrm{deg}\cdot[\mathrm{A}]) - 2\cdot V_{\mathrm{cell}}\cdot(\mathrm{dim}\cdot[\mathrm{A}]\cdot[\mathrm{A}]) \\
&\quad + 2\cdot V_{\mathrm{cell}}\cdot(\mathrm{dim\_r}\cdot[\mathrm{A2}]) \\
\frac{\mathrm{d}([\mathrm{gA}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= -V_{\mathrm{cell}}\cdot(\mathrm{rep}\cdot[\mathrm{gA}]\cdot[\mathrm{B2}]) + V_{\mathrm{cell}}\cdot(\mathrm{rep\_r}\cdot[\mathrm{B2gA}]) \\
\frac{\mathrm{d}([\mathrm{B}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= +V_{\mathrm{cell}}\cdot(\mathrm{ge}\cdot[\mathrm{gB}]) - V_{\mathrm{cell}}\cdot(\mathrm{deg}\cdot[\mathrm{B}]) - 2\cdot V_{\mathrm{cell}}\cdot(\mathrm{dim}\cdot[\mathrm{B}]\cdot[\mathrm{B}]) \\
&\quad + 2\cdot V_{\mathrm{cell}}\cdot(\mathrm{dim\_r}\cdot[\mathrm{B2}]) \\
\frac{\mathrm{d}([\mathrm{gB}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= -V_{\mathrm{cell}}\cdot(\mathrm{rep}\cdot[\mathrm{gB}]\cdot[\mathrm{A2}]) + V_{\mathrm{cell}}\cdot(\mathrm{rep\_r}\cdot[\mathrm{A2gB}]) \\
\frac{\mathrm{d}([\mathrm{A2}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= -V_{\mathrm{cell}}\cdot(\mathrm{rep\_dim}\cdot[\mathrm{S}]\cdot[\mathrm{A2}]) + V_{\mathrm{cell}}\cdot(\mathrm{rep\_dim\_r}\cdot[\mathrm{SA2}]) + V_{\mathrm{cell}}\cdot(\mathrm{dim}\cdot[\mathrm{A}]\cdot[\mathrm{A}]) \\
&\quad - V_{\mathrm{cell}}\cdot(\mathrm{dim\_r}\cdot[\mathrm{A2}]) - V_{\mathrm{cell}}\cdot(\mathrm{rep}\cdot[\mathrm{gB}]\cdot[\mathrm{A2}]) + V_{\mathrm{cell}}\cdot(\mathrm{rep\_r}\cdot[\mathrm{A2gB}]) \\
\frac{\mathrm{d}([\mathrm{B2}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= -V_{\mathrm{cell}}\cdot(\mathrm{rep\_dim}\cdot[\mathrm{R}]\cdot[\mathrm{B2}]) + V_{\mathrm{cell}}\cdot(\mathrm{rep\_dim\_r}\cdot[\mathrm{RB2}]) + V_{\mathrm{cell}}\cdot(\mathrm{dim}\cdot[\mathrm{B}]\cdot[\mathrm{B}]) \\
&\quad - V_{\mathrm{cell}}\cdot(\mathrm{dim\_r}\cdot[\mathrm{B2}]) - V_{\mathrm{cell}}\cdot(\mathrm{rep}\cdot[\mathrm{gA}]\cdot[\mathrm{B2}]) + V_{\mathrm{cell}}\cdot(\mathrm{rep\_r}\cdot[\mathrm{B2gA}]) \\
\frac{\mathrm{d}([\mathrm{A2gB}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= +V_{\mathrm{cell}}\cdot(\mathrm{rep}\cdot[\mathrm{gB}]\cdot[\mathrm{A2}]) - V_{\mathrm{cell}}\cdot(\mathrm{rep\_r}\cdot[\mathrm{A2gB}]) \\
\frac{\mathrm{d}([\mathrm{B2gA}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= +V_{\mathrm{cell}}\cdot(\mathrm{rep}\cdot[\mathrm{gA}]\cdot[\mathrm{B2}]) - V_{\mathrm{cell}}\cdot(\mathrm{rep\_r}\cdot[\mathrm{B2gA}])
\end{aligned}
$$

$$\frac{\mathrm{d}([\mathrm{S}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} = -V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim}\cdot[\mathrm{S}]\cdot[\mathrm{A2}]\right)+V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim\_r}\cdot[\mathrm{SA2}]\right)-V_{\mathrm{cell}}\cdot\left(\mathrm{deg\_sr}\cdot[\mathrm{S}]\right)$$

$$\frac{\mathrm{d}([\mathrm{SA2}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} = +V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim}\cdot[\mathrm{S}]\cdot[\mathrm{A2}]\right)-V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim\_r}\cdot[\mathrm{SA2}]\right)$$

$$\frac{\mathrm{d}([\mathrm{R}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} = -V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim}\cdot[\mathrm{R}]\cdot[\mathrm{B2}]\right)+V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim\_r}\cdot[\mathrm{RB2}]\right)-V_{\mathrm{cell}}\cdot\left(\mathrm{deg\_sr}\cdot[\mathrm{R}]\right)$$

$$\frac{\mathrm{d}([\mathrm{RB2}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} = +V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim}\cdot[\mathrm{R}]\cdot[\mathrm{B2}]\right)-V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim\_r}\cdot[\mathrm{RB2}]\right)$$

Positive autoregulation on B with inducers

$$\frac{\mathrm{d}([\mathrm{A}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} = +V_{\mathrm{cell}}\cdot(\mathrm{ge}\cdot[\mathrm{gA}]) - 2\cdot V_{\mathrm{cell}}\cdot(\mathrm{dim}\cdot[\mathrm{A}]\cdot[\mathrm{A}]) + 2\cdot V_{\mathrm{cell}}\cdot(\mathrm{dim\_r}\cdot[\mathrm{A2}]) - V_{\mathrm{cell}}\cdot(\mathrm{deg}\cdot[\mathrm{A}])$$

$$\frac{\mathrm{d}([\mathrm{gA}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} = -V_{\mathrm{cell}}\cdot(\mathrm{rep}\cdot[\mathrm{gA}]\cdot[\mathrm{B2}]) + V_{\mathrm{cell}}\cdot(\mathrm{rep\_r}\cdot[\mathrm{B2gA}])$$

$$\frac{\mathrm{d}([\mathrm{B}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} = +V_{\mathrm{cell}}\cdot(\mathrm{ge}\cdot[\mathrm{gB}]) - 2\cdot V_{\mathrm{cell}}\cdot(\mathrm{dim}\cdot[\mathrm{B}]\cdot[\mathrm{B}]) + 2\cdot V_{\mathrm{cell}}\cdot(\mathrm{dim\_r}\cdot[\mathrm{B2}]) - V_{\mathrm{cell}}\cdot(\mathrm{deg}\cdot[\mathrm{B}]) + V_{\mathrm{cell}}\cdot(\mathrm{aut\_2}\cdot[\mathrm{B2gB}])$$

$$\frac{\mathrm{d}([\mathrm{gB}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} = -V_{\mathrm{cell}}\cdot(\mathrm{rep}\cdot[\mathrm{gB}]\cdot[\mathrm{A2}]) + V_{\mathrm{cell}}\cdot(\mathrm{rep\_r}\cdot[\mathrm{A2gB}]) - V_{\mathrm{cell}}\cdot(\mathrm{aut\_1}\cdot[\mathrm{B2}]\cdot[\mathrm{gB}]) + V_{\mathrm{cell}}\cdot(\mathrm{aut\_3}\cdot[\mathrm{B2gB}])$$

$$\frac{\mathrm{d}([\mathrm{A2}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} = +V_{\mathrm{cell}}\cdot(\mathrm{dim}\cdot[\mathrm{A}]\cdot[\mathrm{A}]) - V_{\mathrm{cell}}\cdot(\mathrm{dim\_r}\cdot[\mathrm{A2}]) - V_{\mathrm{cell}}\cdot(\mathrm{rep}\cdot[\mathrm{gB}]\cdot[\mathrm{A2}]) + V_{\mathrm{cell}}\cdot(\mathrm{rep\_r}\cdot[\mathrm{A2gB}]) - V_{\mathrm{cell}}\cdot(\mathrm{rep\_dim}\cdot[\mathrm{S}]\cdot[\mathrm{A2}]) + V_{\mathrm{cell}}\cdot(\mathrm{rep\_dim\_r}\cdot[\mathrm{SA2}])$$

$$\frac{\mathrm{d}([\mathrm{B2}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} = +V_{\mathrm{cell}}\cdot(\mathrm{dim}\cdot[\mathrm{B}]\cdot[\mathrm{B}]) - V_{\mathrm{cell}}\cdot(\mathrm{dim\_r}\cdot[\mathrm{B2}]) - V_{\mathrm{cell}}\cdot(\mathrm{rep}\cdot[\mathrm{gA}]\cdot[\mathrm{B2}]) + V_{\mathrm{cell}}\cdot(\mathrm{rep\_r}\cdot[\mathrm{B2gA}]) - V_{\mathrm{cell}}\cdot(\mathrm{aut\_1}\cdot[\mathrm{B2}]\cdot[\mathrm{gB}]) + V_{\mathrm{cell}}\cdot(\mathrm{aut\_3}\cdot[\mathrm{B2gB}]) - V_{\mathrm{cell}}\cdot(\mathrm{rep\_dim}\cdot[\mathrm{R}]\cdot[\mathrm{B2}]) + V_{\mathrm{cell}}\cdot(\mathrm{rep\_dim\_r}\cdot[\mathrm{RB2}])$$

$$\frac{\mathrm{d}([\mathrm{B2gA}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} = +V_{\mathrm{cell}}\cdot(\mathrm{rep}\cdot[\mathrm{gA}]\cdot[\mathrm{B2}]) - V_{\mathrm{cell}}\cdot(\mathrm{rep\_r}\cdot[\mathrm{B2gA}])$$

$$\frac{\mathrm{d}([\mathrm{A2gB}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} = +V_{\mathrm{cell}}\cdot(\mathrm{rep}\cdot[\mathrm{gB}]\cdot[\mathrm{A2}]) - V_{\mathrm{cell}}\cdot(\mathrm{rep\_r}\cdot[\mathrm{A2gB}])$$

$$\frac{\mathrm{d}([\mathrm{B2gB}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} = +V_{\mathrm{cell}}\cdot(\mathrm{aut\_1}\cdot[\mathrm{B2}]\cdot[\mathrm{gB}]) - V_{\mathrm{cell}}\cdot(\mathrm{aut\_3}\cdot[\mathrm{B2gB}])$$

$$\frac{\mathrm{d}([\mathrm{S}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} = -V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim}\cdot[\mathrm{S}]\cdot[\mathrm{A2}]\right)+V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim\_r}\cdot[\mathrm{SA2}]\right)-V_{\mathrm{cell}}\cdot\left(\mathrm{deg\_sr}\cdot[\mathrm{S}]\right)$$

$$\frac{\mathrm{d}([\mathrm{SA2}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} = +V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim}\cdot[\mathrm{S}]\cdot[\mathrm{A2}]\right)-V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim\_r}\cdot[\mathrm{SA2}]\right)$$

$$\frac{\mathrm{d}([\mathrm{R}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} = -V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim}\cdot[\mathrm{R}]\cdot[\mathrm{B2}]\right)+V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim\_r}\cdot[\mathrm{RB2}]\right)-V_{\mathrm{cell}}\cdot\left(\mathrm{deg\_sr}\cdot[\mathrm{R}]\right)$$

$$\frac{\mathrm{d}([\mathrm{RB2}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} = +V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim}\cdot[\mathrm{R}]\cdot[\mathrm{B2}]\right)-V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim\_r}\cdot[\mathrm{RB2}]\right)$$

Positive autoregulation on A with inducers

$$\begin{aligned}
\frac{\mathrm{d}([\mathrm{A}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= +V_{\mathrm{cell}}\cdot(\mathrm{aut_2}\cdot[\mathrm{A2gA}]) - V_{\mathrm{cell}}\cdot(\mathrm{deg}\cdot[\mathrm{A}]) + 2\cdot V_{\mathrm{cell}}\cdot(\mathrm{dim_r}\cdot[\mathrm{A2}]) \\
&\quad - 2\cdot V_{\mathrm{cell}}\cdot(\mathrm{dim}\cdot[\mathrm{A}]\cdot[\mathrm{A}]) + V_{\mathrm{cell}}\cdot(\mathrm{ge}\cdot[\mathrm{gA}]) \\
\frac{\mathrm{d}([\mathrm{gA}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= +V_{\mathrm{cell}}\cdot(\mathrm{aut_3}\cdot[\mathrm{A2gA}]) - V_{\mathrm{cell}}\cdot(\mathrm{aut_1}\cdot[\mathrm{A2}]\cdot[\mathrm{gA}]) + V_{\mathrm{cell}}\cdot(\mathrm{rep_r}\cdot[\mathrm{B2gA}]) \\
&\quad - V_{\mathrm{cell}}\cdot(\mathrm{rep}\cdot[\mathrm{gA}]\cdot[\mathrm{B2}]) \\
\frac{\mathrm{d}([\mathrm{B}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= -V_{\mathrm{cell}}\cdot(\mathrm{deg}\cdot[\mathrm{B}]) + 2\cdot V_{\mathrm{cell}}\cdot(\mathrm{dim_r}\cdot[\mathrm{B2}]) - 2\cdot V_{\mathrm{cell}}\cdot(\mathrm{dim}\cdot[\mathrm{B}]\cdot[\mathrm{B}]) \\
&\quad + V_{\mathrm{cell}}\cdot(\mathrm{ge}\cdot[\mathrm{gB}]) \\
\frac{\mathrm{d}([\mathrm{gB}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= +V_{\mathrm{cell}}\cdot(\mathrm{rep_r}\cdot[\mathrm{A2gB}]) - V_{\mathrm{cell}}\cdot(\mathrm{rep}\cdot[\mathrm{gB}]\cdot[\mathrm{A2}]) \\
\frac{\mathrm{d}([\mathrm{A2}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= +V_{\mathrm{cell}}\cdot(\mathrm{rep_dim_r}\cdot[\mathrm{SA2}]) - V_{\mathrm{cell}}\cdot(\mathrm{rep_dim}\cdot[\mathrm{S}]\cdot[\mathrm{A2}]) + V_{\mathrm{cell}}\cdot(\mathrm{aut_3}\cdot[\mathrm{A2gA}]) \\
&\quad - V_{\mathrm{cell}}\cdot(\mathrm{aut_1}\cdot[\mathrm{A2}]\cdot[\mathrm{gA}]) + V_{\mathrm{cell}}\cdot(\mathrm{rep_r}\cdot[\mathrm{A2gB}]) - V_{\mathrm{cell}}\cdot(\mathrm{rep}\cdot[\mathrm{gB}]\cdot[\mathrm{A2}]) \\
&\quad - V_{\mathrm{cell}}\cdot(\mathrm{dim_r}\cdot[\mathrm{A2}]) + V_{\mathrm{cell}}\cdot(\mathrm{dim}\cdot[\mathrm{A}]\cdot[\mathrm{A}]) \\
\frac{\mathrm{d}([\mathrm{B2}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= +V_{\mathrm{cell}}\cdot(\mathrm{rep_dim_r}\cdot[\mathrm{RB2}]) - V_{\mathrm{cell}}\cdot(\mathrm{rep_dim}\cdot[\mathrm{R}]\cdot[\mathrm{B2}]) + V_{\mathrm{cell}}\cdot(\mathrm{rep_r}\cdot[\mathrm{B2gA}]) \\
&\quad - V_{\mathrm{cell}}\cdot(\mathrm{rep}\cdot[\mathrm{gA}]\cdot[\mathrm{B2}]) - V_{\mathrm{cell}}\cdot(\mathrm{dim_r}\cdot[\mathrm{B2}]) + V_{\mathrm{cell}}\cdot(\mathrm{dim}\cdot[\mathrm{B}]\cdot[\mathrm{B}]) \\
\frac{\mathrm{d}([\mathrm{B2gA}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= -V_{\mathrm{cell}}\cdot(\mathrm{rep_r}\cdot[\mathrm{B2gA}]) + V_{\mathrm{cell}}\cdot(\mathrm{rep}\cdot[\mathrm{gA}]\cdot[\mathrm{B2}]) \\
\frac{\mathrm{d}([\mathrm{A2gB}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= -V_{\mathrm{cell}}\cdot(\mathrm{rep_r}\cdot[\mathrm{A2gB}]) + V_{\mathrm{cell}}\cdot(\mathrm{rep}\cdot[\mathrm{gB}]\cdot[\mathrm{A2}]) \\
\frac{\mathrm{d}([\mathrm{A2gA}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= -V_{\mathrm{cell}}\cdot(\mathrm{aut_3}\cdot[\mathrm{A2gA}]) + V_{\mathrm{cell}}\cdot(\mathrm{aut_1}\cdot[\mathrm{A2}]\cdot[\mathrm{gA}])
\end{aligned}$$

$$\frac{\mathrm{d}([\mathrm{S}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} = -V_{\mathrm{cell}}\cdot\left(\mathrm{deg\_sr}\cdot[\mathrm{S}]\right)+V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim\_r}\cdot[\mathrm{SA2}]\right)-V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim}\cdot[\mathrm{S}]\cdot[\mathrm{A2}]\right)$$

$$\frac{\mathrm{d}([\mathrm{SA2}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} = -V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim\_r}\cdot[\mathrm{SA2}]\right)+V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim}\cdot[\mathrm{S}]\cdot[\mathrm{A2}]\right)$$

$$\frac{\mathrm{d}([\mathrm{R}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} = -V_{\mathrm{cell}}\cdot\left(\mathrm{deg\_sr}\cdot[\mathrm{R}]\right)+V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim\_r}\cdot[\mathrm{RB2}]\right)-V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim}\cdot[\mathrm{R}]\cdot[\mathrm{B2}]\right)$$

$$\frac{\mathrm{d}([\mathrm{RB2}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} = -V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim\_r}\cdot[\mathrm{RB2}]\right)+V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim}\cdot[\mathrm{R}]\cdot[\mathrm{B2}]\right)$$

A.1.2 Positive autoregulation on A and B with inducers

$$
\begin{aligned}
\frac{\mathrm{d}([\mathrm{A}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= +2\cdot V_{\mathrm{cell}}\cdot(\mathrm{dim\_r}\cdot[\mathrm{A2}]) - 2\cdot V_{\mathrm{cell}}\cdot(\mathrm{dim}\cdot[\mathrm{A}]\cdot[\mathrm{A}]) + V_{\mathrm{cell}}\cdot(\mathrm{ge}\cdot[\mathrm{gA}]) \\
&\quad + V_{\mathrm{cell}}\cdot(\mathrm{aut\_2}\cdot[\mathrm{A2gA}]) - V_{\mathrm{cell}}\cdot(\mathrm{deg}\cdot[\mathrm{A}]) \\
\frac{\mathrm{d}([\mathrm{gA}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= +V_{\mathrm{cell}}\cdot(\mathrm{aut\_3}\cdot[\mathrm{A2gA}]) + V_{\mathrm{cell}}\cdot(\mathrm{rep\_r}\cdot[\mathrm{B2gA}]) - V_{\mathrm{cell}}\cdot(\mathrm{rep}\cdot[\mathrm{gA}]\cdot[\mathrm{B2}]) \\
&\quad - V_{\mathrm{cell}}\cdot(\mathrm{aut\_1}\cdot[\mathrm{A2}]\cdot[\mathrm{gA}]) \\
\frac{\mathrm{d}([\mathrm{B}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= +2\cdot V_{\mathrm{cell}}\cdot(\mathrm{dim\_r}\cdot[\mathrm{B2}]) - 2\cdot V_{\mathrm{cell}}\cdot(\mathrm{dim}\cdot[\mathrm{B}]\cdot[\mathrm{B}]) + V_{\mathrm{cell}}\cdot(\mathrm{ge}\cdot[\mathrm{gB}]) \\
&\quad + V_{\mathrm{cell}}\cdot(\mathrm{aut\_2}\cdot[\mathrm{B2gB}]) - V_{\mathrm{cell}}\cdot(\mathrm{deg}\cdot[\mathrm{B}]) \\
\frac{\mathrm{d}([\mathrm{gB}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= -V_{\mathrm{cell}}\cdot(\mathrm{rep}\cdot[\mathrm{gB}]\cdot[\mathrm{A2}]) + V_{\mathrm{cell}}\cdot(\mathrm{aut\_3}\cdot[\mathrm{B2gB}]) - V_{\mathrm{cell}}\cdot(\mathrm{aut\_1}\cdot[\mathrm{B2}]\cdot[\mathrm{gB}]) \\
&\quad + V_{\mathrm{cell}}\cdot(\mathrm{rep\_r}\cdot[\mathrm{A2gB}]) \\
\frac{\mathrm{d}([\mathrm{A2}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= +V_{\mathrm{cell}}\cdot(\mathrm{aut\_3}\cdot[\mathrm{A2gA}]) - V_{\mathrm{cell}}\cdot(\mathrm{rep}\cdot[\mathrm{gB}]\cdot[\mathrm{A2}]) - V_{\mathrm{cell}}\cdot(\mathrm{dim\_r}\cdot[\mathrm{A2}]) \\
&\quad + V_{\mathrm{cell}}\cdot(\mathrm{dim}\cdot[\mathrm{A}]\cdot[\mathrm{A}]) - V_{\mathrm{cell}}\cdot(\mathrm{aut\_1}\cdot[\mathrm{A2}]\cdot[\mathrm{gA}]) - V_{\mathrm{cell}}\cdot(\mathrm{rep\_dim}\cdot[\mathrm{S}]\cdot[\mathrm{A2}]) \\
&\quad + V_{\mathrm{cell}}\cdot(\mathrm{rep\_dim\_r}\cdot[\mathrm{SA2}]) + V_{\mathrm{cell}}\cdot(\mathrm{rep\_r}\cdot[\mathrm{A2gB}]) \\
\frac{\mathrm{d}([\mathrm{B2}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= +V_{\mathrm{cell}}\cdot(\mathrm{rep\_r}\cdot[\mathrm{B2gA}]) - V_{\mathrm{cell}}\cdot(\mathrm{rep}\cdot[\mathrm{gA}]\cdot[\mathrm{B2}]) - V_{\mathrm{cell}}\cdot(\mathrm{dim\_r}\cdot[\mathrm{B2}]) \\
&\quad + V_{\mathrm{cell}}\cdot(\mathrm{dim}\cdot[\mathrm{B}]\cdot[\mathrm{B}]) + V_{\mathrm{cell}}\cdot(\mathrm{aut\_3}\cdot[\mathrm{B2gB}]) - V_{\mathrm{cell}}\cdot(\mathrm{rep\_dim}\cdot[\mathrm{R}]\cdot[\mathrm{B2}]) \\
&\quad + V_{\mathrm{cell}}\cdot(\mathrm{rep\_dim\_r}\cdot[\mathrm{RB2}]) - V_{\mathrm{cell}}\cdot(\mathrm{aut\_1}\cdot[\mathrm{B2}]\cdot[\mathrm{gB}]) \\
\frac{\mathrm{d}([\mathrm{B2gA}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= -V_{\mathrm{cell}}\cdot(\mathrm{rep\_r}\cdot[\mathrm{B2gA}]) + V_{\mathrm{cell}}\cdot(\mathrm{rep}\cdot[\mathrm{gA}]\cdot[\mathrm{B2}])
\end{aligned}
$$

$$\frac{\mathrm{d}\left([\mathrm{A2gB}]\cdot V_{\mathrm{cell}}\right)}{\mathrm{d}t} = +V_{\mathrm{cell}}\cdot\left(\mathrm{rep}\cdot[\mathrm{gB}]\cdot[\mathrm{A2}]\right)-V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_r}\cdot[\mathrm{A2gB}]\right)$$

$$\frac{\mathrm{d}\left([\mathrm{B2gB}]\cdot V_{\mathrm{cell}}\right)}{\mathrm{d}t} = -V_{\mathrm{cell}}\cdot\left(\mathrm{aut\_3}\cdot[\mathrm{B2gB}]\right)+V_{\mathrm{cell}}\cdot\left(\mathrm{aut\_1}\cdot[\mathrm{B2}]\cdot[\mathrm{gB}]\right)$$

$$\frac{\mathrm{d}\left([\mathrm{A2gA}]\cdot V_{\mathrm{cell}}\right)}{\mathrm{d}t} = -V_{\mathrm{cell}}\cdot\left(\mathrm{aut\_3}\cdot[\mathrm{A2gA}]\right)+V_{\mathrm{cell}}\cdot\left(\mathrm{aut\_1}\cdot[\mathrm{A2}]\cdot[\mathrm{gA}]\right)$$

$$\frac{\mathrm{d}\left([\mathrm{S}]\cdot V_{\mathrm{cell}}\right)}{\mathrm{d}t} = -V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim}\cdot[\mathrm{S}]\cdot[\mathrm{A2}]\right)+V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim\_r}\cdot[\mathrm{SA2}]\right)-V_{\mathrm{cell}}\cdot\left(\mathrm{deg\_sr}\cdot[\mathrm{S}]\right)$$

$$\frac{\mathrm{d}\left([\mathrm{SA2}]\cdot V_{\mathrm{cell}}\right)}{\mathrm{d}t} = +V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim}\cdot[\mathrm{S}]\cdot[\mathrm{A2}]\right)-V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim\_r}\cdot[\mathrm{SA2}]\right)$$

$$\frac{\mathrm{d}\left([\mathrm{R}]\cdot V_{\mathrm{cell}}\right)}{\mathrm{d}t} = -V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim}\cdot[\mathrm{R}]\cdot[\mathrm{B2}]\right)+V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim\_r}\cdot[\mathrm{RB2}]\right)-V_{\mathrm{cell}}\cdot\left(\mathrm{deg\_sr}\cdot[\mathrm{R}]\right)$$

$$\frac{\mathrm{d}\left([\mathrm{RB2}]\cdot V_{\mathrm{cell}}\right)}{\mathrm{d}t} = +V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim}\cdot[\mathrm{R}]\cdot[\mathrm{B2}]\right)-V_{\mathrm{cell}}\cdot\left(\mathrm{rep\_dim\_r}\cdot[\mathrm{RB2}]\right)$$

A.1.3 CS-MA

$$\begin{aligned}
\frac{\mathrm{d}([\mathrm{A}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= +2\cdot V_{\mathrm{cell}}\cdot(\mathrm{dim_r}\cdot[\mathrm{A2}]) - 2\cdot V_{\mathrm{cell}}\cdot(\mathrm{dim}\cdot[\mathrm{A}]\cdot[\mathrm{A}]) + V_{\mathrm{cell}}\cdot\big(\mathrm{geA}\cdot[\mathrm{gA}]\big) \\
&\quad - V_{\mathrm{cell}}\cdot\big(\mathrm{deg}\cdot[\mathrm{A}]\big) \\
\frac{\mathrm{d}\big([\mathrm{gA}]\cdot V_{\mathrm{cell}}\big)}{\mathrm{d}t} &= +V_{\mathrm{cell}}\cdot\big(\mathrm{rep_r}\cdot[\mathrm{B2gA}]\big) \\
&\quad - V_{\mathrm{cell}}\cdot\big(\mathrm{repA}\cdot[\mathrm{gA}]\cdot[\mathrm{B2}]\big) \\
\frac{\mathrm{d}([\mathrm{B}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= +2\cdot V_{\mathrm{cell}}\cdot(\mathrm{dim_r}\cdot[\mathrm{B2}]) - 2\cdot V_{\mathrm{cell}}\cdot(\mathrm{dim}\cdot[\mathrm{B}]\cdot[\mathrm{B}]) \\
&\quad + V_{\mathrm{cell}}\cdot\big(\mathrm{geB}\cdot[\mathrm{gB}]\big) - V_{\mathrm{cell}}\cdot\big(\mathrm{deg}\cdot[\mathrm{B}]\big) \\
\frac{\mathrm{d}\big([\mathrm{gB}]\cdot V_{\mathrm{cell}}\big)}{\mathrm{d}t} &= +V_{\mathrm{cell}}\cdot\big(\mathrm{rep_r}\cdot[\mathrm{A2gB}]\big) - V_{\mathrm{cell}}\cdot\big(\mathrm{repB}\cdot[\mathrm{gB}]\cdot[\mathrm{A2}]\big) \\
\frac{\mathrm{d}([\mathrm{A2}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= -V_{\mathrm{cell}}\cdot(\mathrm{dim_r}\cdot[\mathrm{A2}]) + V_{\mathrm{cell}}\cdot(\mathrm{dim}\cdot[\mathrm{A}]\cdot[\mathrm{A}]) - V_{\mathrm{cell}}\cdot\big(\mathrm{deg_dim}\cdot[\mathrm{A2}]\big) \\
&\quad + V_{\mathrm{cell}}\cdot\big(\mathrm{rep_r}\cdot[\mathrm{A2gB}]\big) - V_{\mathrm{cell}}\cdot\big(\mathrm{repB}\cdot[\mathrm{gB}]\cdot[\mathrm{A2}]\big) \\
\frac{\mathrm{d}([\mathrm{B2}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= -V_{\mathrm{cell}}\cdot(\mathrm{dim_r}\cdot[\mathrm{B2}]) + V_{\mathrm{cell}}\cdot(\mathrm{dim}\cdot[\mathrm{B}]\cdot[\mathrm{B}]) - V_{\mathrm{cell}}\cdot\big(\mathrm{deg_dim}\cdot[\mathrm{B2}]\big) \\
&\quad + V_{\mathrm{cell}}\cdot\big(\mathrm{rep_r}\cdot[\mathrm{B2gA}]\big) - V_{\mathrm{cell}}\cdot\big(\mathrm{repA}\cdot[\mathrm{gA}]\cdot[\mathrm{B2}]\big) \\
\frac{\mathrm{d}\big([\mathrm{A2gB}]\cdot V_{\mathrm{cell}}\big)}{\mathrm{d}t} &= -V_{\mathrm{cell}}\cdot\big(\mathrm{rep_r}\cdot[\mathrm{A2gB}]\big) + V_{\mathrm{cell}}\cdot\big(\mathrm{repB}\cdot[\mathrm{gB}]\cdot[\mathrm{A2}]\big) \\
\frac{\mathrm{d}\big([\mathrm{B2gA}]\cdot V_{\mathrm{cell}}\big)}{\mathrm{d}t} &= -V_{\mathrm{cell}}\cdot\big(\mathrm{rep_r}\cdot[\mathrm{B2gA}]\big) + V_{\mathrm{cell}}\cdot\big(\mathrm{repA}\cdot[\mathrm{gA}]\cdot[\mathrm{B2}]\big)
\end{aligned}$$

The CS-MA switch was simulated using stochastic dynamics. The stoichiometry matrix and hazards defining the model are shown below:

$$\begin{aligned}
h[1] &= geA \times gA \\
h[2] &= geB \times gB \\
h[3] &= dim \times A^2 \\
h[4] &= dim \times B^2 \\
h[5] &= dim_r \times B2
\end{aligned}$$

Table A.1 CS-MA stoichiometry matrix

1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
-2.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	0.0	-2.0	0.0	0.0	1.0	0.0	0.0
0.0	0.0	2.0	0.0	0.0	-1.0	0.0	0.0
2.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0
0.0	-1.0	0.0	0.0	0.0	-1.0	0.0	1.0
0.0	1.0	0.0	0.0	0.0	1.0	0.0	-1.0
0.0	0.0	0.0	-1.0	-1.0	0.0	1.0	0.0
0.0	0.0	0.0	1.0	1.0	0.0	-1.0	0.0
-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	-1.0	0.0	0.0

$$h[6] = dim\_r \times A2$$

$$h[7] = geA \times gA \times B2$$

$$h[8] = rep\_r \times B2gA$$

$$h[9] = repB \times gB \times A2$$

$$h[10] = rep\_r \times A2gB$$

$$h[11] = deg \times A$$

$$h[12] = deg \times B$$

$$h[13] = deg\_dim \times A2$$

$$h[14] = deg\_dim \times B2$$

A.1.4 DP-MA

$$
\begin{aligned}
\frac{\mathrm{d}([\mathrm{A}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= -V_{\mathrm{cell}}\cdot(\mathrm{deg}\cdot[\mathrm{A}]) + 2\cdot V_{\mathrm{cell}}\cdot(\mathrm{dim\_r}\cdot[\mathrm{A2}]) - 2\cdot V_{\mathrm{cell}}\cdot(\mathrm{dim}\cdot[\mathrm{A}]\cdot[\mathrm{A}]) \\
&\quad + V_{\mathrm{cell}}\cdot(\mathrm{geA}\cdot[\mathrm{gA}]) + V_{\mathrm{cell}}\cdot(\mathrm{aut\_2}\cdot[\mathrm{A2gA}]) \\
\frac{\mathrm{d}([\mathrm{gA}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= -V_{\mathrm{cell}}\cdot(\mathrm{aut\_1}\cdot[\mathrm{A2}]\cdot[\mathrm{gA}]) + V_{\mathrm{cell}}\cdot(\mathrm{rep\_r}\cdot[\mathrm{B2gA}]) - V_{\mathrm{cell}}\cdot(\mathrm{repA}\cdot[\mathrm{gA}]\cdot[\mathrm{B2}]) \\
&\quad + V_{\mathrm{cell}}\cdot(\mathrm{aut\_3}\cdot[\mathrm{A2gA}]) \\
\frac{\mathrm{d}([\mathrm{B}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= +V_{\mathrm{cell}}\cdot(\mathrm{aut\_2}\cdot[\mathrm{B2gB}]) - V_{\mathrm{cell}}\cdot(\mathrm{deg}\cdot[\mathrm{B}]) + 2\cdot V_{\mathrm{cell}}\cdot(\mathrm{dim\_r}\cdot[\mathrm{B2}]) \\
&\quad - 2\cdot V_{\mathrm{cell}}\cdot(\mathrm{dim}\cdot[\mathrm{B}]\cdot[\mathrm{B}]) + V_{\mathrm{cell}}\cdot(\mathrm{geB}\cdot[\mathrm{gB}]) \\
\frac{\mathrm{d}([\mathrm{gB}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= +V_{\mathrm{cell}}\cdot(\mathrm{aut\_3}\cdot[\mathrm{B2gB}]) - V_{\mathrm{cell}}\cdot(\mathrm{aut\_1}\cdot[\mathrm{B2}]\cdot[\mathrm{gB}]) + V_{\mathrm{cell}}\cdot(\mathrm{rep\_r}\cdot[\mathrm{A2gB}]) \\
&\quad - V_{\mathrm{cell}}\cdot(\mathrm{repB}\cdot[\mathrm{gB}]\cdot[\mathrm{A2}]) \\
\frac{\mathrm{d}([\mathrm{A2}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= -V_{\mathrm{cell}}\cdot(\mathrm{aut\_1}\cdot[\mathrm{A2}]\cdot[\mathrm{gA}]) + V_{\mathrm{cell}}\cdot(\mathrm{rep\_r}\cdot[\mathrm{A2gB}]) - V_{\mathrm{cell}}\cdot(\mathrm{repB}\cdot[\mathrm{gB}]\cdot[\mathrm{A2}]) \\
&\quad - V_{\mathrm{cell}}\cdot(\mathrm{dim\_r}\cdot[\mathrm{A2}]) + V_{\mathrm{cell}}\cdot(\mathrm{dim}\cdot[\mathrm{A}]\cdot[\mathrm{A}]) + V_{\mathrm{cell}}\cdot(\mathrm{aut\_3}\cdot[\mathrm{A2gA}]) \\
&\quad - V_{\mathrm{cell}}\cdot(\mathrm{deg\_dim}\cdot[\mathrm{A2}]) \\
\frac{\mathrm{d}([\mathrm{B2}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= +V_{\mathrm{cell}}\cdot(\mathrm{aut\_3}\cdot[\mathrm{B2gB}]) - V_{\mathrm{cell}}\cdot(\mathrm{aut\_1}\cdot[\mathrm{B2}]\cdot[\mathrm{gB}]) + V_{\mathrm{cell}}\cdot(\mathrm{rep\_r}\cdot[\mathrm{B2gA}]) \\
&\quad - V_{\mathrm{cell}}\cdot(\mathrm{repA}\cdot[\mathrm{gA}]\cdot[\mathrm{B2}]) - V_{\mathrm{cell}}\cdot(\mathrm{dim\_r}\cdot[\mathrm{B2}]) \\
&\quad + V_{\mathrm{cell}}\cdot(\mathrm{dim}\cdot[\mathrm{B}]\cdot[\mathrm{B}]) - V_{\mathrm{cell}}\cdot(\mathrm{deg\_dim}\cdot[\mathrm{B2}]) \\
\frac{\mathrm{d}([\mathrm{B2gA}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= -V_{\mathrm{cell}}\cdot(\mathrm{rep\_r}\cdot[\mathrm{B2gA}]) + V_{\mathrm{cell}}\cdot(\mathrm{repA}\cdot[\mathrm{gA}]\cdot[\mathrm{B2}]) \\
\frac{\mathrm{d}([\mathrm{A2gB}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= -V_{\mathrm{cell}}\cdot(\mathrm{rep\_r}\cdot[\mathrm{A2gB}]) + V_{\mathrm{cell}}\cdot(\mathrm{repB}\cdot[\mathrm{gB}]\cdot[\mathrm{A2}]) \\
\frac{\mathrm{d}([\mathrm{B2gB}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= -V_{\mathrm{cell}}\cdot(\mathrm{aut\_3}\cdot[\mathrm{B2gB}]) + V_{\mathrm{cell}}\cdot(\mathrm{aut\_1}\cdot[\mathrm{B2}]\cdot[\mathrm{gB}]) \\
\frac{\mathrm{d}([\mathrm{A2gA}]\cdot V_{\mathrm{cell}})}{\mathrm{d}t} &= +V_{\mathrm{cell}}\cdot(\mathrm{aut\_1}\cdot[\mathrm{A2}]\cdot[\mathrm{gA}]) - V_{\mathrm{cell}}\cdot(\mathrm{aut\_3}\cdot[\mathrm{A2gA}])
\end{aligned}
$$

The stoichiometry matrix and hazards defining the DP-MA model are given below.

Table A.2 DP-MA stoichiometry matrix

1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
-2.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	-2.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
0.0	0.0	2.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0
2.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0
0.0	-1.0	0.0	0.0	0.0	-1.0	1.0	0.0	0.0	0.0
0.0	1.0	0.0	0.0	0.0	1.0	-1.0	0.0	0.0	0.0
0.0	0.0	0.0	-1.0	-1.0	0.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	1.0	1.0	0.0	0.0	-1.0	0.0	0.0
-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	-1.0	0.0	-1.0	0.0	0.0	1.0	0.0
0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	-1.0	0.0
0.0	-1.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	1.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	-1.0
0.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0	0.0

$$h[1] = repA \times gA$$
$$h[2] = repB \times gB$$
$$h[3] = dim \times A^2$$
$$h[4] = dim\_r \times B^2$$
$$h[5] = dim\_r \times B2$$
$$h[6] = deg \times A2$$
$$h[7] = rep\_r \times gA \times B2$$
$$h[8] = rep\_r \times B2gA$$
$$h[9] = repB \times gB \times A2$$
$$h[10] = dim \times A2gB$$
$$h[11] = deg \times A$$
$$h[12] = deg \times B$$

$$h[13] = aut\_1 \times B2 \times gB$$

$$h[14] = aut\_2 \times B2gB$$

$$h[15] = aut\_3 \times B2gB$$

$$h[16] = aut\_1 \times A2 \times gA$$

$$h[17] = aut\_2 \times A2gA$$

$$h[18] = aut\_3 \times A2gA$$

$$h[19] = deg\_dim \times A2$$

$$h[20] = deg\_dim \times B2$$

B Primers

B.1 Primers used during PCR and sequencing

Table B.1 List of primers used for PCR amplification

Direction	Amplifies	Added site	Sequence
Forward	Plac/ara-1	NcoI	TAAGCACCATGGCTCGAGCATAGCATTTTTATCCAT
Reverse	Plac/ara-1	SalI	AAGCAGGTCGACTTCTGTGTGAAATTGTTATCCGC
Forward	Plac/ara-1	XmaI	TAAGCACCCGGGCTCGAGCATAGCATTTTTATCCAT
Reverse	Plac/ara-1	KasI	AAGCAGGGCGCCCTTTCTCCTCTTTAATGAATTCTGTGT
Forward	Plac/ara-1, AraC	EagI	TAAGCACGGCCGCTCGAGCATAGCATTTTTATC
Reverse	Plac/ara-1, AraC	SalI	AAGCAGGTCGACCTAATTAAGCTTTCACGCTG
Forward	LuxR	BsGI	TAAGCATGTACAAGGCCCTTTCGTCTTCAC
Reverse	LuxR	BsGI	AAGCAGTGTACAAGCGATACAATAGTGTGACAA
Forward	mCherry	XmaI	CTCCATATGCTCGTTCCCGGGC
Reverse	mCherry	PstI	CGCTGTCTGCAGCTGCGTTATCCCCTGATTCTGTGGATA
Forward	YFP	EcoRV	ATAGGGAGGCGCCGATGCGTAAAGGGAG
Reverse	YFP	KasI	GCCATAGATATCTTATTATTTGTATAGTTCATCC

C Algorithms

C.1 Clustering algorithms

C.1.1 Deterministic case

Algorithm 8 Clustering the steady state deterministic simulation results

```
1:  for each data point do
2:      if first point then
3:          Make first cluster
4:          cluster counter = 1
5:      else
6:          for each cluster do
7:              if cluster within cluster means ± delta then
8:                  Add to existing cluster
9:                  Update means of clusters
10:             end if
11:             if reached_end and not assigned to cluster then
12:                 cluster counter += 1
13:                 Add new cluster
14:             end if
15:         end for
16:     end if
17: end for
```

C.1.2 Stochastic case

Gap statistic

Algorithm 9 Choosing the optimal number of clusters

```
function Wk(clusters, cluster_centres)
    for each cluster do
        for each point in cluster do
            a = matrix norm (cluster_centre − point)
        end for
        dk = ∑((a)^2) × (2× number of points in cluster)
    end for
    wk = ∑(dk) / (2×(numberofpointsincluster))
    return wk
end function

function Gap_statistic(data, cutoff)
    ks = [1,2,3,4]
    for k in ks do
        cluster_centres, clusters = Kmeans(data, k, cutoff)
        Wk = log(Wk(clusters, cluster_centres))
        Create references datasets
        for each references dataset do
            cluster_centres, clusters = Kmeans(data, k, cutoff)
            BWk = log(Wk(clusters, cluster_centres))
        end for
        Wkb = ∑(BWk) / 10
        sk = sqrt(∑((BWk − Wkb)^2) / 10)
    end for
    sk = sk × sqrt(1 + 1/B)
    return ks, Wk, Wkb, sk, data_centres, clusters
end function

function Distance(data, cutoff)
    ks, logWks, logWkbs, sk, clusters_means, clusts = gap_statistic(data, cutoff)
    gaps = logWks − logWkbs
    optimum number of clusters = gaps[i] ≥ (gaps[i + 1] − sk[i + 1])
    return cluster_counter, clusters_means
end function
```

C.2 K-means clustering

Algorithm 10 Clustering stochastic case

```
1:  function KMEANS CLUSTERING(data, k, cutoff)

2:      function UPDATE_CENTRES(old_centres, values)
3:          centre_coords = mean for each dimension
4:          shift = GETDISTANCE(centre_coords, old_centres)
5:          return shift, centre_coords
6:      end function

7:      function GETDISTANCE(a, b)
8:          dist = sqrt((a[x] − b[x])^2 + (a[y] − b[y])^2)
9:          return dist
10:     end function

11:     while True do
12:         for each point in data do
13:             for each cluster do
14:                 dist =GETDISTANCE(point, cluster centre)
15:             end for
16:             Find cluster with minimum distance
17:             Repopulate clusters
18:         end for
19:         biggest_shift ← 0
20:         for as many times as there are clusters do
21:             shift, cluster centres = UPDATE_CENTRES(old_centres, clusters)
22:             biggest_shift = max between shift, biggest_shift
23:         end for
24:         if biggest_shift ≤ cutoff then
25:             break
26:         end if
27:     end while
28:     return cluster_centres, clusters
29: end function
```

D Additional posterior distributions

D.1 Asymmetric mass action toggle switch posterior distributions

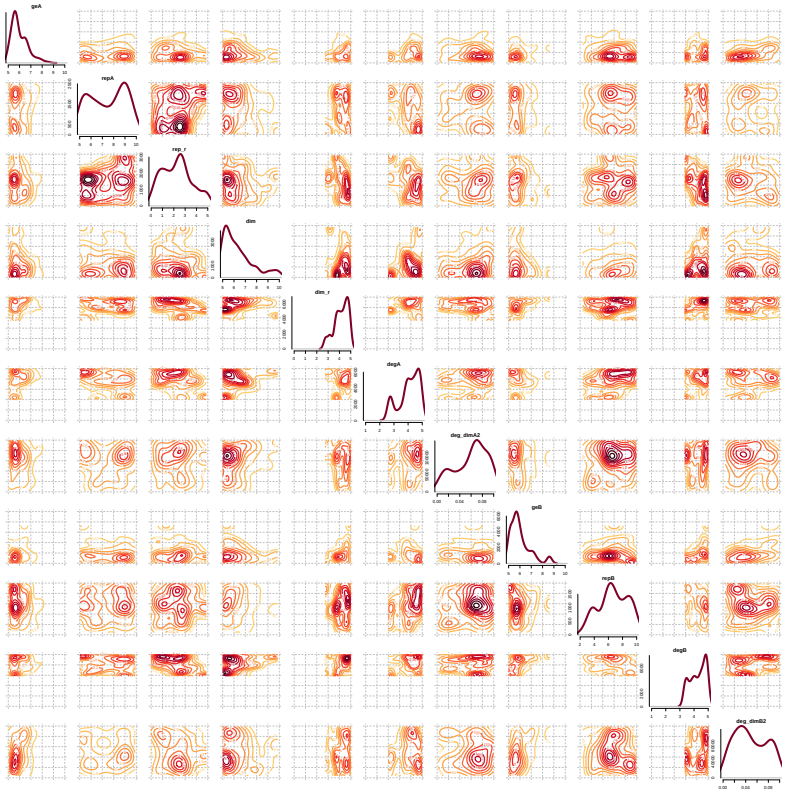

Figure D.1 Asymmetric CS-MA posterior distribution

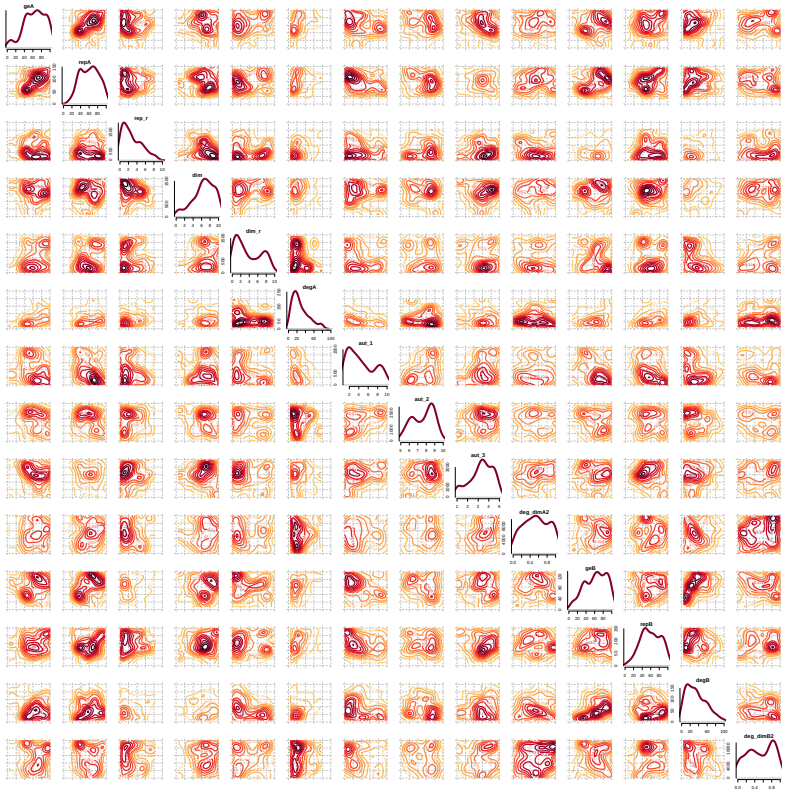

Figure D.2 Asymmetric DP-MA posterior distribution