Bayesian inference and modelling of gene expression dynamics

A thesis submitted to the University of Manchester for the degree of
Doctor of Philosophy
in the Faculty of Biology, Medicine and Health

2023

Joshua M. Burton
Division of Informatics, Imaging and Data Sciences
School of Health Sciences

Contents

Word count: 47489

List of figures

List of tables

List of publications

Burton, J., Manning, C. S., Rattray, M., Papalopulu, N., & Kursawe, J. (2021a). Inferring kinetic parameters of oscillatory gene regulation from single cell time-series data. *Journal of the Royal Society Interface*, *18*(182), 20210393.

Burton, J., Rattray, M., Papalopulu, N., & Kursawe, J. (2023). Continuous-time filtering and variational inference of combined single cell time-series data. *In preparation*.

Funk, S., Abbott, S., Atkins, B. D., Baguelin, M., Baillie, J. K., Birrell, P., Blake, J., Bosse, N. I., Burton, J., Carruthers, J., et al. (2020b). Short-term forecasts to inform the response to the covid-19 epidemic in the uk. *MedRxiv*, 2020–11.

Overton, C. E., Pellis, L., Stage, H. B., Scarabel, F., Burton, J., Fraser, C., Hall, I., House, T. A., Jewell, C., Nurtay, A., et al. (2022a). Epibeds: Data informed modelling of the covid-19 hospital burden in england. *PLoS Computational Biology*, *18*(9), e1010406.

Soto, X., Burton, J., Manning, C. S., Minchington, T., Lea, R., Lee, J., Kursawe, J., Rattray, M., & Papalopulu, N. (2022a). Sequential and additive expression of mir-9 precursors control timing of neurogenesis. *Development*, *149*(19), dev200474.

Terms and abbreviations

ABC	Approximate Bayesian Computation
bHLH	Basic helix-loop-helix
CLE	Chemical Langevin Equation
CNS	Central nervous system
DDE	Delay differential equation
ELBO	Evidence lower bound
ESS	Effective sample size
FISH	Fluorescent *in situ* hybridization
HMC	Hamiltonian Monte Carlo
HPDI	Highest posterior density interval
hpf	Hours post fertilisation
IFFL	Incoherent feed-forward loop
ISH	*In situ* hybridization
KL	Kullback-Leibler
L-BFGS	Limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm
MALA	Metropolis-adjusted Langevin Algorithm
MCMC	Markov chain Monte Carlo
MCSE	Monte Carlo standard error
MH	Metropolis-Hastings
miR	MicroRNA
MLE	Maximum likelihood estimate
NUTS	No-U-Turn Sampler
ODE	Ordinary differential equation
$\hat{\mathbf{R}}$	The R-hat statistic
RK	Runge-Kutta
RWM	Random walk Metropolis
SD	Standard deviation
SDDE	Stochastic delay differential equation
smFISH	Single-molecule fluorescent *in situ* hybridisation

Abstract

The regulation of differentiation is essential for the growth and development of living organisms. Changes in the dynamic expression of certain genes have been associated with differentiation in multiple contexts, including development and cancer. For example, the dynamic expression of the basic helix-loop-helix transcription factors Hes5 and Her6, which are involved in neurogenesis in mice and zebrafish respectively, is mediated by auto-repressive feedback. This auto-repression, when coupled with delays, noise, and nonlinear interactions inherent to biological processes, gives rise to dynamic expression profiles, such as oscillations.

Gene expression dynamics mediated by the auto-repressive feedback motif are influenced by a number of biochemical interactions, including transcription, translation, transcriptional repression and degradation. Therefore, differences in dynamic gene expression between cells that express Hes5 or Her6 may be due to variations in some or all of these interactions. The ability to perform live imaging of gene expression at single-cell resolution presents an opportunity to identify such variations using Bayesian inference methods.

Bayesian inference allows us to estimate biophysical parameters by linking experimental data with mathematical models. However, Bayesian methods that combine stochasticity, delays, and nonlinearity have not been widely adopted. I present an approach for inferring parameters of an auto-negative feedback motif with delay using live-imaging time-series data. This method is applied to published data on murine neural progenitor cells, and the results are used to inform experimental design choices. I subsequently extend this approach and adopt a variational inference method in order to work with combined time-series data from multiple cells with similar dynamic expression. We see a drastic reduction in the uncertainty in our estimates, as well as speed improvements of multiple orders of magnitude. Importantly, our method provides concise and accurate estimates for multiple parameters, such as the production rates of mRNA and protein.

In addition to auto-repression, Her6 dynamics are influenced by miR-9, a microRNA which acts on mRNA stability and protein translation. In a cross-disciplinary collaboration, we show that miR-9 increases in a sharp stepwise manner during zebrafish neurogenesis. To understand the impact of this mode of increase, I develop a mathematical model based on perfect adaptation and interactions between miR-9 and Her6. My results suggest that the stepwise increase facilitates the robustness of cell state transitions in the presence of small-scale fluctuations.

The work in this thesis demonstrates the power of computational methods in interpreting data and understanding how changes in gene expression dynamics are regulated during development.

Declaration of originality

I hereby confirm that no portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright statement

Acknowledgements

I would like to thank Nancy Papalopulu, for always encouraging me, and most importantly for her patience. I would also like to thank Magnus Rattray, for all of his helpful suggestions, and for always having time no matter how busy he was. And Jochen Kursawe, whose enthusiasm for science is contagious – thank you for guiding me in this project, and for helping me find the confidence to ask my own research questions. I feel truly privileged to have been mentored by all of you. I want to thank Mark Muldoon for his supervision of my master's project, and for all of his enlightening and interesting discussions in the early years of my research. He introduced me to the field of mathematical biology, and I am extremely grateful to him for that. I would also like to thank all of the Papalopulu lab members, who provided such a welcoming environment throughout my PhD. I will always think fondly of lunchtimes together on the sofas.

I am grateful to Lorenzo Pellis, and everyone else I worked with in response to the pandemic, for showing me the strength and compassion of the scientific community. In addition, I would like to thank Arno Solin for welcoming me into his research group for 2 months and giving me valuable suggestions for future research. Thanks to Martin, Paul, Severi, Haishan, Rui, and everyone else I met at Aalto University, for making my time in Finland so enjoyable.

To all my friends, especially Josh, Jenny, Hel, Damian, Matt, Will, Tom, Seb and Jo, for being there when I needed them. To Kate, for making me a kinder person, and Olivia, for making me a sillier one. To my Mum and Dad, for their constant love and support. And to Flora, who's been with me every step of the way. You never doubted me and I couldn't have done this without you. Thank you.

This thesis is dedicated to my Grandma, Winifred Gregson.

Chapter 1

Introduction

1.1 Gene expression dynamics and cell fate

The formation and growth of an organism that takes place during embryogenesis is a complex, fine-tuned procedure. Different processes, such as morphogenesis and tissue growth, play roles which must be understood in the context of development, as well as in isolation. The study of these processes is essential to understanding the occurrence, and prevention, of developmental disorders. For example, the failure of cells to take up the appropriate roles within neural tissue represents the major convergence point for neurodevelopmental disorders (Ernst, 2016). This act of adopting a new role is generally referred to as differentiation, but more accurately as cell state transitions or cell fate changes.

Cell fate changes are important for multicellular development and are driven by changes in gene expression. Gene expression is inherently dynamic due to transcriptional pulsing (transcription occurring in pulses or bursts), stochastic fluctuations (random events, like random timings of collisions between molecules), as well as positive or negative feedback loops, which may include delays. The qualitative dynamics of gene expression itself, rather than simply the presence of a gene product or its concentration, may be functionally important (Huang, 2009; Kageyama et al., 2012; Kageyama et al., 2008; Raj & van Oudenaarden, 2008).

Associations between cell fate changes and dynamic gene expression have been observed in various model systems, including somitogenesis, neurogenesis, and in cancer cell lines (Chou et al., 2010; Manning et al., 2019; Marinopoulou et al., 2021; Purvis et al., 2012; Soroldoni & Oates, 2011; Soto et al., 2019). For example, oscillatory expression of tumour suppressor gene p53 has been shown to prevent cells from committing to a fate, suggesting a biological timing mechanism (Purvis et al., 2012). Her and Hes genes are basic helix-loop-helix (bHLH) transcriptional repressors and inhibit neuronal differentiation in zebrafish (Cheng et al., 2015) and mice (Hatakeyama, 2004) respectively. Oscillations in Hes genes are required for the proliferation of neural progenitor cells during neurogenesis in mice (Imayoshi et al., 2013; Manning et al., 2019; Ochi et al., 2020), and oscillatory expression of the Hes orthologs Her1 and

Her7 regulates somite segmentation in zebrafish (Choorapoikayil et al., 2012; Lewis, 2003; A. C. Oates & Ho, 2002). Recently, different dynamic expression profiles of Hes5, such as decreasing expression with or without oscillations, have been linked to different cell fate transitions in murine neural development (Manning et al., 2019). To fully understand the link between gene expression dynamics and cell fate, for example during neurogenesis, it will be essential to determine how different gene expression dynamics emerge.

Many different hypotheses may explain differences in gene expression dynamics, and explanations for the emergence of heterogeneity in clonal populations have been explored at length (Brock et al., 2009; Elowitz et al., 2002; Raj & van Oudenaarden, 2008). These differences may reflect dynamic changes in protein or mRNA degradation, transcriptional or translational activation, transcriptional auto-repression, or post-transcriptional modification. This large number of possible mechanisms makes direct experimental testing of all hypotheses unfeasible, particularly if a combination of effects is to be tested (Lillacci & Khammash, 2010). One approach to overcome this challenge is to combine mathematical modelling and Bayesian inference techniques to evaluate alternative hypotheses based on their ability to describe recorded time series of gene expression. The ideal situation is to choose a biological system where gene expression dynamics influence cell fate decisions, where the interactions between each component are well characterised, and a mathematical model which is able to capture the dynamic behaviour of the gene expression exists. Predictive modelling can then drive focused experimentation to validate the molecular mechanism and be used to identify the key model parameters which influence expression dynamics at the level of a single cell. One such biological system is the auto-repressive negative feedback mechanism that governs the dynamic expression of Hes genes in various contexts (Imayoshi et al., 2013; Kobayashi & Kageyama, 2014; Lewis, 2003; Shimojo et al., 2008; Soto et al., 2020).

1.1.1 Variations in Hes expression demonstrate the need for understanding the heterogeneity of dynamics

As early as 1961 it was suggested that genes might interact by negative feedback or repression (Mahaffy & Pao, 1984). Goodwin (1965) interrogated this idea of negative control mathematically, describing the dynamics of the concentrations of mRNA and protein, in a simple delayed negative feedback model of a single gene. This demonstrated the possibility of undamped oscillatory gene expression, supporting experimental evidence of periodic cellular processes and biological clocks. Evidence of gene expression oscillations has since been found, however, these oscillations are typically circadian, i.e. linked to the day-night cycle (Levi & Schibler, 2007). During embryonic development responses at shorter time scales are possible. One example of this is in the formation of somites in the developing zebrafish embryo, where the protein concentrations of the Her1 and Her7 genes oscillate with periods of around 30 minutes (Lewis, 2003). When these oscillations are simulated mathematically with a simple feedback model using experimentally informed parameters, the generated period is close to the experimentally observed one, suggesting that this model may capture most of the relevant biological information (Lewis, 2003). Similar genes in other systems also oscillate (Imayoshi et al., 2013). The bHLH factors Hes1 and Hes5 in the mouse have been shown to undergo oscillations in neural progenitors, and the extent to which hes1 oscillates in a population of cells has been accurately quantified (Phillips et al., 2017). They have been found to repress their own expression due to their proteins binding to their own transcription site (Bansod et al., 2017; Hatakeyama, 2004; Hirata et al., 2002; Kobayashi & Kageyama, 2014; Manning et al., 2019; Monk, 2003; Tsutsumi et al., 2012). Although oscillatory expression of these genes plays a key role, other dynamic expression profiles have been observed, and have different functional roles. Shimojo et al. (2008) theorised it is likely that sustained expression of Hes1 forms boundary cells in the developing central nervous system. In contrast, oscillatory expression causes the formation of compartment cells in the embryonic brain and persistent and high levels of Hes1 expression inhibit neuronal differentiation and cell cycle progression by repressing the expression of proneural genes and cell cycle regulators.

Notch genes are a family of evolutionarily conserved genes that are involved in cell communication and play a crucial role in the development of various tissues and organs in animals, including humans (Artavanis-Tsakonas et al., 1999). Notch signals control how cells respond to intrinsic or extrinsic developmental cues (Artavanis-Tsakonas et al., 1999) and regulate the maintenance of undifferentiated cells. Hes1 and Hes5 are essential Notch effector genes (Ohtsuka et al., 1999). In the absence of Hes genes, cell differentiation is accelerated – members of the Hes family have been shown to play a key role in maintaining neural progenitors (Bonev et al., 2011). The timing of neural stem cell differentiation has important consequences for the size, shape and histogenesis of the nervous system, including the brain (Kicheva et al., 2014). In embryos lacking Hes1 and Hes5, as well as Hes3 which has a similar role (Hirata et al., 2000), neural stem cells prematurely differentiate into neurons, without generating the later-born cell types, such as oligodendrocytes, astrocytes and ependymal cells, and the brain structures are severely underdeveloped (Hatakeyama, 2004). Overexpression of Sox21, which directly represses Hes5, similarly promotes neuronal differentiation (Tsutsumi et al., 2012).

1.1.2 The role of noise in biological systems

Gene expression dynamics can now be analysed at single-cell resolution over time. Stochastic variation in a population of cells cannot be quantified using population-level measurements, and oscillatory gene expression in single cells cannot be captured using ‘static’ single-cell measurements, e.g. with flow cytometry or single-cell RNA sequencing (Longo & Hasty, 2006). New experimental techniques (see Section 2.2) have uncovered heterogeneous gene expression within populations of cells, and have demanded the development of new mathematical methods which draw from, for example, Markov chain theory, graph theory, Bayesian inference and stochastic calculus, in order to interpret this information (Ezer et al., 2016; Giuggioli & Neu, 2019; Gómez-Schiavon et al., 2017; Minas et al., 2017). Heterogeneity has been shown to have important implications for cell fate decisions for several cell types, ranging from stem cells to cancer cells (Loos, 2016; Michor & Polyak, 2010; Torres-Padilla & Chambers, 2014). So how does gene expression variability arise?

Cell-to-cell variability within clonal populations has been of great interest to biologists for many years (Spudich & Koshland, 1976), and *stochastic effects* have been invoked to explain this phenomenon (Swain et al., 2002). Stochastic effects, or noise, in gene expression, can be broadly classified into two types – *intrinsic* and *extrinsic*. These arise, loosely, from the internal system (regulatory network) or external sources (environment). Intrinsic noise occurs when only a few molecules of a specific type are present in a cell, leading for example to probabilistic binding and dissociation of a gene regulatory protein to and from its site on the DNA (Elowitz et al., 2002), or due to the random nature of a polymerase binding to a promoter (Dublanche et al., 2006). Extrinsic noise, in contrast, refers to external effects that will create differences between cells (Raser & O'Shea, 2005), such as differences in extracellular signalling.

The idea that noise plays an essential functional role in biological systems is not intuitive, since development in living organisms is precisely coordinated. However, cells can take advantage of the variation introduced by noise. For example, noise can allow oscillations to continue undamped (Alonso et al., 2007; Lewis, 2003). Theoretical work in a multi-cell model has predicted cooperative behaviours between cells induced by noise (L. Chen et al., 2005), and variations in the timing of cell differentiation enable cells in fluctuating environments to avoid prematurely committing to a specific fate (Eldar & Elowitz, 2010). Stochastic variation of the cell cycle has been shown to lead to robust proportions of cell differentiation (Gruenheit et al., 2018). Negative effects of noise, like the reduced function of essential proteins (A. Singh, 2011), can be mitigated by transcriptional (negative) auto-regulation (Dublanche et al., 2006). Negative feedback reduces the effects of noise since fluctuations in gene expression are continually pushed towards the mean (Raj & van Oudenaarden, 2008).

One potential consequence of stochastic effects is that the kinetic parameters of a cell, for example, rates of degradation or protein translation, may vary across a cell population. This may be caused by a number of factors, including differences in the concentration of transcription factors for upstream target genes, the location of a cell within the tissue, or post-transcriptional regulation through micro-RNAs.

1.1.3 Micro-RNAs also regulate gene expression

MicroRNAs are small noncoding RNAs, which regulate gene expression at the post-transcriptional level. In zebrafish neurogenesis, the micro-RNA miR-9 regulates Her6 and Her9 by increasing the rate of mRNA degradation and repressing protein translation (Coolen et al., 2012; Galant et al., 2016; Leucht et al., 2008; Soto et al., 2020). Dynamic changes in Her6 protein expression coincide with the onset of miR-9 expression in the hindbrain (Soto et al., 2020). Experimental work (Bonev et al., 2012) and mathematical modelling (Goodfellow et al., 2014; Phillips et al., 2016) have shown that the amount of miR-9 in a cell controls the timing of cell state transitions.

For example, a delay differential equation model of a double-negative feedback loop of miR-9 and Hes1 informed by experimental data (Bonev et al., 2012) exhibits both oscillations and bi-stability (Goodfellow et al., 2014). A key finding of this model is that the amount of active miR-9 provides a mechanism to tune the degradation rate of *Hes* mRNA, and controls the timing of transient oscillations. When this model is extended to account for intrinsic noise, timing to differentiation becomes distributed, and greater amounts of noise increase the uncertainty in this timing (Phillips et al., 2016). This result more accurately reflects experimental data, reinforcing the need to account for intrinsic noise.

Although the amount of miR-9 present is important, how the amount of miR-9 is controlled is not well understood. This question is difficult to address since zebrafish possess seven copies of the miR-9 gene at distinct loci, which are all capable of producing the same mature miR, and it is not clear how they cooperate to determine differentiation. In Chapter 5 it is shown that miR-9 increases in a sharp stepwise manner during zebrafish neurogenesis. We develop a mathematical model based on perfect adaptation and interactions between miR-9 and Her6 and suggest that the stepwise increase of miR-9 facilitates the robustness of cell state transitions in the presence of small-scale fluctuations.

1.2 Aims and objectives

Recent work in murine neural stem cells has identified differences in HES5 protein expression dynamics and hypothesises that these dynamics correlate with cell states (Manning et al., 2019). In particular, cells declining with oscillatory HES5 correlate with interneuron fate, whereas declining non-oscillatory cells correlate with motor neuron fate (Manning et al., 2019).

Understanding the driving forces behind these differences is essential to understanding cell fate decisions during the formation of the nervous system.

The mechanisms behind the oscillatory expression of Hes have previously been illuminated by Monk (2003). This model was subsequently extended to account for intrinsic noise, which enables us to accurately capture the observed dynamics (Section 2.3) (Barrio et al., 2006; Brett & Galla, 2013; Galla, 2009; Manning et al., 2019; Phillips et al., 2016). Parameter inference has been performed on models of Hes via direct numerical solutions of SDE's (Heron et al., 2007). However, this approach does not obtain a closed form likelihood, and therefore relies on sampling parameters conditional on a sampled path of the unobserved states, which is both high-dimensional and inefficient due to strongly correlated trajectories (Calderazzo et al., 2019; Golightly & Wilkinson, 2011). Performing parameter inference on these models could enable the identification of many kinetic parameters of gene expression, such as the rates of transcription, translation, and degradation without performing further experiments. However, methods for parameter inference in this model have not been widely developed. In order to address this, this thesis has the following aims and objectives.

1.2.1 Developing a method for parameter inference in stochastic delayed gene regulation systems (Chapter 3)

A method for parameter inference in stochastic delayed systems has been proposed by Calderazzo et al. (2019), where the authors develop a novel filtering approach to infer parameters from time-series data. We extend this approach in two key ways. We apply this new approach to a widely applicable motif of auto-repression which includes both mRNA and protein. We also extend this approach by implementing a gradient-based sampling method for efficient and accurate inference. Our method enables us to determine specific kinetic parameters of a cell, such as rates of degradation, translation and transcription, given noisy time series data. This work was published in (Burton et al., 2021).

1.2.2 Investigating the influence of data on uncertainty (Chapter 3)

The design of imaging experiments may impact information acquisition, but this relationship has not been well characterised for the auto-negative feedback motif. We explore how parameter uncertainty in our inference method changes when key experimental changes are made, for example how long a cell is imaged for. We also show how our approach can be easily extended to account for multiple data types from additional experiments. This work was published in (Burton et al., 2021).

1.2.3 Scaling up inference to larger datasets (Chapter 4)

Parameter uncertainty may be too high to distinguish cells with qualitatively different protein expression dynamics effectively. To address this, cells can instead be clustered together based on their gene expression, and data from multiple cells can be combined to reduce parameter uncertainty. However, this leads to inference which is restrictively time-consuming. We improve the performance of our Kalman filtering approach, achieving a speed-up of multiple orders of magnitude. Combining our improved method with variational inference enables accurate parameter estimation when the amount of data increases.

1.2.4 Using data to inform model building and exploration (Chapter 5)

Her6/Hes5 dynamics do not only depend on internal parameters of the auto-repressive feedback motif but can also be tuned by external factors. A known external factor modulating Her6 and Hes5 dynamics is the microRNA miR-9. How miR-9 concentrations vary over time, and how these variations affect Her6 dynamics remains an open question. In a joint cross-disciplinary investigation with experimental collaborators, my colleagues show that the amount of mature miR-9, a key regulator of neuronal development, increases during zebrafish neurogenesis in a sharp stepwise manner. To illuminate the role of this stepwise increase, I develop a mathematical model which shows that an adaptive network containing Her6 is insensitive to linear increases in miR-9 but responds to stepwise increases of miR-9, revealing a potential mechanism for robust cell state transitions in the presence of noise. This work was published in (Soto et al., 2022).

Chapter 2

Technical Background

In order to achieve our aims, we will first need to equip ourselves with the necessary tools to describe and pursue an answer to our research question. First, we will look at how statistical models can be built and fit to data. Then we will review experimental techniques for measuring gene expression, and the type of data these experiments gather. Lastly, we will iteratively introduce a mathematical model for gene expression driven by delayed negative feedback, and present an overview of how this model can be used to interpret time-series data.

2.1 Bayesian inference for mechanistic models

We aim to use a mathematical model to infer the kinetic parameters of single cells, using gene expression time-series data. The success of this approach will rely upon how well our model captures the true data-generating process, as well as how informative the collected data is. In order to illustrate the philosophy of our approach, we first consider a simple example. A similar example can be found in Professor Simon Cotter's Introduction to Uncertainty Quantification lecture notes.

2.1.1 Heads or tails?

Consider the following scenario:

> *Alice and Bob have lunch together every day. They each have a favourite place so to decide where to eat, Alice flips a coin. If it lands on heads, they eat at Alice's favourite place, if it's tails they eat at Bob's. Recently, Bob has become suspicious that this coin isn't fair. In the last 30 days, the coin has landed on heads 22 times. How can Bob determine how likely it is that this coin is fair or not?*

Example 2.1.1 (Heads or tails?) Let's suppose that the probability of the coin landing on heads is given by $p \in [0, 1]$. In other words,

$$\begin{aligned} P(\text{heads}) &= p, \\ P(\text{tails}) &= 1 - p. \end{aligned}$$

One approach we could take is to approximate p using the proportion of heads out of the total number of flips, in this case, $22/30 \approx 0.73$. But consider if the coin was flipped three times and only one of them was heads. Would we be happy to approximate p with 1/3? We don't have enough information to be confident in this approximation, so we need some way of quantifying this uncertainty. We also want to be able to account for our prior knowledge that most coins we have come across in the past are fair.

One way to approach this is to use *Bayesian* inference. Bayesian inference provides us with a framework to use data to update our pre-existing beliefs about the likelihood of an event, rather than characterising a hypothesis that we seek to prove or disprove. In this view, the question is not "is this coin fair?", but rather, "how likely is it that this coin is fair?".

We start by using our knowledge and past experiences to construct a *prior* distribution $p(\theta)$ which gives our degree of belief that the unknown value p is equal to θ for each $\theta \in [0, 1]$. In this example, we want a distribution defined over the range $[0, 1]$. A common choice is the Beta distribution, whose probability density function is defined by

$$f_X(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\mathrm{B}(\alpha, \beta)}, \tag{2.1}$$

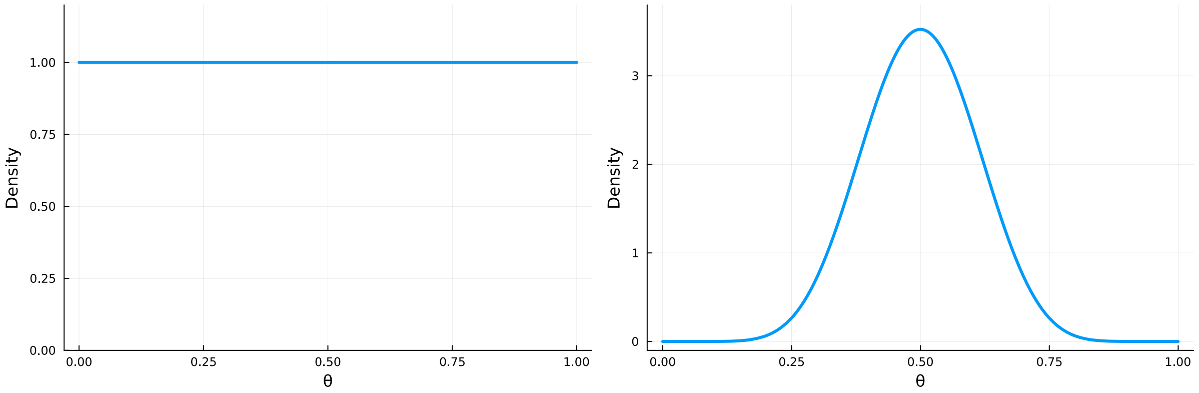
where α, $\beta \in \mathbb{R}^{>0}$, $x \in [0, 1]$, $\mathrm{B}(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$ and Γ is the *Gamma* function.

We could choose $\text{Beta}(1, 1)$, which is uniform over the range $[0, 1]$, if we think that p could take any value in $[0, 1]$ with equal probability. We might have a strong suspicion that the coin is fair since most coins are (almost (Diaconis et al., 2007)), and instead, choose $p(\theta) = \text{Beta}(10, 10)$. Figure 2.1 shows the probability density functions of these two distributions. Suppose Bob wants to give Alice the benefit of the doubt, so he chooses to use $\text{Beta}(10, 10)$ as his prior.

Next, we update this prior distribution to construct the *posterior probability distribution* of θ given the data x,

$$p(\theta \mid x) = \frac{p(x \mid \theta)p(\theta)}{p(x)} \propto p(x \mid \theta)p(\theta). \tag{2.2}$$

This function describes the probability of θ conditioned on some data. It will allow us to determine the most likely values of our parameters, and how much more likely they are than other values. The denominator $p(x)$ is the marginal likelihood of the data and is a normalis-



(a)

(b)

Figure 2.1: Probability density functions for (a) Beta$(1, 1)$ and (b) Beta$(10, 10)$

ing constant defined by

$$p(x) = \int_{\Theta} p(x \mid \theta) p(\theta) \mathrm{d}\theta, \quad \theta \in \Theta, \tag{2.3}$$

which is typically non-trivial to calculate.

The mechanism for updating our prior beliefs is the *likelihood* function $p(x \mid \theta)$. This function tells us how the chance of obtaining the observed data changes as θ varies. In the case of our coinflip example, this is given by

$$p(x \mid \theta) = \binom{30}{22} \theta^{22} (1 - \theta)^{8}, \tag{2.4}$$

where $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ is the binomial coefficient.

Putting the prior and likelihood together, we get

$$\begin{align} p(\theta \mid x) &\propto p(x \mid \theta) \times p(\theta) \tag{2.5} \\ &= \binom{30}{22} \theta^{22} (1 - \theta)^{8} \times \frac{\theta^{9}(1 - \theta)^{9}}{\mathrm{B}(10, 10)} \tag{2.6} \\ &\propto \theta^{31} (1 - \theta)^{17}, \tag{2.7} \end{align}$$

which is proportional to Beta$(32, 18)$.

Figure 2.2 shows the prior and posterior distributions for this example. The posterior distribution has a mean of 0.64 and a standard deviation of 0.067, and 95% of the distribution is contained in the interval $[0.504, 0.766]$ (see Figure 2.3). Given the available data, it seems unlikely that $p = 0.5$, and it's nearly 8 times more likely that the probability of getting heads is 0.64. In this case, Bob might have a word with Alice and suggest they use a new coin.

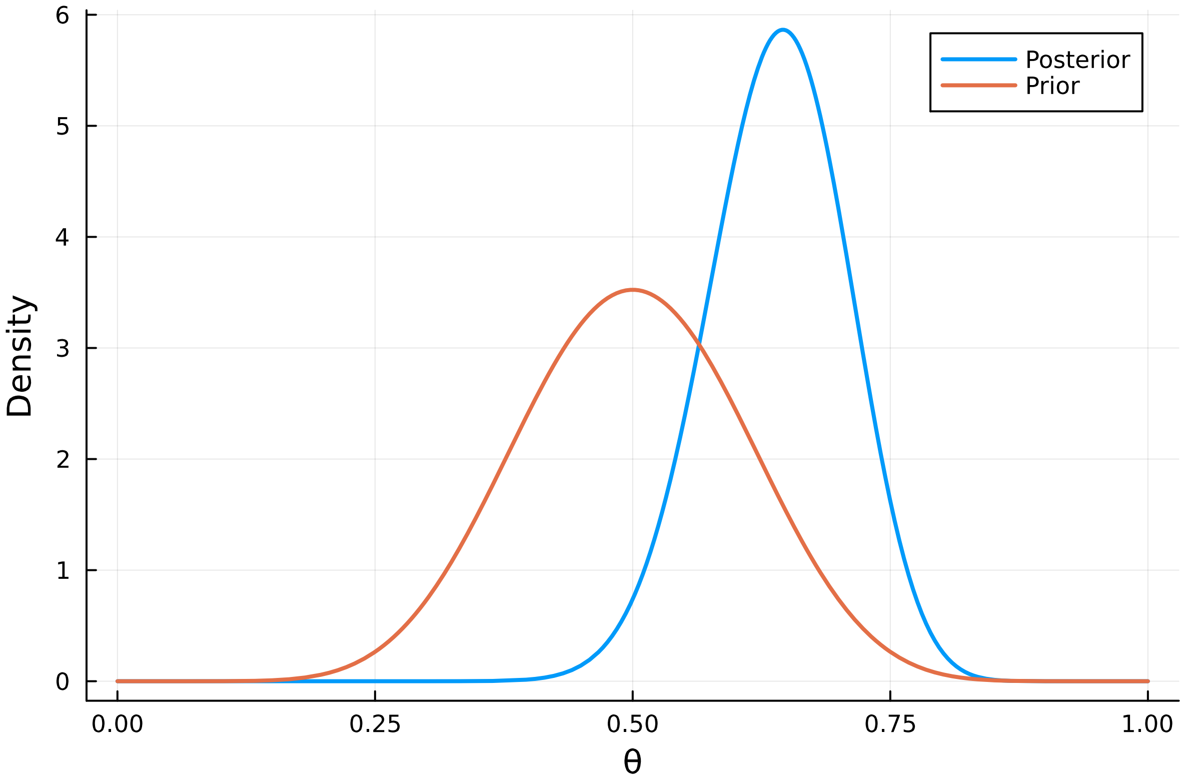


Figure 2.2: The $\text{Beta}(10, 10)$ prior and $\text{Beta}(32, 18)$ posterior distributions.

For Example 2.1.1 the posterior is given by $\text{Beta}(\alpha+n_{\text{H}}, \beta+n_{\text{T}})$, where the prior is $\text{Beta}(\alpha, \beta)$ and n_{H}, n_{T} are the number of heads and tails respectively. If we consider again the scenario where the coin is flipped three times and only one is heads, with a $\text{Beta}(10, 10)$ prior the posterior distribution would be $\text{Beta}(11, 12)$, which has a mean of $\approx\ 0.478$. With the weaker prior $\text{Beta}(1, 1)$, the posterior is $\text{Beta}(2, 4)$. This distribution has a mean of 1/3, however, the standard deviation is 0.178, indicating a lack of certainty. In other words, the Bayesian approach allows us to account for the fact that we have limited information, and this will be reflected in any predictions we make. To more clearly visualise this uncertainty, Figure 2.4 shows the progression of the posterior distribution as data is added in the sequence tails, heads, tails.

2.1.2 Posterior distributions can be obtained via Monte Carlo sampling

In Example 2.1.1, our prior and posterior distribution have the same analytical form, as they are both Beta distributions. When the prior and posterior distribution have corresponding analytical expressions in this way, we refer to this as conjugacy, and say that the prior and posterior are *conjugate* distributions. Example 2.1.1 used a conjugate prior to provide us with posteriors which are well-known distributions. In real applications, this is typically not the case, and posterior distributions won't take an analytical form. How can we find the posterior distri-

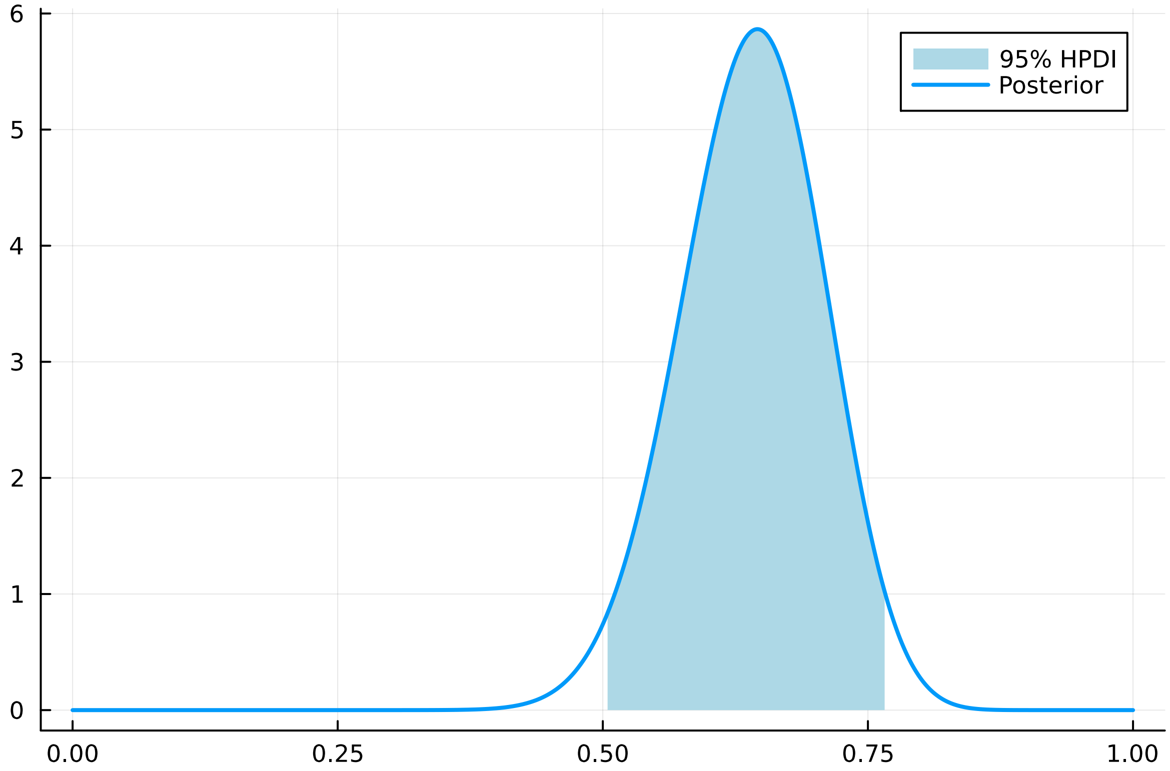


Figure 2.3: The 95% highest posterior density interval for the $\text{Beta}(32, 18)$ posterior distribution.

bution in these cases? There are a number of different approaches, but here we will focus on Markov chain Monte Carlo (MCMC) sampling.

A (discrete-time) Markov chain is a sequence of random variables $X_1, X_2, X_3, \dots$ that satisfy the Markov property,

$$P(X_{n+1} = x_{n+1} \mid X_n = x_n, X_{n-1} = x_{n-1}, \dots, X_1 = x_1) = P(X_{n+1} = x_{n+1} \mid X_n = x_n). \tag{2.8}$$

This is also referred to as *memorylessness* and says that future outcomes are only dependent on the current state of the Markov chain, and are not dependent on past states, conditional on knowing the previous state. For the purposes of this illustration, we restrict the X_i to discrete values, but the following can be extended to general state spaces (see, for example, Chapter 4 of Gilks et al. (1995)).

If there exists some unique collection of probabilities $\{\pi_j\}$ such that for all i,

$$\sum_i \pi_j P_{ij} = \pi_j$$

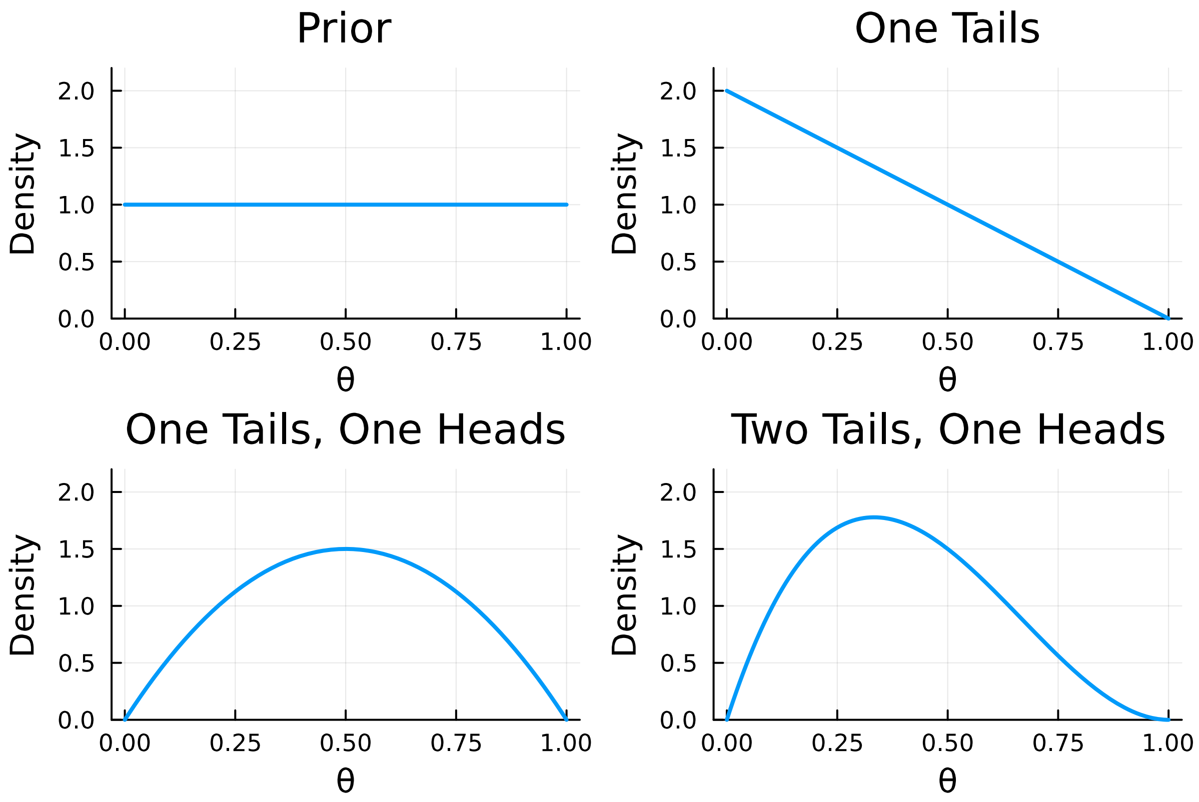


Figure 2.4: The evolution of the posterior distribution as the coin is flipped multiple times, starting with a uniform prior.

and

$$\lim_{n \to \infty} P(X_n = j \mid X_0 = i) = \pi_j,$$

where P_{ij} defines the probability of going from state i to state j, $P(X_{n+1} = j \mid X_n = i)$, then this set of probabilities defines the distribution π, which we call the stationary distribution of the Markov chain. The existence of a stationary distribution is dependent on certain properties (discussed in Chapter 3 of Gilks et al. (1995)).

Example 2.1.2 (Coinflip) Suppose we are flipping a coin, which has an independent probability $p = 0.25$ of landing on heads each time it is flipped. If we define state one as heads and state two as tails, then the *transition probability matrix* is given by

$$P = \begin{bmatrix} 0.25 & 0.75 \\ 0.25 & 0.75 \end{bmatrix}.$$

P_{ij} defines the probability of moving from state i to state j. For example, P_{12} is the probability of the coin landing on tails given that it previously landed on heads. Since these events are independent, this is simply given by the probability of landing on tails, 0.75. These states are represented in Figure 2.5. In practice, Markov chains are not required to solve this example problem. Here, we use it to illustrate how Markov chain theory can be used.

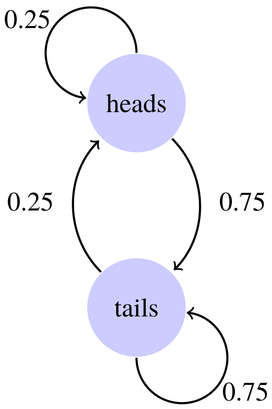


Figure 2.5: Diagram representing the states of the Markov chain for a simple coinflip problem, where the probability of landing on heads is 0.25. The labels represent the transition probabilities.

MCMC describes a class of methods which output a Markov chain whose stationary distribution is our target distribution, $p(\theta \mid x)$. The key is to construct an algorithm that will efficiently locate areas of high posterior probability mass within our pre-defined parameter space. The most common algorithm for doing this is Metropolis-Hastings (Algorithm 1), The simplest version of Metropolis-Hastings is the random walk Metropolis (RWM), which is defined as follows. Starting from an initial set of parameters θ_0, the general procedure is that at step i of the algorithm, a set of parameters θ' are proposed from a distribution q conditioned on the current location, θ_i. In order to determine whether the proposed sample is within (or closer to) an area of high posterior density, an acceptance probability is computed. The acceptance probability

$$\alpha(\theta_i, \theta') = \min\left\{\frac{p(\theta' \mid y)q(\theta' \mid \theta_i)}{p(y \mid \theta_i)q(\theta_i \mid \theta')}, 1\right\} = \min\left\{\frac{p(y \mid \theta')p(\theta')q(\theta' \mid \theta_i)}{p(\theta_i \mid y)p(\theta_i)q(\theta_i \mid \theta')}, 1\right\}$$

is the largest value of $[0, 1]$ given the choice of proposal distribution q which ensures that the stationary distribution of the Markov chain is given by $p(\theta \mid y)$. By construction, this algorithm outputs a Markov chain, and in the limit of infinite samples will converge to the stationary (posterior) distribution.

Random Walk Metropolis-Hastings

A common choice for q is the multivariate normal distribution whose mean is the current parameters and whose variance is given by $h\Sigma$. In other words, at step i of the algorithm, we have

$$\theta' \sim \text{MvNormal}\left(\theta_i, h\Sigma\right), \tag{2.9}$$

Algorithm 1: Metropolis-Hastings Algorithm

```
Input: initial value $\theta_0$, proposal distribution $q$, data $y$, number of iterations $N$
Output: $\theta_1, \theta_2, \dots, \theta_N$
for $i = 0, 1 \dots, N$ do
    sample: $\theta' \sim q(\theta' \mid \theta_i)$
    compute: $\alpha(\theta_i, \theta') \leftarrow \min\left\{\frac{p(\theta'|y)q(\theta'|\theta_i)}{p(y|\theta_i)q(\theta_i|\theta')}, 1\right\} = \min\left\{\frac{p(y|\theta')p(\theta')q(\theta'|\theta_i)}{p(\theta_i|y)p(\theta_i)q(\theta_i|\theta')}, 1\right\}$
    $u \leftarrow$ Uniform$(0, 1)$
    if $u < \alpha(\theta_i, \theta')$ then
        accept the proposal: $\theta_{i+1} \leftarrow \theta'$
    else
        reject the proposal: $\theta_{i+1} \leftarrow \theta_i$
    end
end
```

or equivalently

$$\theta' = \theta_i + \sqrt{h}\Sigma^{1/2}\xi \tag{2.10}$$

where $\xi \sim$ MvNormal$(0, I_d)$, d is the number of parameters, and I_d is the $d-$dimensional identity matrix. This algorithm is an attractive choice due to its simplicity and ease of implementation. For example, Listing 1 shows a Julia implementation of the random walk algorithm to sample from a standard normal distribution.

Crucially, the performance of the random walk depends on the step size, h, and the proposal covariance, Σ. If the step size is too large, areas of high probability may be missed entirely, and if it is too small it will take a large number of steps before areas of high probability are discovered. Figure 2.6 shows the Markov chains for 100 iterations of the random walk algorithm for various step sizes, initialised at $\theta = 0$. The step size can be manually tuned to achieve an optimal acceptance rate of ≈ 0.234[1]. In Figure 2.6, the acceptance rates for $h = 0.1, 20,$ and 200 are 0.90, 0.26, and 0.089 respectively (setting $\Sigma = 1$).

Metropolis-adjusted Langevin algorithm

The Metropolis-adjusted Langevin algorithm (MALA) is similar to the random walk Metropolis but has the proposal distribution

$$\theta' \sim \text{MvNormal}\left(\theta_i + \left(\frac{h\Sigma}{2}\right)\nabla \log p(\theta_i \mid x), h\Sigma\right), \tag{2.11}$$

where $\nabla \log p(\theta_i \mid x)$ is the gradient of the log-posterior at θ_i.

The MALA proposal is derived from a first order Euler-Maruyama approximation to the Langevin

[1] optimal under a set of conditions listed in Roberts and Rosenthal (2001)

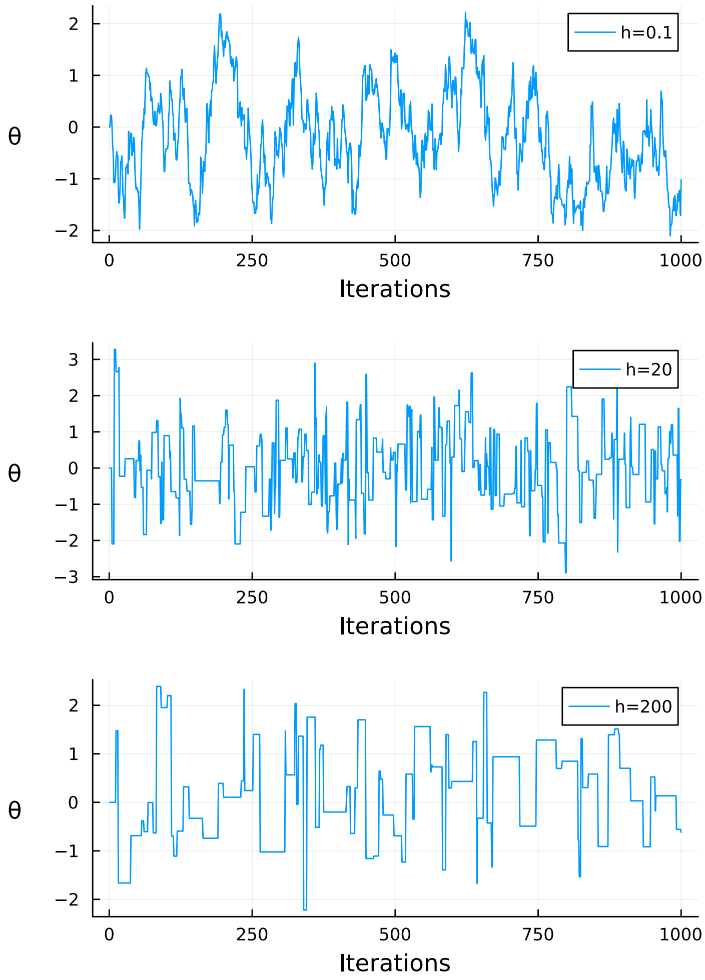


Figure 2.6: Markov chains sampling from a standard normal distribution using the random walk Metropolis algorithm with various step sizes. When h is small, consecutive samples are close to each other and highly correlated, and when h is large, consecutive samples can be far away from each other, but proposals are rejected more frequently.

diffusion SDE defined by

$$\mathrm{d}\theta(t) = \frac{1}{2}\nabla \log p(\theta(t) \mid x)\mathrm{d}t + \mathrm{d}B(t), \tag{2.12}$$

whose stationary distribution is given by $p(\theta \mid x)$, where $B(t)$ is a d-dimensional Brownian motion and $\theta \in \mathbb{R}^d$ (Roberts & Tweedie, 1996).

The proposal here tends to move the chain in the direction of $\nabla \log p(\theta_i \mid x)$, i.e. towards larger values of the target distribution p, which helps the chain to converge more quickly. The optimal acceptance rate for MALA is 0.574 (Roberts & Rosenthal, 2001), which means fewer overall proposals are needed. The gradient needs to be computed at each step, which will increase the per-iteration computation time. Listing 2 shows a Julia implementation of MALA to sample from a standard normal distribution.

```julia
using Distributions

function random_walk_normal(N, h; init=0)
    theta = zeros(N)
    theta[1] = init
    for i in 2:N
        u = rand(Uniform())
        theta_prop = theta[i-1] + sqrt(h)*rand(Normal())
        alpha = min(1, logpdf(Normal(), theta_prop) - logpdf(Normal(), theta[i-1]))
        if log(u) < alpha
            theta[i] = theta_prop
        else
            theta[i] = theta[i-1]
        end
    end
    return theta
end
```

Listing 1. A Julia function to sample from the standard normal distribution using the random walk Metropolis algorithm.

The effect of the gradient in the proposal distribution can be easily seen when comparing RWM and MALA with an initialisation that is far from the mode, even when using the same values for h and Σ (Figure 2.7). The gradient drives the proposal toward the mode of the distribution, meaning the Markov chain is in an area of high probability much more quickly than the random walk algorithm ($\sim$10 iterations with MALA compared to $\sim$60 with RWM). Further, consecutive samples become uncorrelated much more quickly (Figure 2.8).

In higher dimensions, this difference in performance is amplified, as the search space becomes unfeasibly large for the 'guess-and-check' strategy of RWM (Betancourt, 2017). Using information about the geometry of the posterior distribution exponentially reduces the number of proposals we have to check as the dimension increases (Durmus et al., 2017).

```julia
using Distributions

function mala_normal(N, h; init=0)
    theta = zeros(N)
    theta[1] = init
    for i in 2:N
        u = rand(Uniform())
        theta_prop = theta[i-1] + (-theta[i-1]*h/2) + sqrt(h)*rand(Normal())
        alpha = min(1, logpdf(Normal(), theta_prop) - logpdf(Normal(), theta[i-1]))
        if log(u) < alpha
            theta[i] = theta_prop
        else
            theta[i] = theta[i-1]
        end
    end
    return theta
end
```

Listing 2. A Julia function to sample from the standard normal distribution using the Metropolis-adjusted Langevin algorithm.

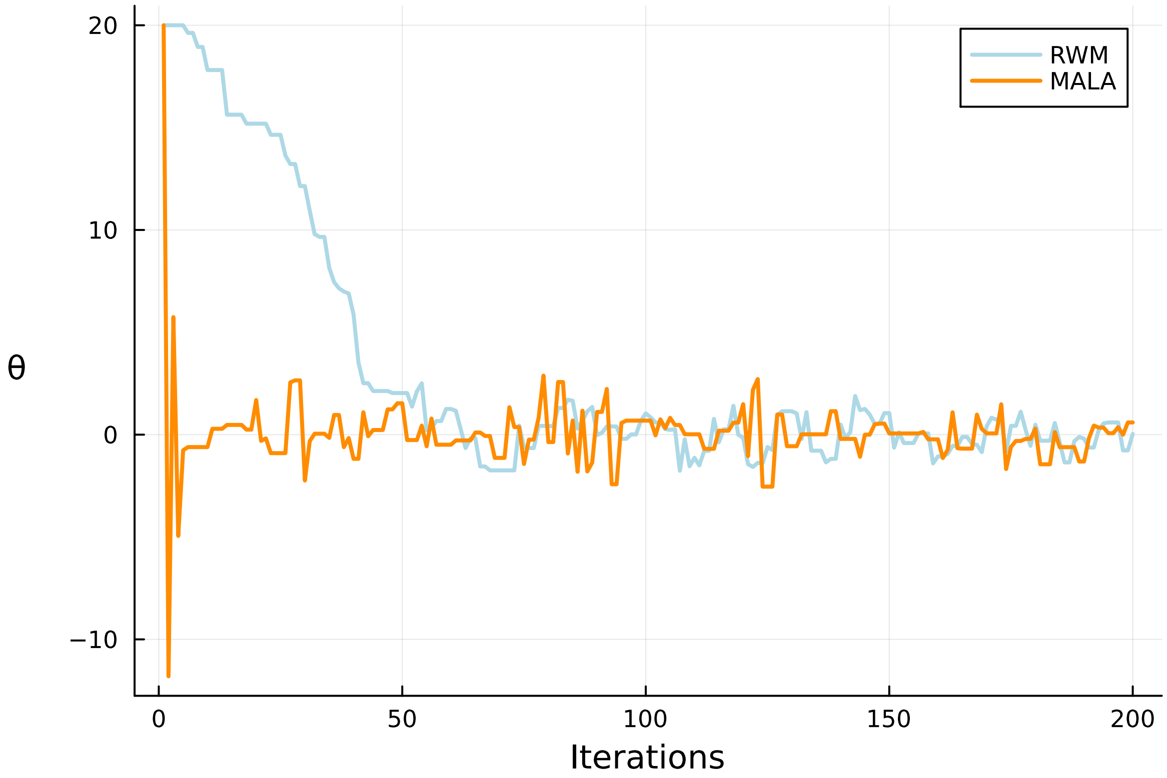


Figure 2.7: A comparison of the RWM and MALA algorithms sampling from a standard normal distribution, initialised at $\theta = 20$. The Markov chain produced by MALA converges on the high probability mass of the distribution almost immediately, compared to the slow approach made by the RWM algorithm.

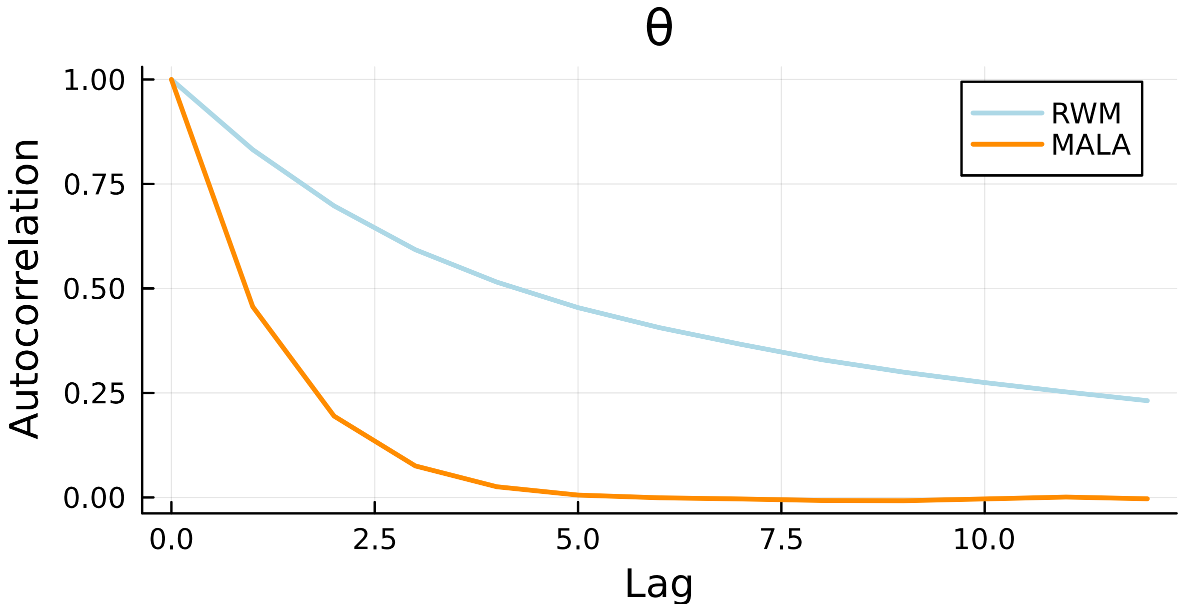


Figure 2.8: A comparison of the autocorrelation of the Markov chains from both the RWM and MALA algorithms, shown in Figure 2.7. Consecutive samples obtained using RWM are highly correlated, meaning we cannot treat them as independent samples. On the other hand, consecutive samples from MALA become uncorrelated after a lag time of 5.

Hamiltonian Monte Carlo

State-of-the-art MCMC samplers used by modern probabilistic programming languages (Ge et al., 2018; Salvatier et al., 2016; Stan Development Team, 2023) are based on Hamiltonian Monte Carlo (HMC) (Duane et al., 1987).

In HMC, the target distribution is reformulated as the *potential energy* of a particle in a mathematically constructed physical system (Betancourt, 2017). This physical system is constructed in such a way that ensures conservative dynamics. In other words, if our parameters θ are considered as particles, q, in the physical system, with auxiliary momentum variable p which conserves the dynamics of the system, then the Hamiltonian of this system is defined by

$$\begin{aligned} H(q,p) &\equiv -\log \pi(q,p) \\ &\equiv -\log \pi(p \mid q) - \log \pi(q) \\ &\equiv K(p,q) + V(q), \end{aligned}$$

where $\pi(q)$ is the density of the target distribution. We can then generate trajectories over the state space by solving Hamilton's equations,

$$\begin{aligned} \frac{\mathrm{d}q}{\mathrm{d}t} &= \frac{\partial K}{\partial p}, \\ \frac{\mathrm{d}p}{\mathrm{d}t} &= -\frac{\partial K}{\partial q} - \frac{\partial V}{\partial q}, \end{aligned}$$

to obtain a proposal for the next sample in the Markov chain.

In practice, these equations are solved with *symplectic integrators*, which are configured by two parameters, a step size ϵ and an order, K. The standard choice is the leapfrog integrator (Leimkuhler & Reich, 2005). The effectiveness of HMC is highly dependent on the choice of algorithm parameters, such as the step size and the number of steps in the leapfrog integrator, and the covariance of the proposal distribution. The No-U-Turn sampler (NUTS) is a variant of HMC that avoids the necessity of manually tuning these parameters while generating independent samples at least as efficiently as standard HMC (M. D. Hoffman & Gelman, 2014). Modern probabilistic programming languages like Turing (Ge et al., 2018) also allow users to easily define and sample from probabilistic models. For example, Listing 3 shows code which allows a user to sample from a standard normal distribution with NUTS using Turing in the Julia programming language.

Example 2.1.3 (Coinflips with Turing) We can use MCMC directly to estimate the probability p of flipping heads, given our input data of 22 heads out of 30 flips. Listing 4 shows

```julia
using Turing

@model function turing_normal()
    y ~ Normal(0,1)
end

chain = sample(turing_normal(), NUTS(), 2_000)
```

Listing 3. Code to obtain 2000 standard normal samples using NUTS via Turing.

the Julia code to produce 2000 samples from the posterior distribution. Figure 2.9 shows a plot of the Markov chain produced by this code, as well as the density of this Markov chain alongside the Beta$(32, 18)$ distribution. The mean and standard deviation of the chain are 0.64 and 0.067 respectively, with a 95% HPDI of $[0.506, 0.766]$, which aligns very closely with the Beta$(32, 18)$ distribution, as expected.

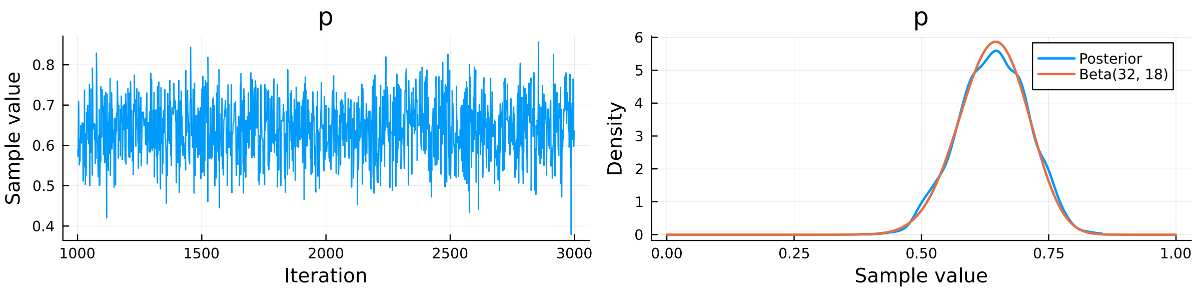
```julia
using Turing

data = vcat(zeros(8), ones(22))

@model function coinflip(y)
    p ~ Beta(10,10)
    for i in 1:length(y)
        y[i] ~ Bernoulli(p)
    end
end

model = coinflip(data)
chain = sample(model, NUTS(), 2_000)
```

Listing 4. A simple coin flip example using Turing



(a)

(b)

Figure 2.9: **Computational MCMC results align with our theoretical derivation of the coin flipping example posterior. A.** The Markov chain for the parameter p, produced using the NUTS sampler in Turing.jl. **B.** The kernel density of the Markov chain (blue) is shown together with the density of the Beta$(32, 18)$ distribution.

MCMC Diagnostics

The target posterior distribution is typically unknown, and MCMC is only guaranteed to converge to the posterior as the number of samples tends to infinity. We may converge much sooner than this, and a number of statistics have been developed to test for convergence of our Markov chains.

Best practice when performing MCMC is to compute multiple (usually 4 or more) Markov chains, in parallel. How well the chains are exploring the posterior can be assessed with the $\hat{R}$ statistic, which evaluates the ratio of within- and between-chain variance (Vehtari et al., 2020). Another measure of convergence is the effective sample size (ESS), which approximates the true number of independent and identically distributed samples from the posterior distribution that the Markov chain contains.

Current recommendations for practice suggest that users should compute at least four chains, and only use the sample if $\hat{R} < 1.01$ and the ESS is greater than 400 for each parameter (Vehtari et al., 2020).

Example 2.1.4 (Diagnostics in practice) MCMC diagnostics are implemented in Turing and are provided to the user alongside the posterior samples. Furthermore, parallelization is easily achieved with minimal changes to the code. Listing 5 shows the Julia code to produce 8000 samples from the posterior distribution of our coin-flipping example across 4 chains in parallel. Table 2.1 reports on various statistics computed internally, which can be used to diagnose any problems in the inference.

```julia
using Turing

data = vcat(zeros(8), ones(22))

@model function coinflip(y)
    p ~ Beta(10,10)
    for i in 1:length(y)
        y[i] ~ Bernoulli(p)
    end
end

model = coinflip(data)
chain = sample(model, NUTS(), MCMCThreads(), 2_000, 4)
```

Listing 5. A simple coin flip example using Turing

Parameter	**Mean**	**SD**	$\hat{R}$	**ESS**
p	0.6381	0.0666	1.0004	4491

Table 2.1: Summary and diagnostic statistics of the Markov chain computed in Listing 5.

Using Bayes theorem (eq. (2.2)), we are able to define the posterior distribution of the parameters of our model, given some observed data. MCMC algorithms allow us to generate correlated samples from this distribution. Biological experiments produce a variety of types of data, which each require bespoke likelihood functions. In the following section, we discuss a variety of biological experiments that measure gene expression in single cells.

2.2 Gene expression can be measured with a variety of techniques

Heterogeneity of gene expression within a population of cells of the same cell type is ubiquitous (Huang, 2009). In the past, population-averaged data has been obtained from microarrays or Western blots. Now, techniques such as flow cytometry (Pyne et al., 2009) and fluorescent microscopy (Schroeder, 2011) provide data which is more information-rich. Typically, gene expression heterogeneity is investigated by interrogating the gene expression variability across a cell population at fixed points. Increasing amounts of evidence suggest that dynamic patterns of gene expression can also show heterogeneity and vary between cell populations (Manning et al., 2019; Soto et al., 2019). There are multiple experimental methods for obtaining gene expression data from cells (e.g. protein copy number), and here the advantages and implications of each one are discussed.

2.2.1 *In situ* Hybridization

In situ hybridization (ISH) is a technique used to study gene expression and involves a hybridization reaction between a labelled nucleotide probe and complementary target DNA or RNA sequences (Jin & Lloyd, 1997). Developed in 1969, traditionally these hybrids were detected by autoradiographic emulsion for radioactively labelled probes or by histochemical chromogen development, but this was replaced in 1977 with indirect immunofluorescence, which quickly gained popularity (Jin & Lloyd, 1997; Rudkin & Stollar, 1977). The hybridization signals in fluorescent *in situ* hybridization (FISH) can be visualised by fluorescent microscopy. Typically, cells have to be fixed, meaning that time-series data is not obtainable and instead snapshot data is collected. A number of mathematical methods for analysing and inferring information from snapshot data have been developed and are discussed in Section 2.3.1. An important development for FISH came in the form of single-molecule FISH, which is able to localize and quantify individual RNA molecules (Femino et al., 1998; Raj et al., 2008). In Chapter 5, *in situ* hybridization techniques are used to investigate miR-9 expression during early zebrafish development.

2.2.2 Single-cell RNA sequencing

Single-cell RNA sequencing (scRNAseq) provides the mRNA expression profiles for individual cells by isolating them and following a number of steps, such as reverse transcription and RNA amplification. Recent methods encapsulate single cells in droplets in a microfluidic device. These droplets carry a DNA ‘barcode’ which uniquely labels the information derived from the cell (Klein et al., 2015; Macosko et al., 2015). This provides a snapshot mRNA measurement like with smFISH but avoids the invasive cell fixing process which may perturb the cell state. It also allows for much higher throughput, typically thousands of cells.

2.2.3 Flow Cytometry

Flow cytometry is a powerful tool for the analysis of multiple individual cell parameters from heterogeneous populations (Picot et al., 2012). Thousands of cells per second can pass one by one through one or more laser beams, which measure scattered light and fluorescence emissions. Cells can be sorted into sub-populations by measured parameters, for example, protein expression. Flow cytometry can indicate the amount of protein expressed in a single cell on the basis of the intensity of fluorescence (Butts & Sternberg, 2009). This leads to snapshot data (see Section 2.3.1) which can be analysed using software platforms such as Cell Quest (BD Biosciences) and FlowJo (Tree Star, Inc.). These packages generate statistics on the data, such as percentiles, mean numbers, and mean fluorescent intensity values for a sample (Butts & Sternberg, 2009).

2.2.4 Endogenous gene knock-in

Another way to observe gene expression is by gene knock-in, or using a gene editing technology such as CRISPR-Cas9 (Shen et al., 2014). These techniques involve either substitution of DNA sequence information or the insertion of extra information not found within a genetic locus. For example, a gene's DNA sequence might be extended so that a fluorophore will be attached to the translated protein. In this case, whenever the gene of interest is expressed so is the added protein, and endogenous gene expression dynamics can be measured using fluorescent imaging. Absolute molecule numbers can then be studied with fluorescence correlation spectroscopy (FCS). This tool allows us to translate fluorescence intensity into protein concentrations by analysing the diffusion properties of fluorescent molecules in a sub-femtoliter volume (Politi et al., 2018). Hence it is possible to obtain protein copy numbers in a single cell over time, with some level of error ($\sim$ 0.25–6% (Politi et al., 2018)). This data contains dynamic information at single-cell resolution. However, a single cell time series, although information-dense, may only consist of ~50–100 data points. In contrast, a snapshot from a population of thousands of cells produces a much greater amount of data. Work must be done to indicate the amount of data necessary to make sufficiently good inferences to inform experimental procedure, which we explore in Chapter 3.

2.3 Modelling gene expression and inference techniques

Here, we introduce two main types of data, snapshot and time series, and discuss approaches for interpreting them in the context of a mathematical model. Then, we introduce the auto-repressive feedback model which we will use to interpret single-cell gene expression.

2.3.1 Theoretical approaches to understanding gene expression are informed by data type

Approaches to understanding dynamic gene expression are dictated by the type of data available. As more advanced experimental techniques are developed we become able to observe dynamic gene activity with finer resolution, greater accuracy, and longer time periods. This makes it feasible to fit mathematical models to these data, which enables us to gain further mechanistic insights and validate these through experimental predictions. Parameter inference may give access to difficult-to-measure quantities and help pinpoint differences between cell populations.

In recent years, researchers have started to develop Bayesian inference techniques that can be used to identify molecular underpinnings of gene expression variation. However, current methods have limitations. One type of method focuses on population-level heterogeneity in snapshot data, thus ignoring dynamic properties of gene expression at the single cell level (Gómez-Schiavon et al., 2017). A second type of method analyses protein expression time-series data while ignoring non-linear effects of negative feedback (Minas et al., 2017). Initial efforts for inference in systems including non-linear interactions as well as delays exist, however, they have not been applied to describe protein expression dynamics (Calderazzo et al., 2019).

Snapshot data

Single-cell snapshot data can be represented mathematically by $\mathcal{D} = \{\{y_j(t_k)\}_j\}_{k=1}^{n_t}$, where n_t are the number of snapshot time points, and $y_j(t_k)$ gives the value being measured, for example, fluorescence intensity or amount of mRNA, for cell j at time point t_k. In particular, cells are not tracked over time, so even if there are multiple time points it is only possible to analyse heterogeneity at the population level. Due to the heterogeneity across populations, it may be possible to infer underlying mechanisms if additional biophysical constraints are imposed (Weinreb et al., 2018). Using population-averaged data, cellular heterogeneity can not be captured and sub-populations remain concealed (Altschuler & Wu, 2010).

Parameter inference with time-series data

Time-series data takes the form $\mathcal{D} = \{\{y_j(t_k)\}_{k=1}^{n_t}\}_j$. The distinction here from snapshot data is that individual cells are tracked over multiple time points, giving dynamic gene expression information at single-cell resolution.

Some mathematical and technological advancements for inferring model parameters using time-series data have been made (Minas et al., 2017). The basic model of gene expression upon which these efforts are built is given by

$$\frac{\mathrm{d}m(t)}{\mathrm{d}t} = \tau(t) - \mu_m m(t), \tag{2.13}$$

$$\frac{\mathrm{d}p(t)}{\mathrm{d}t} = \alpha_p m(t) - \mu_p p(t), \tag{2.14}$$

where $m(t)$ is the amount of mRNA transcript at time t, $p(t)$ is the amount of protein at time t, $\tau(t)$ is a non-negative function that describes transcription, α_p is the rate of translation, and μ_m, μ_p are the mRNA and protein degradation rates respectively (Finkenstädt et al., 2013). If we let $y_p(t_i), i = 1, \dots, n$, be the discretely observed imaging protein time series, then assuming the mean of the data is proportional with unknown factor $\kappa > 0$ to the concentration of the reporter protein, we have

$$y_p(t_i) = \tilde{p}(t_i) + \epsilon(t_i), \text{ where } \tilde{p}(t) = \kappa p(t),$$

and $\epsilon(t_i), i = 1, \dots, n$ are independent normally distributed random variables with mean zero and unknown variance $\sigma^2(t_i)$. If the observations are mRNA expression levels then instead we have

$$y_m(t_i) = \tilde{m}(t_i) + \epsilon(t_i), \text{ where } \tilde{m}(t) = \kappa m(t).$$

Parameter inference can be performed on data either by sampling from the posterior using MCMC (discussed earlier, also see Jenkins et al. (2013)), or by linear regression. It is not clear how well either method would perform on single-cell data, especially given that the model used is deterministic. Also, the transcriptional time delay is not incorporated into the model, which is necessary to allow us to capture experimentally observed dynamics. Further, the algorithms require some amount of input from the user about the distribution of degradation rates for protein and mRNA, which are not always known and therefore will introduce additional uncertainty which is not accounted for.

Indeed, due to relatively low molecular concentrations for genes of interest in single cells, deterministic models are not able to capture the effect of spontaneous interactions between individual molecules. Stochastic effects are important for many key intracellular processes (McAdams & Arkin, 1999), and not accounting for them can result in large errors in model predictions (Fearnhead et al., 2014; D. J. Wilkinson, 2018).

The natural next step is to extend the above equations to account for noise, for example with the Chemical Master Equation (Gillespie, 2007). This is done by considering each possible individual interaction between molecules, and defines a stochastic differential equation which is, in general, not possible to solve directly. Numerical solutions are available, for example using the stochastic simulation algorithm (Gillespie, 2007), but are typically slow, posing problems for parameter inference methods which are already computationally intensive. Nonetheless, parameter estimation in SDE models requires knowledge of the Markovian transition density (Golightly & Wilkinson, 2005). As a result, a number of methods for performing parameter inference in models which account for stochasticity have been developed. For example, Golightly and Wilkinson (2005) use a diffusion approximation to perform Bayesian inference on a prokaryotic autoregulatory gene network. Reactions are assumed to occur at a constant rate, and the number of reactions that happen in a given time interval is modelled by a Poisson distribution. This gives rise to a simplified stochastic differential equation, from which transition probabilities can be obtained using an Euler discretisation. A drawback of this approach is that the assumption of a Poisson distribution for the number of reactions results in an approximation that does not accurately simulate the true dynamics, due to ignoring the discreteness of the underlying process (Boys et al., 2008). Furthermore, when observational noise is included into the model, the inference performs poorly and does not converge. Boys et al. (2008) instead consider a specific number of reactions in a given time interval $(0, T]$ and, by choosing a conjugate choice of prior for the rate constants derive the posterior distribution

$$p(\theta_k \mid \boldsymbol{y}) \sim \Gamma\left(a_k + r_k, b_k + \int_0^T g_k(y(t))\mathrm{d}t\right),$$

where θ_k are the rate constants, $\boldsymbol{y}$ are the discretely observed gene expression data, a_k and b_k are constants chosen for the prior distributions such that $\theta_k \sim \Gamma(a_k, b_l)$, r_k is the number of type k reactions and g_k defines the reaction propensity. The key problem is that the number and times of each reaction are not known, and have to be somehow inferred from discrete state measurements. This is done using a reversible jump algorithm, combined with a block updating strategy which proposes reaction times using Poisson process approximations to the reaction processes. This method outperforms the diffusion approximation approach, but has much poorer computational efficiency, restricting the number of applicable models.

Other approaches to inference in stochastic reaction networks include sequential Monte Carlo, which relies on multiple simulations from different initial conditions (e.g. particle filtering) (Golightly & Wilkinson, 2011), moment closure based inference (Milner et al., 2013; Zechner et al., 2012), and the linear noise approximation (Fearnhead et al., 2014; Hey et al., 2015). Each of these approaches has their own set of strengths and drawbacks, but typically perform poorly in non-linear systems (Boys et al., 2008). Furthermore, none of them have been applied to models with delayed reactions. Heron et al. (2007) extends the approach of (Golightly & Wilkinson, 2005) to systems with delayed reactions, using bridge building techniques (Elerian et al., 2001).

A more recent method has been developed which incorporates time delays, accounts for stochasticity, and allows us to compute the likelihood, $p(y \mid \theta)$, for single cell protein expression time series data (Calderazzo et al., 2019). The key to this computation is the *Kalman filter*, an algorithm which aims to accurately predict the state of a system (in this case the number of mRNA transcripts and protein molecules) from a series of noisy measurements (Kalman, 1960).

As well as developing methods to infer parameters, it is necessary to understand how much data is needed to accurately infer the true parameters for a given model. Cao and Grima (2019) describe in detail the accuracy of parameter estimation for auto-regulatory transcriptional feedback loops, using time-series data. Their inference methods use summary statistics of the data (mean, variance, and higher-order moments), thus ignoring the effects of dynamic gene expression at the single-cell level. They identified with *in silico* data that for sample sizes larger than a thousand, the number of time points did not affect the accuracy significantly. They also concluded that although their moment-based inferences performed well, the system they used (which did not explicitly describe mRNA) was weakly non-linear, for which these methods perform well. Higher error rates would be expected if they incorporated cooperativity in protein-DNA binding reactions (e.g. in the form of a Hill function). It was also found that accuracy declined rapidly when they varied parameters across cells within a population (Cao & Grima, 2019). This illustrates the necessity of performing inference at either the single-cell or sub-population level, especially for populations with a small number of cells and where gene expression dynamics are important.

2.3.2 Kalman filtering

A Kalman filter is an algorithm which produces accurate estimates of unknown variables of a dynamical system using a series of sparse observations over time which are subject to some degree of error. This is achieved by estimating a joint probability distribution over the variables at each time frame, using an iterative two-step 'predict then update' process. This is the basic concept: At iteration n, we use the dynamical system and our current knowledge to predict the temporal variation of the protein (and mRNA) up to the $n + 1$ observation time point.

Then, the prediction at the $n+1$ observation time point is compared to the actual observation, and the previously predicted (unobserved[2]) states are updated based on this comparison.

Applications of Kalman filters include robotics (Kelly, 1994), neuroscience (Wolpert & Ghahramani, 2000), navigation (Duan et al., 2016; Mirzaei & Roumeliotis, 2008), object tracking (Farahi & Yazdi, 2020; Xiong et al., 2015), and climate forecasting (Hargreaves et al., 2004) among others. Here I will provide a brief overview of the Kalman filtering approach, as well as some additional extensions. For a comprehensive overview, refer to Särkkä (2013). Consider the probabilistic state space model

$$\begin{aligned} \mathbf{X}_k &\sim p(\mathbf{X}_k \mid \mathbf{X}_{k-1}), \\ y_k &\sim p(y_k \mid \mathbf{X}_k), \end{aligned}$$

where $\mathbf{X}_k$ is the state of the system at time k, and y_k is the measurement at time k. The dynamic model is $p(\mathbf{X}_k \mid \mathbf{X}_{k-1})$ and the measurement model is $p(y_k \mid \mathbf{X}_k)$. We are interested in the marginal posterior distribution of $\mathbf{X}_k$ at each k given the history of measurements,

$$p(\mathbf{X}_k \mid y_{1:k}).$$

The Kalman filter (Kalman, 1960) is the closed form solution to the marginal posterior distribution when both the dynamic and measurement models are linear and Gaussian,

$$\begin{aligned} \mathbf{X}_k &= \mathbf{A}_{k-1}\mathbf{X}_{k-1} + \mathbf{q}_{k-1}, \\ y_k &= F\mathbf{X}_k + \varepsilon_k, \end{aligned}$$

where $\mathbf{A}$ is the transition matrix of the dynamic model, F is the measurement model, $\mathbf{q}$ is the process noise, $\varepsilon_k \sim \mathcal{N}(0, \Sigma_\varepsilon)$ and the state space prior is Gaussian, $\mathbf{X}_0 \sim \mathcal{N}(\rho_0, P_0)$.

The filtering equations can be evaluated in closed form as

$$\begin{aligned} p(\mathbf{X}_k \mid y_{1:k-1}) &= \mathcal{N}(\mathbf{X}_k \mid \rho_k, P_k), \\ p(\mathbf{X}_k \mid y_{1:k}) &= \mathcal{N}(\mathbf{X}_k \mid \rho_k^*, P_k^*), \\ p(y_k \mid y_{1:k-1}) &= \mathcal{N}(y_k \mid F\rho_k, FP_kF^T + \Sigma_\varepsilon), \end{aligned}$$

where ρ_k and P_k are the state space mean and covariance at time k given observations $y_{1:k-1}$, and ρ_k^* and P_k^* are the updated mean and covariance given $y_{1:k}$.

Models of gene expression are typically nonlinear, in which case the above defined Kalman filter is not applicable. The extended Kalman filter extends the Kalman filter to nonlinear mod-

[2] The number of unobserved states can be specified. A finer discretization can be chosen, but this reduces the efficiency of the algorithm.

els,

$$\begin{aligned}\mathbf{X}_k &= f\left(\mathbf{X}_{k-1}\right) + \mathbf{q}_{k-1},\\ y_k &= F\mathbf{X}_k + \varepsilon_k,\end{aligned}$$

where f is the dynamic (nonlinear) model function. At each estimation step in the extended Kalman filter, the non-linear system is linearised around the current estimate using Taylor series (Khodarahmi & Maihami, 2023). When the system is highly nonlinear, this linearisation step can result in poor performance (Julier & Uhlmann, 2004). The unscented Kalman filter uses a sampling technique called the unscented transformation which propagates a set of sample points around the mean through the nonlinear prediction and update functions and forms new mean and covariance estimates (Julier & Uhlmann, 2004). Other approaches include the second-order extended Kalman filter (Särkkä, 2013; Singer, 2002), particle filters (Andrieu et al., 2010; Doucet & Johansen, 2011; Golightly & Wilkinson, 2011), the ensemble Kalman filter (Evensen, 2003) and the square root Kalman filter (Van der Merwe & Wan, 2001).

In this thesis I make use of a variant of the Kalman filter, the so called Extended Kalman-Bucy Filter (Singer, 2002).

Formally, suppose $\rho(t) = \mathbb{E}[\mathbf{X}(t) \mid y_{0:t}]$ and $P(t) = \text{Var}[\mathbf{X}(t) \mid y_{0:t}]$, where $y_{0:t} = y_0, y_1, \ldots, y_t$. We wish to find estimates of the mean and variance of $\pi(x_{t+\Delta t} \mid y_{0:t+\Delta t})$ — that is, our belief in the true value at time $t + \Delta t$, $x_{t+\Delta t}$, given all observations up to that time, $y_{0:t+\Delta t}$. The relationship between x_t and y_t is given by $y_t = Fx_t + \epsilon_t$, where $\epsilon_t \sim \mathcal{N}(0, \Sigma_\epsilon)$ and F is a $q \times p$ matrix (where q is the dimension of y_t and p is the dimension of x_t). The prediction step is performed using the Euler-Maruyama approximation. We can derive the unobserved states approximate mean and variance equations over a time-interval δ_t by

$$X_{t+\delta_t} = X_t + g(X_t)\delta_t + a(X_t)\Delta B_t + o(\delta_t),$$

where $\Delta B_t = \sqrt{\delta_t} Z_t$, Z_t is a standard normal and $A(\cdot) = a(\cdot)a(\cdot)^T$.

Dropping the terms of order $o(\delta_t^2)$, it follows that

$$\begin{aligned}\mathbb{E}[X_{t+\delta_t} \mid y_{0:t}] &\approx \mathbb{E}[X_t \mid y_{0:t}] + \mathbb{E}[g(X_t) \mid y_{0:t}]\delta_t,\\ \text{Var}[X_{t+\delta_t} \mid y_{0:t}] &\approx \text{Var}[X_t \mid y_{0:t}] + \text{Cov}[X_t, g(X_t) \mid y_{0:t}]\delta_t\\ &\qquad + \text{Cov}[g(X_t), X_t \mid y_{0:t}]\delta_t + \mathbb{E}[A(X_t) \mid y_{0:t}]\delta_t.\end{aligned}$$

Let $\rho(t) = \rho_t$. The first order Taylor expansion of $g(\cdot)$ and $A(\cdot)$ about ρ_t is

$$g(X_t) \approx g(\rho_t) + J_g(\rho_t)(X_t - \rho_t),$$

$$A(X_t) \approx A(\rho_t) + J_A(\rho_t)(X_t - \rho_t),$$

where J denotes the Jacobian matrix. These expansions are then plugged in to the previous equations, allowing their propagation under linearity. Letting $P(t) = P_t$, rearranging and taking the limit as $\delta_t \to 0$ gives

$$\begin{aligned} d\rho_t &= g(\rho_t)dt, \\ dP_t &= J_g(\rho_t)P_t dt + P_t^T J_g(\rho_t)^T dt + A(\rho_t)dt. \end{aligned}$$

In Chapter 3 we show how this approach can be extended to systems with a fixed delay term. Further details for distributed delay systems can be found in Calderazzo et al.; Mbalawata et al.; Särkkä; Singer (2019, 2013, 2013, 2006). The consequence of using a Kalman filter is that we obtain the likelihood of all the observations given some model parameters, $p(y \mid \theta)$, in a tractable form:

$$\prod_{i=0}^{n-1} \phi(y_{i \cdot l}; F\rho_{i \cdot l}, FP_{i \cdot l}F^T + \Sigma_\epsilon), \tag{2.15}$$

where n is the number of observations, l is the time (in minutes) between observations, and

$$\phi(y; \mu, \Sigma) = \frac{1}{\sqrt{\det(2\pi\Sigma)}} \exp\left(-\frac{1}{2}(y-\mu)^T \Sigma^{-1}(y-\mu)\right).$$

This can then be used in an iterative Markov chain Monte Carlo scheme, such as Metropolis random walk, as was done by Calderazzo et al. (2019).

Algorithm 2 shows how the Kalman filter is combined with a Metropolis random walk approach in order to infer parameters of interest from noisy gene expression data.

In the following section, we iteratively introduce our mathematical model of interest, the auto-negative feedback loop. In Chapter 3 we will implement a Kalman filtering approach for this model, and extend it to make use of gradient-based sampling techniques.

2.3.3 Iterative model building for Hes gene expression

A basic mathematical model for dynamic gene expression can be expressed using ordinary differential equations (ODEs), where the rate of change of concentration of mRNA and protein over time is calculated over a given time span. The simplest model is given by Equations (2.13) and (2.14), with $\tau(t) = \alpha_m \in \mathbb{R}^{\geq 0}$, which asserts that mRNA is transcribed at a constant rate α_m, and degrades at a constant rate μ_m. Similarly, protein degrades at a constant rate μ_p and is

Algorithm 2: Parameter inference via Kalman filtering

```
Input: initial value $\theta_0$, proposal distribution $q$, data $y$, number of iterations $N$
Output: $\theta_1, \theta_2, \ldots, \theta_N$
for $i = 0, 1 \ldots, N$ do
    sample: $\theta' \sim q(\theta' \mid \theta_i)$
    for $j = 0, 1, \ldots, t$ do
        compute: $\rho_j(\theta')$ and $P_j(\theta')$ using the Kalman filter
    end
    compute the likelihood: $\prod_{i=0}^{n-1} \phi(y_{i\cdot l}; F\rho_{i\cdot l}, FP_{i\cdot l}F^T + \Sigma_\epsilon)$
    compute: $\alpha(\theta_i, \theta') \leftarrow \min\left\{\frac{p(\theta'|y)q(\theta'|\theta_i)}{p(y|\theta_i)q(\theta_i|\theta')}, 1\right\} = \min\left\{\frac{p(y|\theta')p(\theta')q(\theta'|\theta_i)}{p(\theta_i|y)p(\theta_i)q(\theta_i|\theta')}, 1\right\}$
    $u \leftarrow$ Uniform$(0, 1)$
    if $u < \alpha(\theta_i, \theta')$ then
        accept the proposal: $\theta_{i+1} \leftarrow \theta'$
    else
        reject the proposal: $\theta_{i+1} \leftarrow \theta_i$
    end
end
```

translated at a rate proportional to the concentration of mRNA, $\alpha_p m$.

In our system of study, we know that the rate of transcription is negatively affected by protein concentration. To account for this, we can modify the mRNA production term to include a function which decreases monotonically from 1 to 0 as protein concentration increases from 0 to ∞.

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \alpha_m f(p) - \mu_m m, \tag{2.16}$$

$$\frac{\mathrm{d}p}{\mathrm{d}t} = \alpha_p m - \mu_p p. \tag{2.17}$$

A popular choice for this function $f(p)$ is the *repressive Hill function* defined by

$$f(p) = \frac{1}{1 + \left(\frac{p}{P_0}\right)^h}, \tag{2.18}$$

where h is the Hill coefficient and P_0 is the repression threshold. Figure 2.10 shows the Hill function defined above for various values of Hill coefficient, h. The number of protein molecules ranges from 0 to 1000, and as the number of protein molecules increases, so does the repression of transcription. As the Hill coefficient increases, so does the steepness of the curve at the intersection point. As $h \to \infty$, the Hill function acts as an on-off switch for transcription. At the repression threshold $P_0 = 500$, transcription is reduced to half its basal rate.

So far our model considers both mRNA and protein expression, as well as repression from protein concentration via the use of a Hill function. Biologically, in order for a protein to re-

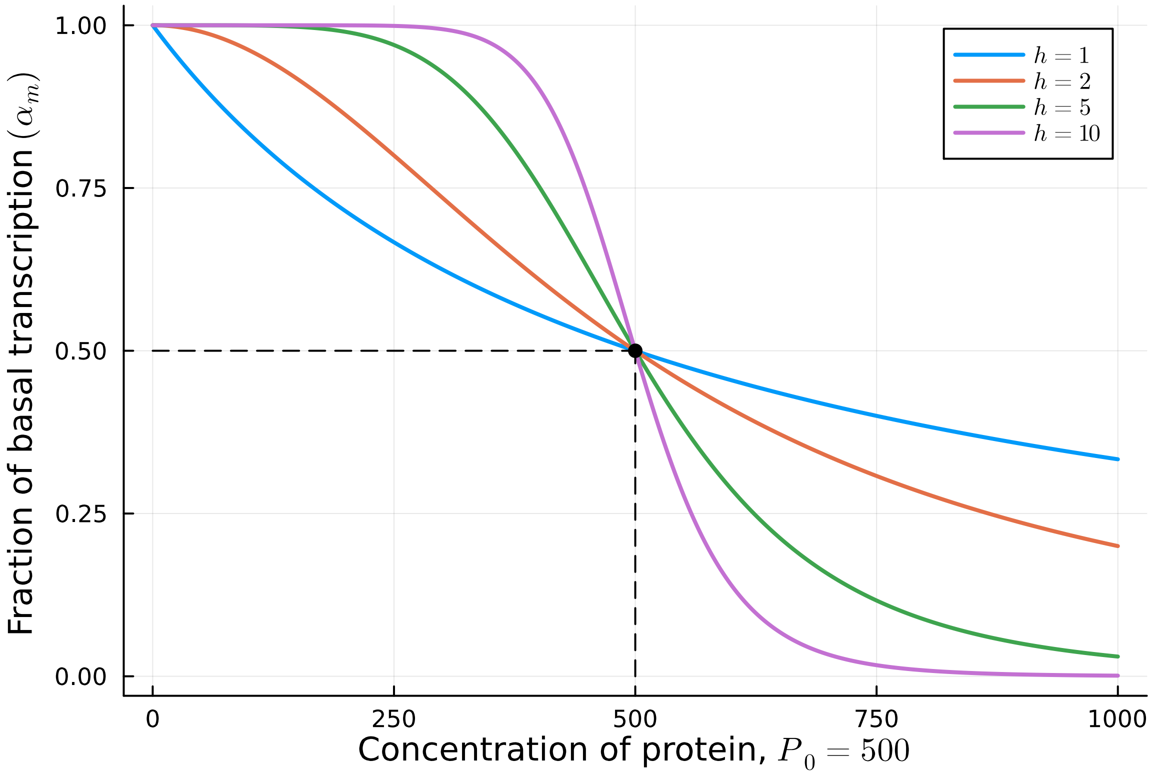


Figure 2.10: A plot of the Hill function for the values $h = 1, 2, 5$, and 10. The repression threshold, P_0 is 500.

press its own transcription, individual molecules must travel to the nucleus and bind to their own receptor on the DNA, and this process is not instantaneous. Further, the process of transcription also takes time. To account for this, we introduce one additional change, which is that the production rate of mRNA at time t is dependent on the amount of protein at some (unknown) time in the past, $t - \tau$.

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \alpha_m f(p(t-\tau)) - \mu_m m, \tag{2.19}$$

$$\frac{\mathrm{d}p}{\mathrm{d}t} = \alpha_p m - \mu_p p. \tag{2.20}$$

In Figure 2.11 we compare the difference between protein dynamics for the systems with and without delay.

We can see in Figure 2.11 that the protein expression in the system with delay is oscillatory but eventually converges to a stable steady state. In experimental data, this trend is typically not seen, and sustained oscillations have been observed. The data is also typically noisier than that produced by a deterministic system. This is due to a finite number of molecules leading

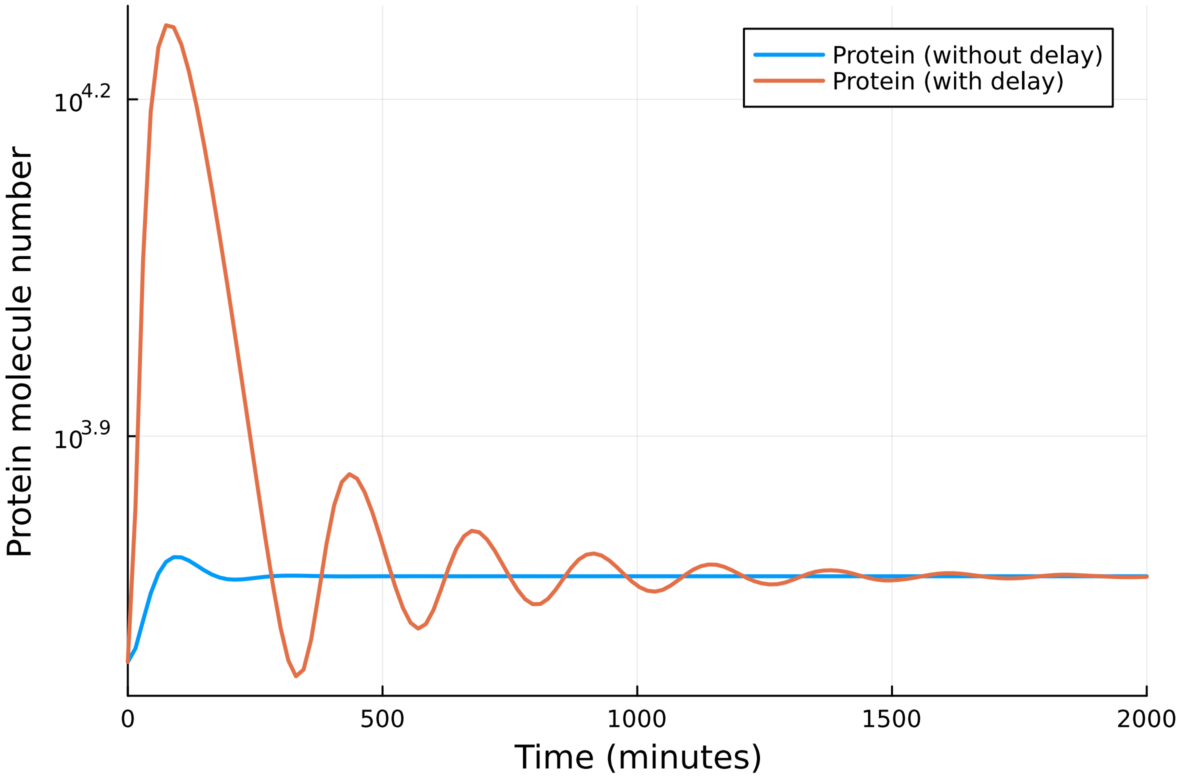


Figure 2.11: Oscillatory protein expression emerges from delayed transcriptional repression.

to transcription and translation happening in a probabilistic way, i.e. only when the relevant molecules interact in the cell. We will use the chemical Langevin equation (CLE) to approximate the stochastic system, which is defined by

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \alpha_m f(p(t-\tau)) - \mu_m m + \sqrt{\alpha_m f(p(t-\tau)) + \mu_m m}\xi_m, \tag{2.21}$$

$$\frac{\mathrm{d}p}{\mathrm{d}t} = \alpha_p m - \mu_p p + \sqrt{\alpha_p m + \mu_p p}\xi_p, \tag{2.22}$$

where ξ_m and ξ_p are temporally uncorrelated, independent Gaussian white noises. A full derivation of the above is given in Chapter 3. Figure 2.12 compares the deterministic delay system with the stochastic system using the same parameters and initial conditions. We can see that oscillations are sustained in the stochastic system and are damped in the deterministic system, for this choice of parameters. This model motivates our inference efforts in Chapters 3 and 4.

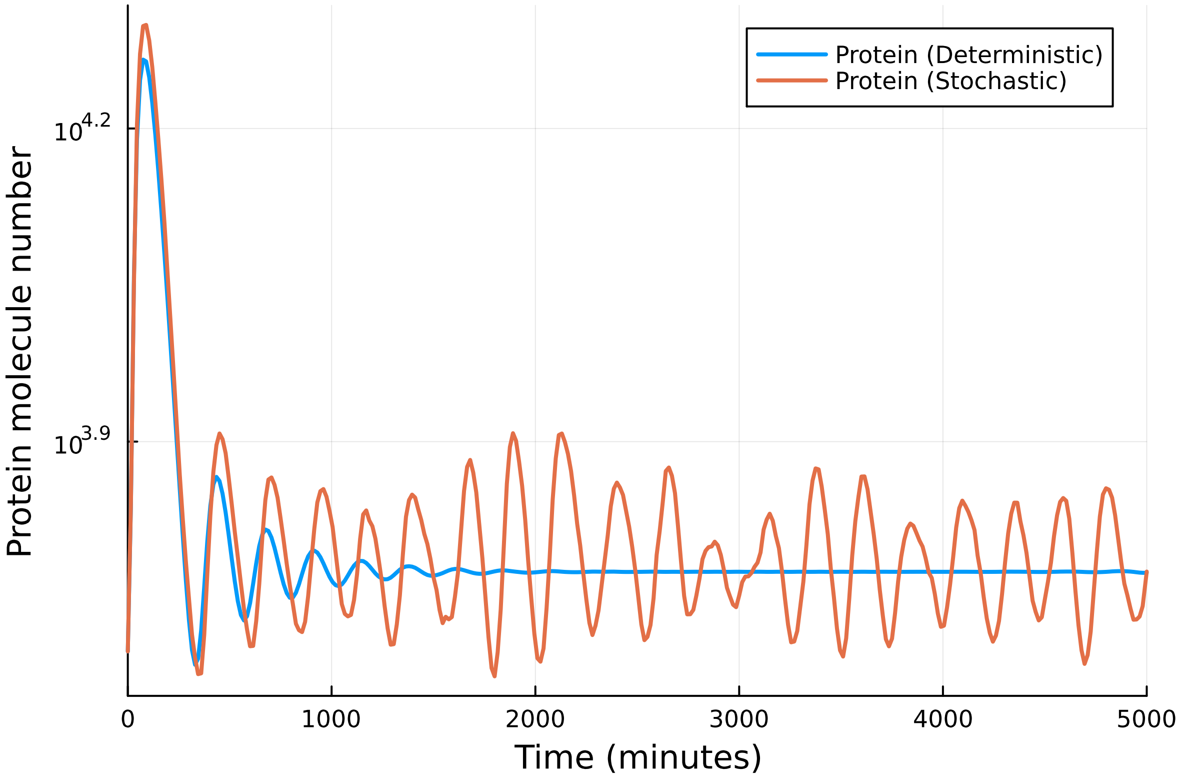


Figure 2.12: Intrinsic noise allows for sustained oscillations which are otherwise damped in the deterministic model.

Chapter 3

Inferring kinetic parameters of oscillatory gene regulation from single cell time-series data

Joshua Burton, Cerys S. Manning, Magnus Rattray, Nancy Papalopulu and Jochen Kursawe

3.1 My contribution

I conceived the study together with J.K., C.M., N.P. and M.R.. I developed all code for the Kalman filtering algorithm, including the calculation of the likelihood gradients, aided by discussion and code review with J.K. who contributed to the algorithm development. I wrote all the code related to MCMC sampling and the adaptive sampling strategies. I also wrote all code for processing and analysing data, simulating data, running *in-silico* experiments, and making figures. I produced all figures, with the exception of panels A–C of Figure 3.1 which were provided by C.M.. I co-wrote the initial draft of the manuscript with J.K., and redrafted it with helpful comments and suggestions from J.K., N.P. and M.R..

3.2 Publication status

This paper was accepted in *J. R. Soc. Interface* on 26th August 2021 and was published on 29th September 2021 (Burton et al., 2021).

Abstract

Gene expression dynamics, such as stochastic oscillations and aperiodic fluctuations, have been associated with cell fate changes in multiple contexts, including development and cancer. Single cell live imaging of protein expression with endogenous reporters is widely used to observe such gene expression dynamics. However, the experimental investigation of regulatory mechanisms underlying the observed dynamics is challenging, since these mechanisms include complex interactions of multiple processes, including transcription, translation, and protein degradation. Here, we present a Bayesian method to infer kinetic parameters of oscillatory gene expression regulation using an auto-negative feedback motif with delay. Specifically, we use a delay-adapted nonlinear Kalman filter within a Metropolis-adjusted Langevin algorithm to identify posterior probability distributions. Our method can be applied to time series data on gene expression from single cells and is able to infer multiple parameters simultaneously. We apply it to published data on murine neural progenitor cells and show that it outperforms alternative methods. We further analyse how parameter uncertainty depends on the duration and time resolution of an imaging experiment, to make experimental design recommendations. This work demonstrates the utility of parameter inference on time course data from single cells and enables new studies on cell fate changes and population heterogeneity.

Keywords: Parameter inference, Bayesian methods, Gene expression oscillations, MCMC, Kalman filters, Stem cell differentiation

Lay summary: The cells in an organism can change their state and behaviour by varying the concentration of key molecules, so called transcription factors. For some transcription factors, extensive amounts of time series data reveal oscillations of their concentrations in time. Here, we introduce a statistical method that can analyse such data to infer the rates at which the transcription factor molecules are produced and degraded, and how strongly oscillating transcription factors self-regulate. Our tool may be used to identify differences between individual cells, and is aimed to facilitate studies on the relationship between transcription factor oscillations and cell behaviour.

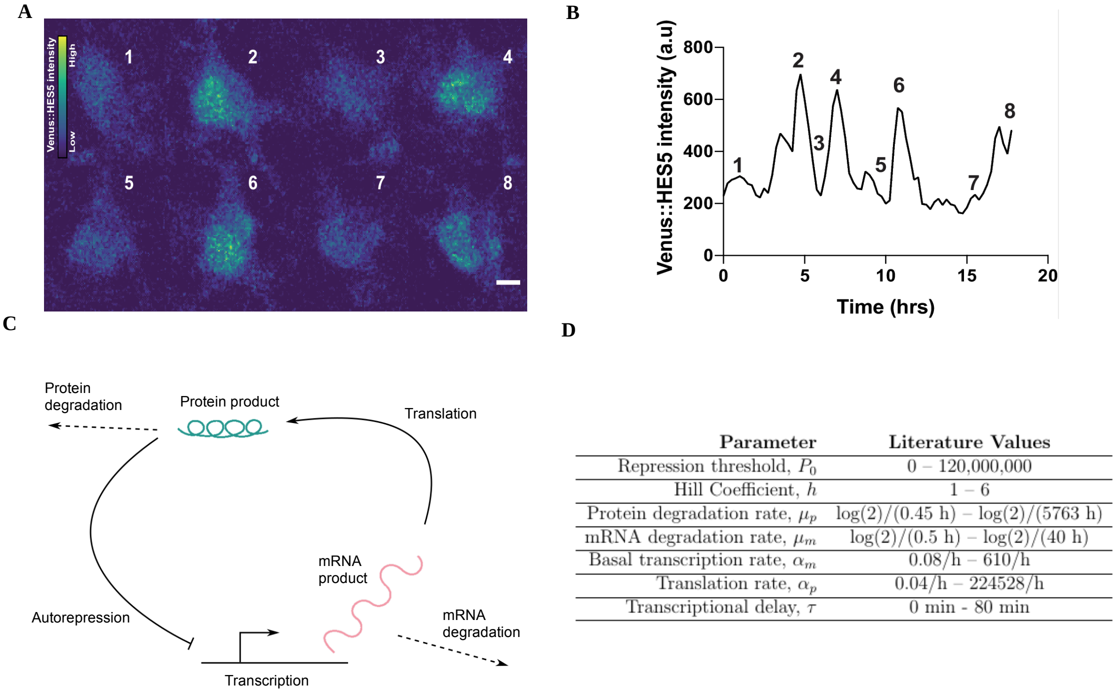
3.3 Introduction

The identification of regulatory mechanisms that control gene expression may have important implications in biological systems. Cell state transitions are a key contributor to many processes in healthy and diseased tissue, and as such they play a major role in development, regeneration, and cancer. There is an increasing amount of literature uncovering the relationship between gene expression dynamics, i.e. dynamic changes in protein copy numbers from a single gene, and cell state transitions (Bansod et al., 2017; Imayoshi et al., 2013; Kobayashi & Kageyama, 2014; Manning et al., 2019; Purvis et al., 2012; Soto et al., 2020; Tan et al., 2012). For example, Imayoshi et al. (2013) used optogenetics to show that oscillatory expression of the transcription factor ASCL1 promotes cell proliferation of mouse neural progenitor cells, whereas sustained expression promotes differentiation. Manning et al. (2019) linked aperiodic HES5 protein expression dynamics to murine neural progenitors, and declining oscillatory dynamics to differentiating neurons. Further evidence by Soto et. al. and Phillips et. al. (Phillips et al., 2016; Soto et al., 2020) demonstrates the contribution of gene expression noise to tuning oscillatory dynamics and influencing dynamically driven cell state transitions.

Experimentally, the dynamics of gene expression can be studied using a variety of approaches. Accurate measurements of protein dynamics are made through live-imaging of transcription factors in single cells, which provides real-time information on gene regulation and identifies cell-to-cell heterogeneity. This can be achieved through fluorescent fusion reporters (Longo & Hasty, 2006), where endogenously expressed proteins are attached to fluorescent reporter molecules. Fluorescence microscopy can then be used to obtain time series data that quantify protein expression levels over time (Figure 3.1A and B, supplementary Section A.10). It may further be possible to translate the fluorescence intensity into exact protein copy numbers (Manning et al., 2019; Soto et al., 2020). Fluorescent protein reporters are widely used to research the role of transcription factor dynamics in cell differentiation events, and have provided dynamic data on gene expression in various contexts, such as neural differentiation, circadian regulation, and cell cycle regulation (Alber et al., 2018; Imayoshi et al., 2013; Manning et al., 2019; Pepe-Mooney et al., 2019; Soroldoni & Oates, 2011; Yang et al., 2020).

Mechanistically, dynamic gene expression is controlled by multiple processes, including transcriptional pulsing (transcription occurring in pulses or bursts), stochastic fluctuations (due to a limited number of molecules), gene regulatory interactions and translational control. In order to understand how these processes interact to modulate gene expression dynamics it is necessary to use mathematical models.

Within systems biology, mathematical models are often represented as a collection of gene regulatory motifs (Alon, 2019; Novák & Tyson, 2008). One very common motif is the delay-mediated, auto-repressive negative feedback loop (Figure 3.1C), which gives rise to oscillations and other dynamic patterns of gene expression that have been observed in somitogenesis, neurogenesis, and in cancer cell lines (Chou et al., 2010; Manning et al., 2019; Purvis et al., 2012; Soroldoni & Oates, 2011; Soto et al., 2020). In this motif, a protein represses the transcription of its own gene. In combination with delays that are intrinsic to biological systems, this admits a range of dynamic behaviours, most notably oscillations at the mRNA and protein level. Regulation of gene expression through the auto-negative feedback motif contributes to cell state changes in multiple systems, including neural differentiation (Chang et al., 2008; Iwasaki et al., 2020; Manning et al., 2019).



Parameter	Literature Values
Repression threshold, P_0	0 – 120,000,000
Hill Coefficient, h	1 – 6
Protein degradation rate, μ_p	log(2)/(0.45 h) – log(2)/(5763 h)
mRNA degradation rate, μ_m	log(2)/(0.5 h) – log(2)/(40 h)
Basal transcription rate, α_m	0.08/h – 610/h
Translation rate, α_p	0.04/h – 224528/h
Transcriptional delay, τ	0 min - 80 min

Figure 3.1: **Time series data of protein expression can be modelled with an auto-negative feedback motif.** **A.** Stills from a movie of a single cortical neural progenitor *in vitro* with Venus::HES5 knock-in reporter. Colour bar shows Venus::HES5 intensity. Stills taken at time points 1.75h (1), 4.5h (2), 6h (3), 7h (4), 9.5h (5), 10.75h, (6), 15.5h (7), 17.25h (8). Scale bar 5μm. For details on data collection see supplementary Section A.10.1, Imaging of Primary NS cells. **B.** Venus::HES5 intensity time series of cell in A. **C.** Graphical representation of the auto-negative feedback motif. **D.** Model parameter values taken from previously published experiments and theoretical considerations. (Goodfellow et al., 2014; Honkela et al., 2015; Lewis, 2003; Manning et al., 2019; Monk, 2003; Phillips et al., 2016; Schwanhüusser et al., 2011).

Despite great advances in the collection of dynamic data on gene expression, and the modelling of these data, challenges remain when calibrating models to data. Even simple mathematical models, such as the auto-negative feedback motif (Figure 3.1C), employ multiple model parameters that correspond to biophysical quantities. For example, the auto-negative feedback motif uses rates of transcription, translation, degradation, and other parameters to predict protein and mRNA expression dynamics. Each of these parameters can take a large range of values (Figure 3.1D). For many application areas, parameter inference, i.e. identifying which parameters correspond to a given experimentally obtained data set, remains an open problem, since it requires the 'inverse' of the model, which typically cannot be computed directly. However, solving this problem bears great potential for the research of gene expression dynamics and its links to cell fate. Identifying which parameter changes correspond to observed differences in protein expression dynamics may illuminate the molecular pathways that contribute to cell fate control, and identify new sources of heterogeneity within a cell population.

The need for parameter estimation in biological systems has motivated extensive research in recent years, with a variety of approaches being developed for different types of data (Babtie & Stumpf, 2017; Browning et al., 2020; Liepe et al., 2014; Warne et al., 2019). Techniques using Bayesian inference have emerged as a preferred approach due to their ability to quantify uncertainty in the face of noisy data, which is a common feature of biological experiments (McAdams & Arkin, 1999), by representing parameters with distributions, rather than point estimates (Cao & Grima, 2019; Finkenstädt et al., 2013; Gómez-Schiavon et al., 2017; Honkela et al., 2015; Jenkins et al., 2013; Munsky et al., 2012; Toni et al., 2009; Vallejos et al., 2016). Placing probability distributions over our parameters, rather than treating them as point estimates, allows us not only to determine the most likely values for each of the parameters, given some data, but also to quantify our uncertainty in them.

To achieve parameter estimation with uncertainty quantification, Bayesian inference aims to identify the posterior distribution of the model under consideration, denoted $\pi(\boldsymbol{\theta} \mid \mathbf{y})$, where $\boldsymbol{\theta}$ and $\mathbf{y}$ are the model parameters and observed data respectively. The posterior distribution describes the probability of the model parameters given observed data, and can be calculated using Bayes' rule

$$\begin{aligned} \pi(\boldsymbol{\theta} \mid \mathbf{y}) &= \frac{\pi(\mathbf{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\pi(\mathbf{y})} \\ &\propto \pi(\boldsymbol{\theta})\pi(\mathbf{y} \mid \boldsymbol{\theta}). \end{aligned} \tag{3.1}$$

Here, $\pi(\mathbf{y} \mid \boldsymbol{\theta})$ is referred to as the likelihood, and is a measure of the fit of a statistical model to the observed data, given specific values of the model parameters. The prior probability, $\pi(\boldsymbol{\theta})$, is a distribution which outlines one's beliefs in the parameters $\boldsymbol{\theta}$ before any new data is taken into account. These prior distributions can be informed using published data (Figure 3.1D), as well as physical constraints (e.g. rate constants must be positive). To visualise the posterior distribution and use it in further analysis it is common to work with computationally generated samples from this distribution. Posterior probabilities may be difficult to compute directly, hindering the efficient generation of these samples (Beaumont et al., 2002; Ramachandra Murthy et al., 2019).

Specifically, it may not be possible to calculate posterior probabilities if the likelihood of the model is not available. In these cases, Approximate Bayesian Computation (ABC) can be used. However, ABC reduces the data to a small number of summary statistics, which inevitably decreases the accuracy of inference (Fearnhead & Prangle, 2012). If an expression for the likelihood is available and can be calculated at given parameter points, the calculation of the marginal likelihood $\pi(\mathbf{y})$ often poses a further challenge in Bayesian inference, since it may require the numerical integration of the likelihood and prior probability. To overcome this challenge, sampling from the exact posterior distribution can be achieved using Markov chain Monte Carlo (MCMC) techniques, such as the Metropolis-Hastings random walk (Hastings, 1970).

MCMC methods can produce samples from the posterior distribution $\pi(\boldsymbol{\theta} \mid \mathbf{y})$ even if the integration factor $\pi(\mathbf{y})$ is unknown. In many scenarios, the reconstruction of a posterior distribution using MCMC sampling can be slow, in particular if the parameter space is high-dimensional, if the calculations of the likelihood are computationally expensive, or if parameters are highly correlated within the posterior distribution (M. D. Hoffman & Gelman, 2014). In these scenarios, more efficient Hamiltonian Monte Carlo (HMC) or Metropolis-adjusted Langevin algorithm (MALA) methods are preferable (Betancourt, 2017; Duane et al., 1987; Roberts & Tweedie, 1996). HMC and MALA algorithms additionally require the gradient of the posterior probability with respect to the model parameters and can result in orders-of-magnitude faster convergence of the sampled distribution to the posterior distribution, especially for high-dimensional distributions or when parameter correlations are present (Betancourt, 2017; Girolami & Calderhead, 2011).

For time series data specifically, a common approach to calculating the likelihood is the Kalman filter. The Kalman filter is an algorithm which calculates the likelihood of the data at each time point, given a mathematical model of stochastic dynamics, and an observation noise model. It can generally be applied to Markov processes, where dynamic changes over time only depend on the current state of the system, and not past states. The Kalman filter is a powerful method to calculate posterior probabilities if delays are not present in the model (Lillacci & Khammash, 2010), and can be extended to estimate the gradient of the likelihood function, making gradient-based sampling of the posterior distribution possible (Mbalawata et al., 2013).

A number of recent methods focus specifically on time series of gene expression (Clerx et al., 2019; Fan et al., 2015; Kojima et al., 2008; Manning et al., 2019; Minas et al., 2017; Moles et al., 2003; Warne et al., 2019). For the study of oscillatory gene expression, a wide array of studies discuss time series data of protein concentrations, such as in Figure 3.1A,B, as well as the description of these data through the auto-negative feedback motif (Figure 3.1C). Despite this, a reliable Bayesian inference method for this popular combination of data and model is still missing. Since the model includes delays, the widely-used Kalman filter approaches are not applicable. Recently, Calderazzo et al. (2019) have addressed this problem by identifying a method to introduce delays into the Kalman filter (Calderazzo et al., 2019), indicating that accurate Bayesian inference for the auto-negative feedback motif on time series data of gene expression may be possible. However, this approach lacks the ability to calculate gradients of the posterior probability distribution, thus preventing the use of efficient gradient-based sampling methods. Furthermore, while Calderazzo et al. (2019) applied their method to a motif containing negative feedback, this method has not yet been applied to the widely used motif in Figure 3.1C, which includes mRNA in addition to protein.

Here, we present a Bayesian inference pipeline that can be used as a non-invasive method to measure kinetic parameters of gene expression emerging from the auto-negative feedback motif using protein expression time course data. We extend the Kalman filtering method presented by Calderazzo et al. (2019) by introducing a recursive implementation to calculate the gradient of the likelihood. This enables us to embed the non-linear delay-adapted Kalman filter into a state-of-the-art MALA sampling algorithm. This extension enhances the robustness of the inference, making it more suitable for use in typical experimental settings.

Our method is able to capture multiple kinetic parameters of gene expression simultaneously using time course data from single cells, and outperforms previous approaches. We demonstrate the accuracy of our method on *in silico* data, provide an example of how the method can be applied to experimental data, and show how the method can be used to obtain experimental design recommendations. This work is paving the way for the use of Bayesian inference methods for the investigation of gene expression dynamics and their links to cell fate.

3.4 Methods

In this section, we give an overview of the key components of our method. First, we introduce the mathematical model for the auto-negative feedback motif. Then, we discuss how we use a delay-adapted non-linear Kalman filter to approximate the likelihood function. Lastly, we provide details on data processing. Descriptions of our method that require longer derivations, as well as further details on data collection, are provided in the Supplementary Information. This includes our implementation of two MCMC methods, Metropolis-Hastings random walk (MH) and MALA, as well as our proposed algorithm to compute the gradient of the likelihood function, which is a major technical advancement in this paper. The availability of this gradient enables the use of a wider range of MCMC samplers, such as MALA, which we use throughout the paper.

3.4.1 The negative feedback chemical reaction network

Here we consider a widely used model of gene expression, that incorporates knowledge of the auto-repressive negative feedback loop (Figure 3.1C). Our model describes both protein and mRNA expression dynamics over time at the level of a single cell, accounting for transcription and translation, as well as degradation. We include a delay in the model, representing the time taken from the initiation of transcription until the production of a transcript and its removal from the nucleus. We further account for the effect of transcriptional auto-repression, where a high abundance of the target protein inhibits transcription of the mRNA (Hatakeyama, 2004; Lewis, 2003; Monk, 2003).

Let $p(t)$ and $m(t)$ define the number of protein and mRNA molecules, respectively, at time t for a gene of interest. Gene expression is often subject to stochastic effects due to finite molecule numbers. To reflect this we model the system with delayed Chemical Langevin Equations (Brett & Galla, 2014; Galla, 2009; Gillespie, 2000),

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \alpha_m f(p(t-\tau)) - \mu_m m + \sqrt{\alpha_m f(p(t-\tau)) + \mu_m m}\xi_m, \tag{3.2}$$

$$\frac{\mathrm{d}p}{\mathrm{d}t} = \alpha_p m - \mu_p p + \sqrt{\alpha_p m + \mu_p p}\xi_p, \tag{3.3}$$

where ξ_m, ξ_p denote Gaussian white noise, i.e.

$$\begin{aligned} \langle \xi_m(t_1)\xi_m(t_2)\rangle &= \delta(t_1 - t_2), \\ \langle \xi_p(t_1)\xi_p(t_2)\rangle &= \delta(t_1 - t_2), \\ \langle \xi_m(t_1)\xi_p(t_2)\rangle &= 0, \end{aligned}$$

where $\delta(\cdot)$ is the Dirac-delta function.

The parameters μ_m, μ_p, α_m, and α_p describe the rate of mRNA degradation, protein degradation, basal transcription rate in the absence of protein, and translation rate respectively. The transcriptional delay is given by τ, and auto-repression is taken into account via the use of a Hill function

$$f(p(t-\tau)) = \frac{1}{1 + [p(t-\tau)/P_0]^h}, \tag{3.4}$$

reducing the rate of transcription for increasing amounts of protein p at time $t - \tau$ (Hirata et al., 2002). Here, τ, the time delay, is the duration of the transcription process. The Hill function (eq. (3.4)) is close to one when the protein at time $t - \tau$ is much less than the repression threshold P_0 and close to zero when the protein at time $t - \tau$ is much more than the repression threshold. The steepness of the transition from one to zero can be regulated by the Hill coefficient h. The Hill coefficient reflects the extent of cooperativity between ligand binding sites for the gene of interest (Weiss, 1997).

From equations (3.2) and (3.3) we can see that the instantaneous rate of transcription is determined by $\alpha_m f(p(t-\tau))$. This allows us to define an approximation for the average rate of transcription as

$$\alpha_T = \alpha_m f(\hat{p}) \tag{3.5}$$

where $\hat{p}$ is the average expression of protein. This average expression of the protein may be obtained from simulated or experimental data.

We simulate the stochastic differential equations (3.2) and (3.3) using the Euler-Maruyama method with a time step $\Delta t = 1$ min, which is chosen sufficiently small to ensure numerical accuracy of the scheme.

A deterministic version of the model in equations (3.2) and (3.3) was first developed by Monk (2003) in order to describe gene expression oscillations of Hes1, p53 and NF-κB, and various versions of the model have since been widely studied (Brett & Galla, 2014; Galla, 2009; Goodfellow et al., 2014; Lewis, 2003; Manning et al., 2019; Monk, 2003; Phillips et al., 2016). In particular, when molecular copy numbers of mRNA and protein are low, we expect the rate processes of transcription, translation, and degradation to stochastically vary with time. This effect is accounted for by the noise terms in the Chemical Langevin Equations (3.2) and (3.3) (Brett & Galla, 2014; Galla, 2009; Gillespie, 2000).

3.4.2 The likelihood function can be evaluated through Kalman filtering

The Kalman filter is an algorithm which calculates the likelihood function for linear stochastic differential equations describing time-series data (Särkkä, 2013). The Kalman filter evaluates the likelihood of each time-point recording consecutively. The full likelihood is then the product of these individual likelihoods, exploiting the Markov property of the underlying stochastic process. The Kalman filter can be extended to non-linear dynamical systems by using piecewise-linear Gaussian approximations (Singer, 2002).

Here, we implement a Kalman filter, extended to account for non-linearity and delay, in order to evaluate the likelihood that our observed data results from the model in eqs. (3.2) and (3.3) at a given parameter combination. This likelihood can then be used to infer model parameters for a given experimentally observed time series recording of gene expression. The resulting posterior distribution may then represent testable predictions on the biophysical characteristics of the gene of interest, such as transcription, translation, and degradation.

Our Kalman filter implementation uses a finer discretisation on the time axis than that given by the observation interval. Specifically, we introduce z hidden states between consecutive observations. Introducing such hidden states is common when applying Kalman filters to non-linear stochastic differential equations. It increases the accuracy of a piece-wise linear Gaussian approximation. In the following, the time variable t will assume integer values numbering all discretisation time points, i.e. $t = 0, 1, ..., nz$, where n is the total number of observations.

It is possible to show that the likelihood of a set of observations given specific model parameters can be expressed as (Calderazzo et al., 2019)

$$\pi(\mathbf{y} \mid \boldsymbol{\theta}) = \prod_{i=0}^{n-1} \phi(y_{i \cdot z}; F\rho_{i \cdot z}(\boldsymbol{\theta}), FP_{i \cdot z}(\boldsymbol{\theta})F^T + \Sigma_\epsilon), \tag{3.6}$$

where the subscript $i \cdot z$ denotes multiplication of i and z and

$$\phi(x; \mu, \Sigma) = \frac{1}{\sqrt{\det(2\pi\Sigma)}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right) \tag{3.7}$$

is the multivariate Normal distribution. Note that here we have explicitly shown the dependence of P and ρ on $\boldsymbol{\theta}$, but will leave it out for readability for the rest of the paper. The true, unobserved state of the system at time t is given by $\boldsymbol{X}(t) = x_t = [m(t), p(t)]^{\mathrm{T}}$, and the relationship between x_t and the observed data y_t is given by $y_t = Fx_t + \epsilon_t$, where $\epsilon_t \sim \mathcal{N}(0, \Sigma_\epsilon)$ and F is a 1×2 matrix. Thus, F and ϵ represent our measurement model. Throughout, we use $F = [0, 1]$, since we aim to apply our method to data on protein expression dynamics, where measurements of mRNA levels are not available. The value Σ_ϵ is called the measurement variance, and describes the observation noise introduced through the experimental measurement process. The variables ρ and P represent the *state space mean* and *state space variance* respectively. We define $y_{0:t} = [y_0, y_z, y_{2z}, \ldots, y_t]^T$, and write $\rho_t = \mathbb{E}[\mathbf{X}(t) \mid y_{0:t-1}]$ and $P_t = \mathrm{Cov}(\mathbf{X}(t), \mathbf{X}(t) \mid y_{0:t-1})$.

The Kalman filter calculates ρ_t, and P_t in eq. (3.6) using an iterative process with two main steps. At iteration k, the first k observations have been used to infer a probability distribution over the true state of the system $\mathbf{X}(t)$ for all discretisation time points up to $t = kz$. This probability distribution is characterised by it's mean $\rho^*_{kz} = \mathbb{E}[\mathbf{X}(t) \mid y_{0:kz}]$ and covariance $P^*_{kz} = \mathrm{Cov}(\mathbf{X}(t), \mathbf{X}(t) \mid y_{0:kz})$.

In the Kalman filter *prediction step* we then use the model to calculate the predicted probability distribution for protein and mRNA copy numbers at the next observation time point, $\mathbf{X}((k+1)z)$. We use this prediction to evaluate the likelihood of the observed data at the $k+1$ observation time point. Before the prediction for the next observation is made, the Kalman filter *update step* is applied, in which the probability distribution of the state space up to observation $k+1$ is updated to take the measurement at $t=(k+1)z$ into account.

For our update step we derive an expression for the mean and variance of the state space distribution $\pi(x_{t-\tau:t} \mid y_{0:t})$, denoted $\rho^*_{t-\tau:t}$ and $P^*_{t-\tau:t}$ respectively. That is, the likelihood of our state space estimates from the past time $t-\tau$ to the current time, t, given all of our current observations. This is necessary in order to accurately predict the state space distribution at the next observation time point, $\pi(x_{t+\Delta t} \mid y_{0:t})$, as past states can affect future states due to the presence of delays. We provide detailed derivations of our Kalman filter prediction and update steps in the Supplementary Information Section A.1.

3.4.3 Implementation of MCMC sampling algorithms

The aim of our inference algorithm is to generate independent samples from the posterior distribution, $\pi(\boldsymbol{\theta} \mid \mathbf{y})$. In this paper, we compare results from two different sampling methods, MH and MALA. The MH algorithm and MALA are two of the most widely used MCMC methods for drawing random samples from a probability distribution. For completeness, we provide their algorithms in the Supplementary Information Sections A.2 and A.3.

Drawing proposals using MALA requires the calculation of the gradient of the log-posterior $U(\boldsymbol{\theta})$, which we outline in Section A.4. This is achieved by iteratively computing the derivatives of state space mean, ρ_t, and state space variance, P_t, with respect to each parameter, as detailed in Section A.5.

3.4.4 Trends in the data are identified by Gaussian processes

Before applying our inference method we detrend protein expression time series using Gaussian process regression, in order to identify and exclude data that show significant long-term trends (Pedregosa et al., 2011; Rasmussen & Williams, 2005) (see Section 3.5.3 for further motivation). Specifically, we make use of a *scaled* squared exponential Gaussian process combined with white noise, whose kernel is given by

$$k(t,t') = \gamma \exp\left(\frac{-\|x(t)-x(t')\|^2}{2l^2}\right) + \eta\delta(t-t'), \tag{3.8}$$

where $\|x(t) - x(t')\|$ is the Euclidean distance between $x(t)$ and $x(t')$, l is the lengthscale, and $\gamma, \eta \in (0, \infty)$. In the Gaussian process regression the hyperparameters γ, l, and η are found using constrained optimisation.

The initial value of the lengthscale is 1000 minutes, and is bounded uniformly in the range (1000 min, 2000 min). The lower bound of this range, 1000 minutes, was chosen to ensure that detrending does not perturb ultradian dynamics in the data. The upper bound, 2000 minutes, was chosen sufficiently large to ensure that detrending is not affected by it. The initial value of the parameter γ is the variance of the data, σ^2_{data}, and is restricted by a uniform prior to $(0.1\sigma^2_{\text{data}}, 2\sigma^2_{\text{data}})$. The parameter η has an initial value of 100, and is restricted by a uniform prior to (10^{-5}, σ^2_{data}). Here, $x(t)$ and $x(t')$ represent our protein expression time course data at time t and t' respectively. We identified data without a significant long-term trend manually by visual inspection (see Section 3.5.3, Figure 3.4) and removed any residual trend before applying our inference method.

3.5 Results

Single cells in a seemingly homogeneous population can change cell fate based on gene expression dynamics. The control of gene expression dynamics can be understood with the help of mathematical models, and by fitting these models to experimentally measured data. Here, we analyse our new method for parameter inference on single-cell time series data of gene expression using the widely used auto-negative feedback motif. We first validate our method by showing the performance of our algorithm (Algorithm 2) on *in silico* data sets. We then demonstrate the utility of our method by applying it to experimentally measured data and, finally, use our method to analyse how parameter uncertainty may depend on certain properties of the data, as well as the experimental design.

3.5.1 Sampled posterior distributions agree with analytical derivations for one-dimensional parameter inference.

We first test our inference method on *in silico* data from the forward model of the auto-negative feedback motif (Figure 3.1C). This is done using Chemical Langevin Equations, as detailed in Section 3.4.1. Specifically, we emulate an in silico imaging experiment by selecting simulated data in sparse intervals of Δt_{obs} mins and mimic measurement noise by adding random perturbations to each observation time point (Figure 3.2A). These perturbations are drawn from a Gaussian distribution with variance Σ_ϵ. Testing the method on *in silico* data first is beneficial, since ground truth parameter values are known *a priori* for the generated *in silico* data sets,

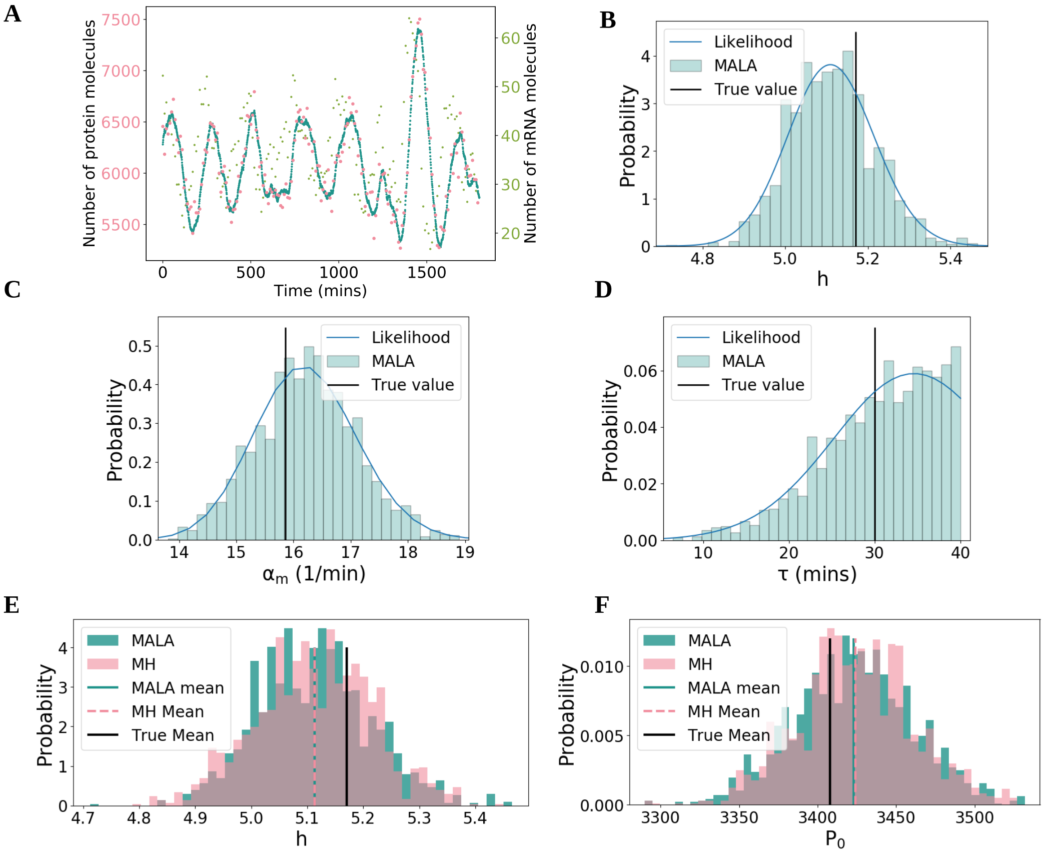


Figure 3.2: **Our algorithm accurately samples posterior distributions. A.** Simulated experimental data. Protein copy numbers are simulated using the Chemical Langevin Equation (see Section 3.4, blue dots). Experimental observations are emulated every five minutes by adding Gaussian noise to the protein copy number (pink). The parameter values used were $P_0 = 3407.99, h = 5.17, \mu_m = \log(2)/30, \mu_p = \log(2)/90, \alpha_m = 15.86, \alpha_p = 1.27, \tau = 30, \Sigma_\epsilon = 10000$, and the simulated mRNA copy numbers are also included (green dots). **B, C, D.** Posterior distributions for one-dimensional inference. For individual model parameters, posterior distributions were inferred while keeping all other parameters fixed, respectively. Shown above are the inferred marginal posteriors for the Hill coefficient (B), transcription rate (C), and transcriptional delay (D) respectively as histograms, using MALA as the underlying sampling algorithm (see Section A.2) for 2500 iterations. The blue lines are the exact likelihood calculations. The sampled and exact distributions coincide. **E, F.** Histograms for both MALA and MH on the 1-dimensional problem for the Hill coefficient (E) and repression threshold (F).

and can be compared to the obtained posterior distributions.

We start by applying our inference method to simple test cases, where the true values of all but one parameter are known, and only the remaining, unknown, parameter value is inferred (Figure 3.2). This allows us to compare our sampled posterior distributions to the exact likelihood, which can be calculated in these one-dimensional examples using equation eq. (3.6). If our inference method is accurate, the sampled posterior distribution should closely match the exact likelihood if the Markov chain has converged (see Section A.7). We find that this is indeed the case, for example, *in silico* data sets (Hill coefficient, transcription rate and transcriptional delay in Figure 3.2B–D, repression threshold and translation rate in Figure A.1). Additionally, ground truth parameter values lie well within the support of the posterior distribution (Figure 3.2B–D, supplementary Figure A.1, vertical black lines).

Our proposed inference method uses the MALA sampler, which relies on calculating likelihood gradients (see Section A.4). The comparison with exact calculations in Figure 3.2B–D and supplementary Figure A.1 validates our implementation of MALA and the associated computations of the likelihood gradient. In order to further test our implementation of MALA, and the associated computations of the likelihood gradients, we compare our results to posterior distributions sampled using the MH algorithm, which does not require gradient calculations. Despite an expected slower convergence of the MH algorithm, this comparison is feasible for one-dimensional posterior distributions, which typically can be well approximated with a few thousand samples. The sampled means have a relative difference below 0.03%, and the standard deviations fall within 4% of each other (Table 3.1, supplementary Table A.3). This comparison reveals that posterior distributions from both samplers agree well with each other (Figure 3.2E–F, Figure A.2), and further validates the implementation of the individual likelihood gradients.

Parameter	**True Value**	$\boldsymbol{\mu}$ **(MALA)**	$\boldsymbol{\mu}$ **(MH)**	$\boldsymbol{\sigma}$ **(MALA)**	$\boldsymbol{\sigma}$ **(MH)**
Repression threshold, P_0	3408	3422	3423	37.51	36.65
Hill Coefficient, h	5.17	5.113	5.112	0.100	0.104

Table 3.1: The true values for the parameters which were used to generate the data in Figure 3.3A, alongside the means, $\boldsymbol{\mu}$, and standard deviations, $\boldsymbol{\sigma}$ of the corresponding one-dimensional posterior distributions, from both the MALA and MH algorithms (Figure 3.2E,F).

3.5.2 Our method allows for simultaneous inference of multiple model parameters

Having validated the method on one-dimensional posterior distributions, we further test the performance of the method by simultaneously inferring multiple model parameters from a single *in silico* data set and comparing the resulting posterior distribution to the ground truth parameter combination (Figure 3.3A,B). Since we cannot measure convergence of the sampled posterior through comparison to the true posterior distribution in the multi-dimensional case, we rely on typical MCMC convergence diagnostics (Section A.7).

We choose a data set that shares characteristics with typically collected time course data from single cells. Specifically, our *in silico* data set is of similar length and observation intervals as previously analysed by Manning et al. (2019). In this paper, the degradation rates of protein and mRNA have been measured, so we assume these measurements as known values, leaving five unknown parameter values to infer. The prior distributions were uniform, defined by the range of values given in Table A.1, and log-uniform for α_m and α_p (see Section A.6 for de-

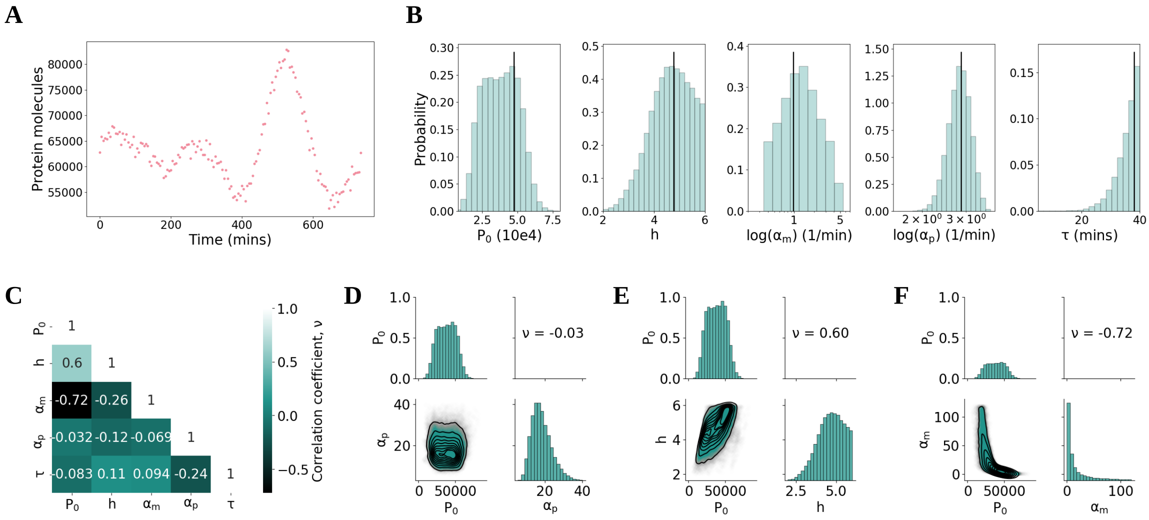


Figure 3.3: **MCMC sampling enables simultaneous inference of multiple parameters. A.** An *in silico* data set was generated using parameter values $P_0 = 47515, h = 4.77, \mu_m = \log(2)/30, \mu_p = \log(2)/90, \alpha_m = 2.65, \alpha_p = 17.61, \tau = 38.0, \Sigma_\epsilon = 10^6$. **B.** Our method is applied to the data set in (A) to sample the joint posterior distribution over five parameters. The marginal posteriors for each parameter are shown. All marginal posterior means are within half a standard deviation of the true value. **C.** A (symmetric) correlation matrix which shows the correlation coefficient, ν, between samples for each pair of parameters. Diagonal entries are, by definition, perfectly correlated ($\nu = 1$), and off-diagonal entries take values in the range $[-1, 1]$. **D,E,F.** Joint posterior distributions show the relationship between different pairs of parameters. D shows the repression threshold, P_0 is not correlated with the protein translation rate, α_p. E shows a correlation of $\nu = 0.6$ between the repression threshold and the Hill coefficient, h. In other words, for this data set, a higher Hill coefficient indicates a higher repression threshold (and vice-versa). Finally, F shows a strong negative correlation between P_0 and α_m.

tails).

We find that the marginal posterior means, i.e. values of largest probability, all lie within maximally half a standard deviation of the ground truth values (Table 3.2). This indicates that a high degree of accuracy in the inference can be achieved with the amount of data typically gathered from a single cell.

Parameter	**True Value**	**Mode**	**SD**
Repression threshold, P_0	47515	44915	12434
Hill Coefficient, h	4.77	4.41	0.80
Log basal transcription rate, $\log(\alpha_m)$	0.975	1.45	1.24
Log translation rate, $\log(\alpha_p)$	2.869	2.808	0.288
Transcriptional delay, τ	38.0	39.87	4.39

Table 3.2: The true values for the parameters which were used to generate the data in Figure 3.2A, alongside the means, $\boldsymbol{\mu}$, and standard deviations, $\boldsymbol{\sigma}$, using MALA.

Simultaneous inference of multiple parameters further allows for the investigation of pairwise parameter correlations, using correlation coefficient ν (Figure 3.3C). Pairwise correlations provide crucial information on how posterior distributions can be constrained further. Specifically, the strong correlation between the repression threshold, P_0, and the logarithm of the basal transcription rate, $\log(\alpha_m)$ (Figure 3.3E), highlights that the data in Figure 3.3A is consistent with either high repression thresholds and low transcription, or vice versa. Such strong pairwise correlations (Figure 3.3E,F) imply that gaining new information on one of the two parameters would constrain the other. This is not the case when parameters are uncorrelated, such as the transcriptional delay and the translation rate (Figure 3.3D), and experimentally measured values on either of these parameters would not inform the other.

3.5.3 Parameter inference on single cell data outperforms previous approaches and may reveal underlying mechanisms for population heterogeneity

Next, we seek to evaluate the performance and utility of our method by applying it to experimentally measured data. Specifically, we investigate data on gene expression oscillations in mouse spinal cord neural progenitor cells (Manning et al., 2019) (see supplementary Section A.10.2, Conversion of Venus::HES5 intensity to molecule number), and compare our method to results on parameter inference from ABC (Figure 3.4A). In this previous approach, the inference was performed using population-level summary statistics of the collected time-course data. This resulted in posterior distributions with high parameter uncertainty. Specifically, the marginal posterior distributions for the Hill coefficient and the transcriptional delay were close to uniform, illustrating that the provided summary statistics did not contain sufficient information to constrain the uniform prior distribution. The remaining parameters had distinct modes. Nonetheless, parameter uncertainty was high since the spread of the posterior distribution was comparable to that of the prior (Manning et al., 2019). Importantly, this previous approach did not allow for the comparison of posterior distributions between single cells.

When applying our method to time series data from fluorescence microscopy experiments, it is necessary to address that our model of the auto-negative feedback motif cannot describe long-term trends in data. Specifically, the model of the auto-negative feedback loop considered here is designed to describe ultradian oscillations that typically have periods shorter than 10 hours (Galla, 2009; Goodfellow et al., 2014; Monk, 2003), and cannot describe variations in protein numbers on longer timescales, such as one would expect from a slow up- or down-regulation of the gene in the tissue. Hence, we only apply our algorithm to protein expression time series that we expect to be accurately modelled by eqs. (3.2) and (3.3) by excluding data that show significant long-term trends. In order to identify such time series, we first remove trends from the time series that vary on lengthscales longer than 10 hours by using Gaussian process regression (see Section 3.4.4). Then, we manually identify all the time series for which the detrended and raw time series visually agree (Figure 3.4B) and select these

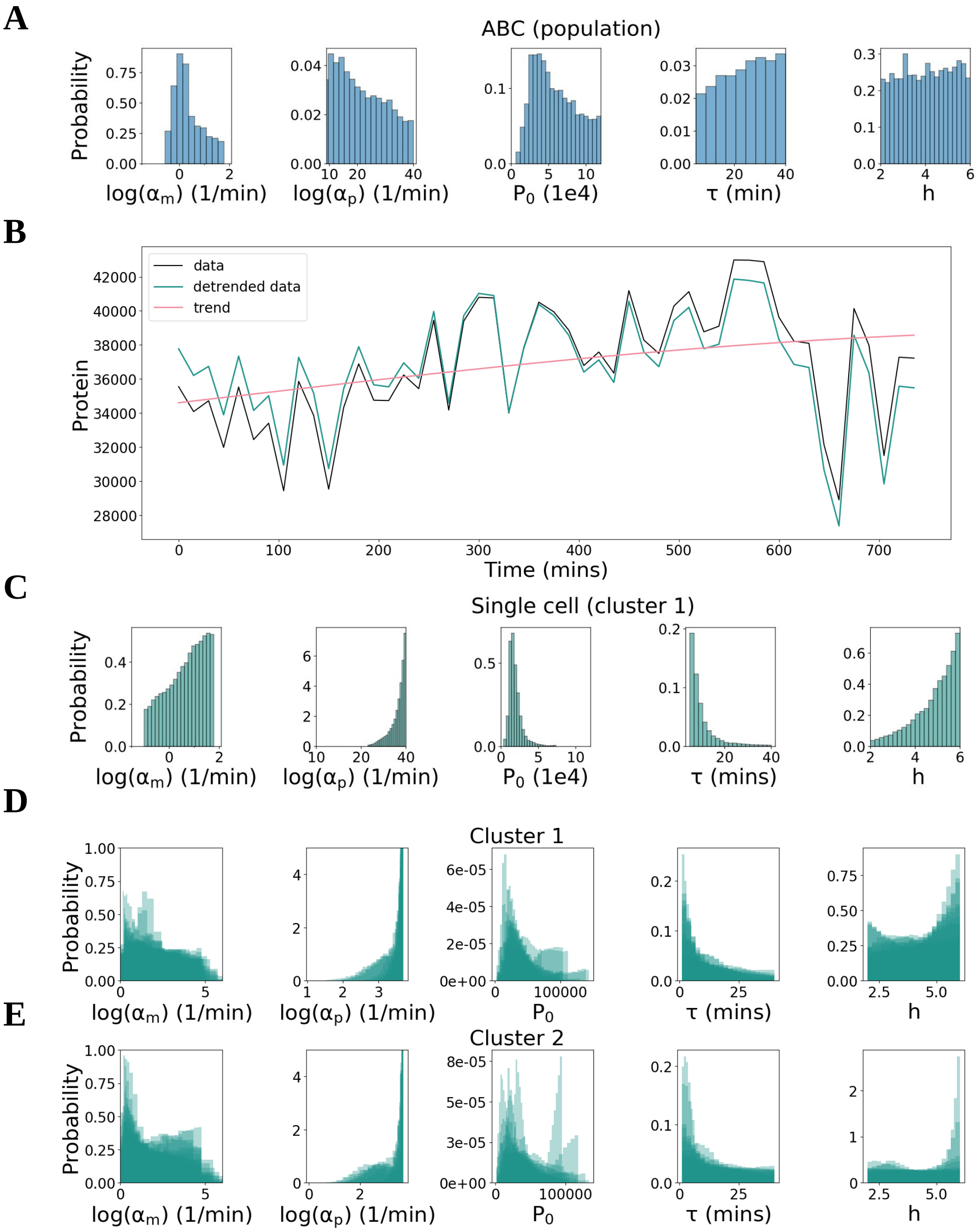


Figure 3.4: **Parameter inference on single cell data outperforms previous approaches.**

for inference.

In order to identify a suitable value for the measurement variance Σ_ϵ we rely on previous estimates (Manning et al., 2019). Manning et al. (2019) decomposed the measured time series into two contributions, one from a time-varying signal with finite auto-correlation time, and

Figure 3.4: **Parameter inference on single cell data outperforms previous approaches. A.** Marginal posterior distributions for each parameter are obtained from the ABC algorithm using summary statistics in place of likelihood evaluations. **B.** Detrending of *in vivo* single cell protein expression data (black line). The sampling interval is 15 minutes, and the length of the time course is 12 hours (720 mins). The mean is subtracted from the data and a squared exponential Gaussian process determines the long term trend (pink line). This trend is removed from the data and the mean is added back in (green line). **C.** Marginal posteriors from MALA for the detrended single cell data shown in panel B. **D,E.** Marginal posterior distributions for every cell in clusters 1 and 2 respectively.

one from a time-varying signal for which consecutive observations are uncorrelated (Manning et al., 2019). This second contribution follows an identical distribution as the measurement error in our model, and was estimated to contribute 10% of the total variance across all detrended time series. Hence, we set

$$\Sigma_\epsilon = \frac{0.1}{N_c}\sum_{i=1}^{N_c}\sigma_i^2, \tag{3.9}$$

where N_c is the number of considered traces, and σ_i^2 is the variance for the ith detrended data set.

We find that our method can identify more accurate posterior distributions than the previous ABC-based approach by using single cell time series of gene expression only (Figure 3.4C vs. Figure 3.4A.). For the single-cell gene expression time course in Figure 3.4B, we find that there is still comparatively high uncertainty on the basal transcription rate (α_m in Figure 3.4C), as the support of the marginal posterior distribution reflects that of the uniform prior distribution. However, for all other model parameters that are inferred from this time course, the marginal posterior distributions are narrower than the prior, and than previously identified marginal posterior distributions from ABC (Figure 3.4C).

Having investigated marginal posterior distributions from a single cell we proceed to analyse to what extent these posterior distributions can vary across multiple cells in the population. Among the experimental data considered here, hierarchical clustering has previously identified two sub-populations (denoted as clusters 1 and 2) which have distinct gene expression dynamics and which also do not have strong long-term trends (Manning et al., 2019), such as down-regulation of gene expression. For clusters 1 and 2 there are 19 and 22 cells respectively which we have selected for negligible trends (see Section 3.4.4).

We find that the posterior distributions inferred from multiple cells share features that are conserved across all cells and both populations (Figure 3.4D,E). Specifically, the marginal posterior distributions of the translation rate α_p are all larger than exp(2)/min, and biased to larger values. Similarly, the marginal posterior distributions for the delay τ cover the entire range of considered values, and are biased towards smaller values, with most likely values below 10 minutes. These observations appear to hold true for both clusters considered here, and they highlight that parameter inferences obtained from our method are biologically reproducible, which is a necessary feature to enable the use of the method in practical applications.

In contrast, for the basal transcription rates α_m and the Hill coefficient h, marginal posterior distributions vary between individual cells, suggesting that there is an underlying heterogeneity of these parameters across the cell population. However, the remaining parameter uncertainty is too high to reliably identify differences between cells and clusters of cells, raising the question of how imaging protocols may need to be changed in order to achieve lower uncertainty on typical parameter estimates.

3.5.4 Longer time course data improves accuracy of inference more effectively than more frequent sampling

Typically, longer imaging time series can only be collected at the cost of a lower imaging frequency. When designing experiments, it may be desirable to choose an imaging protocol that optimises the parameter inference towards high accuracy and low uncertainty. However, parameter uncertainty may not only be influenced by the imaging protocol, but also by the bifurcation structure of the underlying dynamical system (Roesch & Stumpf, 2019). Hence, we next analyse how posterior distributions depend on the frequency of sampling, on the length of the imaging interval, and on the position in parameter space. To evaluate the performance of our inference, we investigate the uncertainty using *relative uncertainty*, RU$_{\boldsymbol{\theta}}$ (Section A.7, eq. (A.26)), which quantifies the spread of the posterior distribution. We use this metric to quantify the performance of our inference method on a number of synthetic data sets with different lengths and sampling frequencies, and for different locations in parameter space.

We choose two locations in parameter space that correspond to two different values of oscillation coherence, thus producing qualitatively different expression dynamics (Figure 3.5A, Table A.2). The oscillation coherence is a measure of the quality of observed oscillations (supplementary Section A.8). Choosing parameter combinations with different coherence thus ensures that these correspond to different positions within the bifurcation structure of the auto-negative feedback loop (Galla, 2009; Momiji & Monk, 2008; Zhuang & Zhu, 2010).

We first analyse to what extent collecting data for a longer sampling duration may reduce parameter uncertainty (Figure 3.5B,D). We find that a longer sampling duration can strongly decrease parameter uncertainty. Doubling the length of the time series reduces the uncertainty by 19% on average for the high coherence parameter combination, and 7.1% on average for the low coherence parameter combination. A tripling of the available data leads to reductions in uncertainty by 29.8% and 18.3% and for high and low coherence respectively.

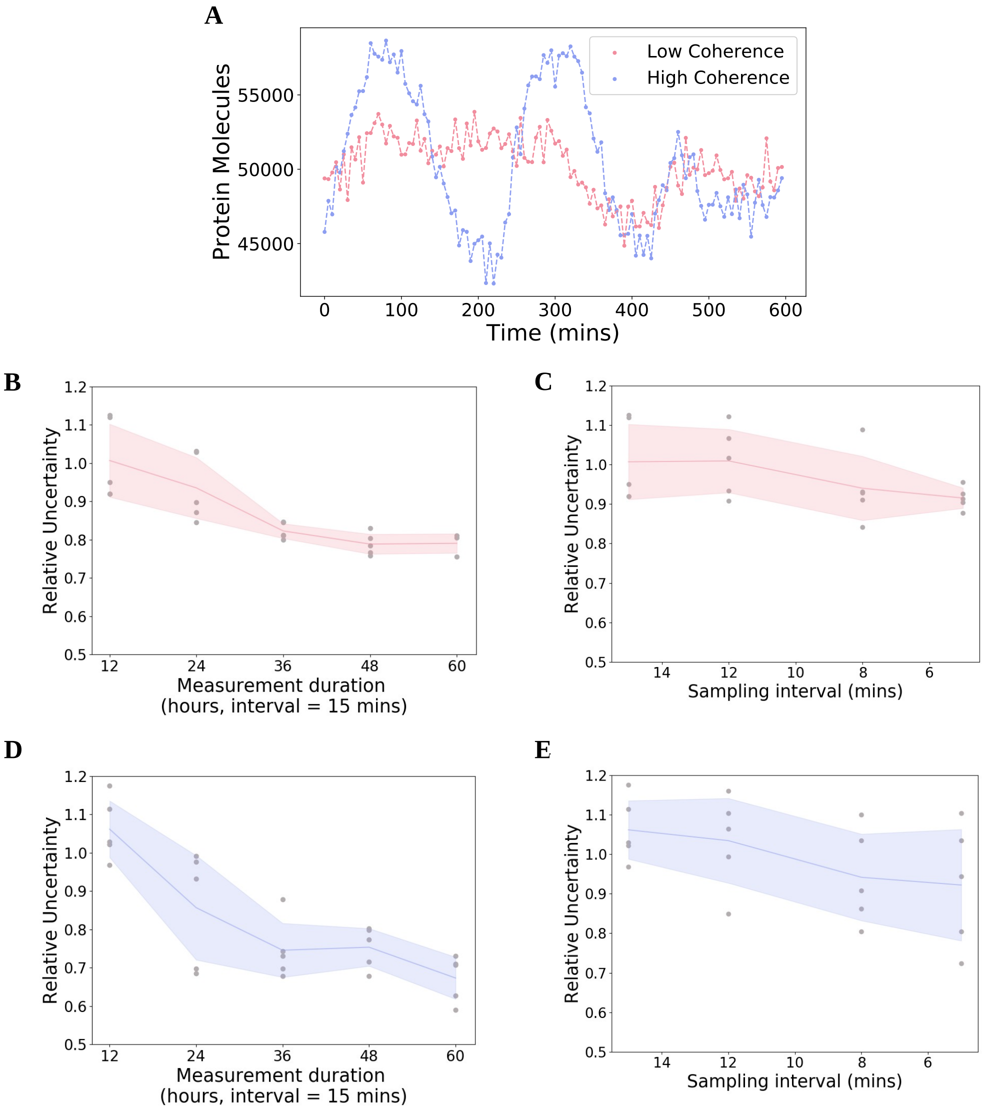


Figure 3.5: **Increasing the length of time course data improves inference more than increased sampling frequency. A.** Two examples of *in silico* protein observations, one which has low coherence (pink) and another with high coherence (purple). Exact parameter combinations can be found in Table A.2. **B.** RU$_{\boldsymbol{\theta}}$ for low coherence data sets sampled with different lengths, from 12 hours to 60 hours. As length increases from 12 hours to 36 hours, both the mean and standard deviation decrease. **C.** RU$_{\boldsymbol{\theta}}$ for low coherence data sets sampled with different frequencies. As frequency increases from 15 minutes to 5 minutes, the mean decreases. **D.** Same as B with the high coherence data sets. **E.** Same as C with the high coherence data sets.

In contrast, an increase in sampling frequency leads to a smaller decrease in parameter uncertainty on average (Figure 3.5C,E). Specifically, doubling the amount of data only leads to a decrease by 11.3% in the case of the high coherence parameter combination, and 6.7% in the case of low coherence. A tripling of the available data leads to reductions in uncertainty of 13.2% and 9.1% for low and high coherence respectively.

We find that analogous conclusions hold true if inference accuracy is analysed (ME$_{\boldsymbol{\theta}}$, supplementary Section A.7, eq. (A.27), instead of uncertainty (Figure A.3). Accuracy increases with longer sampling durations and shorter imaging intervals, and longer sampling durations have a stronger effect than shorter imaging intervals.

3.5.5 Additional measurements of mRNA copy numbers improve estimates of the average transcription rate

In the previous section, we have analysed the impact of changes in the imaging protocol on parameter uncertainty overall. Alternatively, it may be desirable to identify interventions that reduce uncertainty for particular parameters of interest. For example, an important quantity of interest may be the average rate of transcription of the investigated gene, introduced as α_T in equation (3.5). In many *in silico* examples of our parameter inference, this average rate of transcription α_T is poorly inferred, with the mean of the posterior distribution being up to 5 times larger than the ground truth value. This is for example the case in Figure 3.6A. In this and other examples, the ground truth value lies outside the 85% highest density interval (HDI) of the posterior distribution (Figure 3.6A–C). Intuitively, one may assume that estimates for the rate of transcription are improved if measurements of mRNA copy numbers, in addition to protein expression dynamics, are considered in the inference.

Hence we next assume that, in addition to data on the dynamics of protein expression, measurements of mRNA copy numbers have been conducted on the observed cells. Specifically, we generate *in silico* data mimicking a single-molecule *in situ* hybridisation (smFISH) experiment. Such smFISH experiments generate distributions of mRNA copy numbers, thus providing a snapshot of mRNA levels across a population at a fixed time point (Femino et al., 1998; Raj et al., 2008). To account for this additional data, we incorporate the observed distribution of mRNA copy numbers into our likelihood function, such that it effectively penalises parameters for which inferred copy numbers of mRNA are outside the experimentally observed range (see supplementary Section A.9).

We find that this inclusion of mRNA information collected from a cell population leads to more accurate inference of the average transcription rate for single cells, using our algorithm (Algorithm 2). Observing example data sets from Figure 3.5, the posterior distributions cover multiple orders of magnitude if only protein expression data is considered in Figure 3.6D,E, with the mean of the distribution being 5.4 times larger than the true value in Figure 3.6D, and 2.4 times larger in Figure 3.6E, respectively. Upon inclusion of mRNA information, these posterior distributions are instead concentrated around the true value, with a relative error below 15.3%. In both examples the ground truth is contained within the 65% HDI. In Figure 3.6F a posterior distribution that is already close to the true value gets further constrained

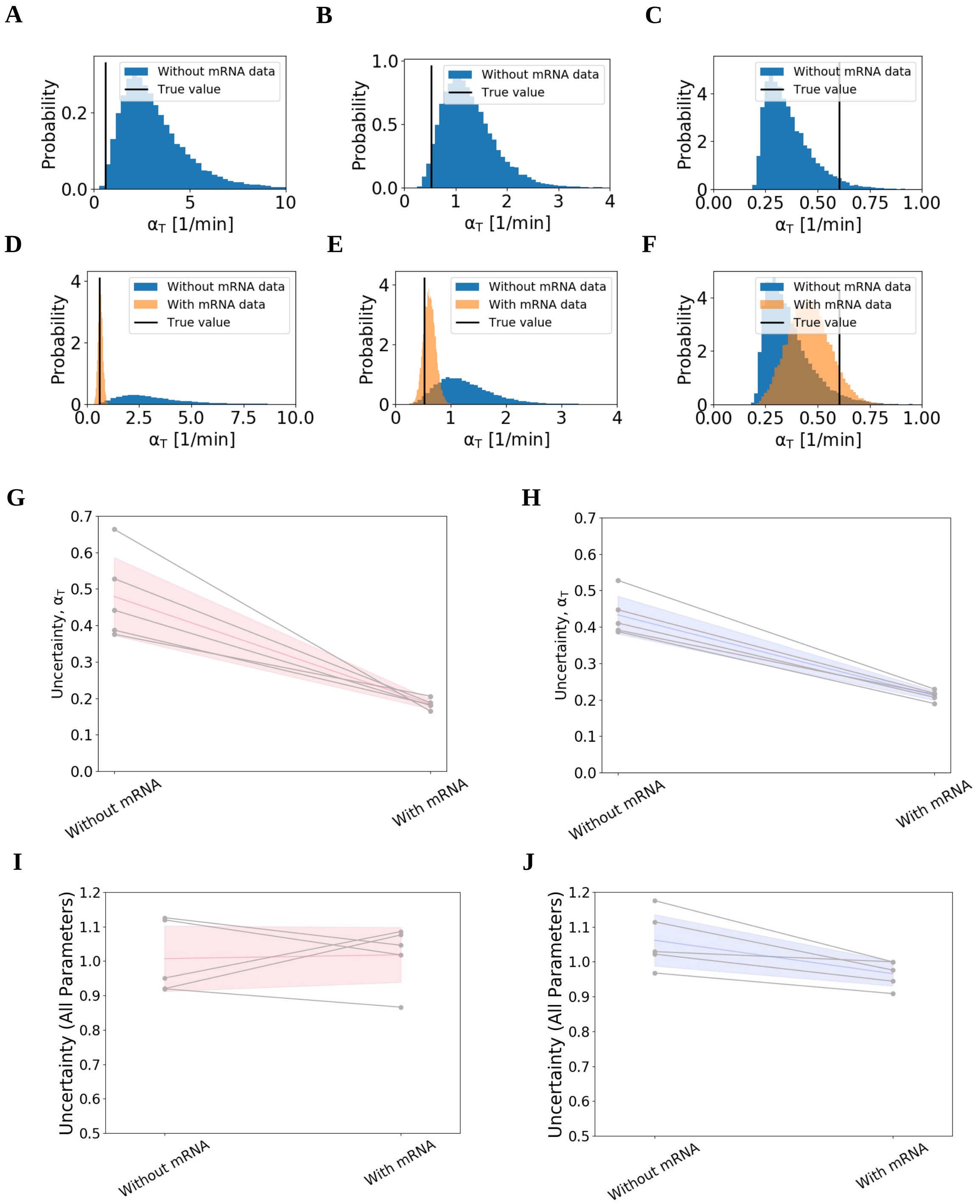


Figure 3.6: **Additional measurements of mRNA copy numbers improve estimates of the average transcription rate.**

by the additional mRNA data. In these examples, the observed reduction of uncertainty on the inferred transcription rate is accompanied by a reduction of uncertainty on estimated mRNA copy numbers for individual cells, as inferred by the Kalman filter (see e.g. supplementary Section A.10.2A vs. Section A.10.2B). Investigating the uncertainty on the average inferred

Figure 3.6: **Additional measurements of mRNA copy numbers improve estimates of the average transcription rate. A,B,C.** Posterior distributions of the average rate of transcription, α_T, calculated using the posterior samples of three example data sets from Figure 3.5. The ground truth value (vertical black line) is poorly estimated by these posteriors. **D,E,F.** The same three posterior distributions for α_T as in A–C, this time comparing posterior samples drawn without mRNA information (blue) and with mRNA information (orange). Here the ground truth value (vertical black line) is much more closely inferred when mRNA information is included. **G.** Uncertainty of α_T for the low coherence data sets from Figure 3.5. Uncertainty in α_T is reduced by more than 60% when mRNA information is included. Uncertainty on α_T is calculated using the coefficient of variation, defined by the posterior standard deviation over the posterior mean, $\hat{\sigma}_{\alpha_T}/\hat{\mu}_{\alpha_T}$. **H.** Same as in G but for high coherence data sets. Here mRNA information reduces uncertainty in α_T by more than 50%. **I.** Values of relative uncertainty RU$_{\boldsymbol{\theta}}$ (eq. (A.26) in supplementary Section A.7) for low coherence data sets (cf. Figure 3.5) with and without additional data on mRNA copy numbers. **J.** RU$_{\boldsymbol{\theta}}$ for high coherence data sets from Figure 3.5 with and without additional data on mRNA copy numbers. In G–J the coloured lines and shaded areas represent the mean and standard deviation across observed values, respectively. Data sets in A–C are all sampled for a duration of 12 hours, and with sampling intervals of 8 minutes, 8 minutes, and 5 minutes respectively. All data sets considered in G–J are sampled every 15 minutes for a duration of 12 hours and the parameters correspond to the low and high coherence parameter sets respectively (see Table A.2).

transcription rate across data sets introduced in Figure 3.5, we observe a reduction in uncertainty of 61.8% for low coherence parameter combinations (Figure 3.6G), and a reduction in uncertainty of 51.2% for high coherence parameter combinations (Figure 3.6H).

How does this improved estimate of transcription rate affect overall uncertainty across parameter space, as analysed in Figure 3.5? Counter-intuitively, we find that this inclusion of mRNA data into our parameter inference does not reduce overall parameter uncertainty (Figure 3.6I,J). For data sets from the low coherence parameter combination, the relative uncertainty increases by 1.1% on average when mRNA information is included (Figure 3.6I). For data sets from the high coherence parameter combination, uncertainty decreases slightly (9.1% on average (Figure 3.6J)). Importantly, this reduction of uncertainty is considerably smaller than the reduction of uncertainty observed when longer measurement durations are considered (cf. Figure 3.5D). We make analogous observations as inference accuracy is analysed (Figure A.4), instead of uncertainty. Inference accuracy is not reduced for high coherence data sets when data on mRNA copy numbers are included, and it is only slightly reduced for some of the low coherence data sets, with the effect being much smaller than the effect of considering longer time course data (cf. Figure A.3).

The effect that overall uncertainty is not decreased as new data on mRNA copy numbers are included contradicts the intuition that more accurate inference of the average rate of transcription α_T will also reduce uncertainty on model parameters regulating α_T, such as the basal transcription rate, α_m and the repression threshold, P_0. This effect may be attributed to correlations between these parameters, which we typically observe in our posterior distributions (see Figure 3.3F). For the data set in Figure 3.6A, inference of α_T is improved upon the inclusion of mRNA information. This leads to a tighter coupling between the parameters α_m and P_0 (Figure A.6). However, this constraining of the posterior distribution is not reflected in either of the marginal posterior distributions. Thus, although the inclusion of *in silico* smFISH data reduces the spread of the posterior distribution overall, uncertainty within the marginal posterior distributions is not reduced, and individual parameter estimates are not improved. An additional factor is that data from smFISH experiments may be considered to reflect the time-averaged mRNA copy number distribution of single cells. Hence, these data might not reduce uncertainty on parameters that are expected to predominantly alter the dynamics rather than the level of expression, such as the transcriptional delay τ and Hill coefficient h. Hence, to better infer these parameters, other strategies, e.g. those discussed in Figure 3.5, may be required.

We conclude that distributions of mRNA copy numbers from population-level measurements can be used to infer average transcription rates for individual cells, using our inference method, which may facilitate the study of transcriptional dynamics in the context of gene expression oscillations. Together with results from Figure 3.5, this illustrates how our method may be used to evaluate the benefit of different experiments *in silico*, and highlights that our method can be naturally extended to utilise additional data of different types.

3.6 Discussion

The aim of this work was to develop a statistical tool that can be used to infer kinetic parameters of the auto-negative feedback motif, based on typically collected protein expression time series data from single cells. Importantly, the stochastic nature of the involved processes demanded a method that enables uncertainty quantification. We have achieved our aim by embedding a non-linear delay-adapted Kalman filter into the MALA sampling algorithm. Our method can generate accurate posterior distributions for the simultaneous inference of multiple parameters of the auto-negative feedback motif. The produced posterior distributions are more informative than those from previous approaches. Since our method can be applied to data from single cells, it enables the investigation of cell-to-cell heterogeneity within cell populations. It can further be used to make experimental design recommendations, which we demonstrated by investigating how parameter uncertainty may depend on the position in parameter space, the sampling frequency, and the length of the collected time series data. Additionally, our method may be extended to account for the presence of different types of data, for example to improve estimates of the transcription rate for individual cells.

Often, new inference algorithms are presented on a single data set, and due to necessary tuning requirements of the involved sampling methods, further data sets are not considered. However, it is important to understand the behaviour of a method for a range of data sets if we wish to make experimental design recommendations. It is an achievement of this paper that we provide a method that demonstratively can reliably infer parameters, even when the size and structure of the data are changed significantly.

The mathematical model underlying our method aims to describe the dynamic expression of a protein which is controlled by auto-negative feedback. The success of our inference relies upon how well this model approximates reality. Mathematical models for the oscillatory expression of transcription factors are informed by experimental research (Bonev et al., 2012; Hirata et al., 2002) and have been developed over time (Lewis, 2003; Momiji & Monk, 2008; Monk, 2003; Phillips et al., 2016; Soto et al., 2020). Existing model extensions include interactions with other genes or microRNAs (Goodfellow et al., 2014) and future models could include effects of transcriptional bursting (Tunnacliffe & Chubb, 2020). The simple model used here provides a starting point for developing inference algorithms for further models including non-linear, stochastic interactions as well as delays, and future validation of experimental predictions can be used to guide data-driven model improvements. To this end, our algorithm may enable model selection to identify gene regulatory network properties, such as interactions between multiple transcription factors.

Chemical Langevin Equations such as eqs. (3.2) and (3.3) approximate the full stochastic dynamics of the system by assuming Gaussian increments. Further, our Kalman filter assumes that measurement errors follow a Gaussian distribution, and are not correlated between consecutive time points. The likelihood calculations within the Kalman filter assume that distributions of protein copy numbers, which are predicted by eqs. (3.2) and (3.3), can be approximated by Gaussian distributions.

The Gaussian approximation within the Chemical Langevin Equation can break down when molecule concentrations are very low, resulting in an inaccurate simulation of the dynamics. We do not expect this to be a problem for data analysed in this paper, since protein copy numbers throughout our analysis are around 50,000 protein molecules per nucleus. In other applications, the validity of the Chemical Langevin Equation may be explicitly tested on samples from the posterior distribution by directly comparing simulated expression time series with those obtained from an exact sampling algorithm, such as the Gillespie algorithm (Gillespie, 2007). Similarly, simulations of the Chemical Langevin Equation can be used to test assumptions on the Gaussianity of the state space made within the Kalman filter. In cases where these assumptions do not hold, alternative inference algorithms, such as particle filter methods, may need to be developed.

For Bayesian inference problems it is common to use MCMC samplers, such as MH or MALA. We have found that combining a delay-adapted non-linear Kalman filter and MALA can allow us to infer parameters of the auto-negative feedback motif. This builds on previous approaches which applied a Kalman filter in the context of a different transcriptional feedback motif with delay (Calderazzo et al., 2019). MCMC algorithms typically require tuning which may be data specific. We have taken steps to reduce additional input from the user by using MALA, which proposes new samples based on the gradient of the target posterior, hence accounting for geometric properties of parameter space, which can result in faster, more robust performance on some distributions (Girolami & Calderhead, 2011). MALA also has fewer tuning parameters than other algorithms, such as HMC. This allows us to more easily incorporate adaptation into our algorithm (Andrieu & Thoms, 2008). Surprisingly, the MALA sampler did not result in faster convergence than MH on example posteriors from our model (see Supplementary Figure A.7). Hence, the added computational cost of calculating likelihood gradients will not be beneficial in all applications, especially since, in our model, gradient calculations increase the computational cost of individual parameter samples by a factor of 12. We expect the availability of likelihood gradients to achieve a speed-up in high-dimensional problems, where convergence speeds of MALA scale with $d^{1/3}$, rather than d^1 for MH (Roberts & Rosenthal, 2001), for model dimension d. Note, that more efficient MCMC algorithms can eliminate the problem of tuning entirely (Girolami & Calderhead, 2011). These methods rely on the computation of the Hessian, i.e. the second derivative of the likelihood function. Deriving expressions for the Hessian and investigating the efficiency of the resulting algorithm is thus a potential avenue for future work.

In our applications of the algorithm to experimentally measured data, we detrended the data before applying our inference (Figure 3.4B). Such detrending is commonly used when analysing time series of oscillatory signals (Manning et al., 2019; Mönke et al., 2020; Phillips et al., 2017). The detrending removes signal fluctuations from the recorded time series that vary on a much longer time scale than the ultradian oscillations that are being considered. This is necessary, since our model cannot describe such long-term fluctuations. Specifically, independently of the model parameter chosen, simulated traces from the Chemical Langevin Equation (eqs. (3.2) and (3.3)) do not include long-term trends. Hence, detrending prevents any bias that the presence of a long-term trend in the data may introduce to the parameter inference. When the algorithm will be applied to data from other transcription factors, we recommend excluding data that contains trends with timescales that are longer than the fluctuations and oscillations that are expected to emerge from the auto-negative feedback, in line with previous detrending recommendations (Mönke et al., 2020; Phillips et al., 2017). Presumably, variations in the long-term trend of the data stem from a time dependence of one or multiple of the model parameters due to regulatory processes that our model does not account for. Hence, future improvements to our algorithm may be developed where the temporal variation of model parameters is inferred, instead of one static value.

When applying our inference method to experimental data (Figure 3.4), we relied on previously reported values for the measurement variance, Σ_ϵ, in the data set that we considered

(Manning et al., 2019). When users seek to apply our algorithm to other data where previously published values are not available for Σ_ϵ, this parameter can be inferred following the same procedure as reported in Manning et al. (2019).

Our algorithm opens up the investigation of research problems, such as cell-to-cell heterogeneity in dynamic gene expression, which would previously not have been accessible. In future applications, our algorithm may provide a non-invasive method to measure the kinetic parameters of the gene of interest, such as the translation and transcription rates, or properties of the gene's promoter, which are described by the repression threshold and Hill coefficient parameters in our model. On experimental data sets where multiple, qualitatively different dynamics are observed (D'Urso & Brickner, 2017; Elowitz et al., 2002; Godwin et al., 2017), our algorithm may provide insights into the mechanistic origin of these different dynamics, by identifying differences in inferred parameter values between the observed cells or cell populations. In order to classify whether observed differences between posterior distributions are significant, one can construct the posterior distribution describing the difference between parameter values from both cells or populations, and test whether the credible interval of that distribution contains zero (Gelman et al., 2020). To facilitate such analysis, our method may for example be combined with clustering algorithms on the time series data, such as Gaussian mixture modelling. Since different dynamic patterns of gene expression have been observed in multiple studies of auto-repressing transcription factors (Manning et al., 2019; Soto et al., 2020), we anticipate that these approaches will spark new scientific investigations.

Throughout, we have assumed that measurements in the form of protein copy numbers per nucleus are available over time. To collect such data, it is necessary to combine endogenous fluorescent reporters with FCS in order to translate reporter intensity values to molecule concentrations. Future versions of our algorithm may be applied to data where FCS is not available, if one extends our measurement model (F, Section 3.4.2) to include an unknown, linear scaling parameter between protein copy numbers and imaged intensity values.

We highlight that the impact of this work is not limited to a single gene in a single model system. The conceptual framework and derivations described here are applicable to any system which can be described by delayed stochastic differential equations, although there may be computational limitations as model sizes increase.

Ethics statement

Animal experimentation: All animal work was performed under regulations set out by the UK Home Office Legislation under the 1986 United Kingdom Animal Scientific Procedures Act.

Data accessibility

All code and data used in this article are freely available on our GitHub repository,

https://github.com/kursawe/hesdynamics.

Authors' contributions

J.B., C.M., M.R., N.P. and J.K. conceived the study and contributed to the manuscript. J.B. developed and implemented the algorithm, ran the simulations, performed the analysis, and produced the figures. J.K. contributed to the algorithm development. M.R., N.P. and J.K. supervised the study. J.B. and J.K. co-wrote the manuscript with input from C.M., M.R. and N.P.. C.M. provided microscopy data and FCS analysis. All authors gave final approval for publication and agree to be held accountable for the work performed therein.

Competing interests

We declare we have no competing interests.

Funding

This work was supported by a Wellcome Trust Four-Year PhD Studentship in Basic Science to J.B. (219992/Z/19/Z) and a Wellcome Trust Senior Research Fellowship to N.P. (090868/Z/09/Z). C.M. was supported by a Sir Henry Wellcome Fellowship (103986/Z/14/Z) and a University of Manchester Presidential Fellowship. M.R.'s work was supported by a Wellcome Trust Investigator Award (204832/B/16/Z).

Acknowledgements

The authors would like to acknowledge the use of the Computational Shared Facility at The University of Manchester.

Chapter 4

Continuous-time filtering and variational inference of combined single cell time-series data

Joshua Burton, Magnus Rattray, Nancy Papalopulu and Jochen Kursawe

4.1 My contribution

I designed the study, in particular, I proposed the use of variational inference for performance gains and determined the suitable algorithm to use for the model. I also proposed the idea to redesign the algorithm in order to account for continuous time delays. I developed and implemented the Kalman filter algorithm, with helpful design suggestions from J.K.. I wrote the initial draft of the manuscript and redrafted it with helpful comments and suggestions from J.K., N.P. and M.R.. I wrote all the code for *in-silico* data simulation, creation of figures, Bayesian inference models and subsequent analysis. I developed the software package for the Kalman filtering algorithm and deployed all online documentation relating to the package.

4.2 Publication status

We plan to submit this paper for publication in 2023.

Abstract

Expression dynamics of a large number of genes are influenced by noise and negative feedback. These dynamics, such as oscillations, are increasingly linked to cell fate decisions. However, the underlying mechanisms that control these dynamics are not well understood. Stochastic delay differential equations offer a powerful framework for modelling gene expression dynamics over time, but accurately inferring the parameters of these models from time-series data remains a computational challenge. Previous methods lead to uncertain estimates with data from a single cell, and increasing the amount of available data makes inference computationally prohibitive. Here, we extend our previously implemented Bayesian method in order to more efficiently infer kinetic parameters of gene expression using time-series data from multiple cells. We reformulate our computational implementation to allow estimation of the gradient of the data likelihood with respect to the delay parameter so that we can apply a variational inference algorithm to infer parameters for clusters of cells with qualitatively similar dynamics, and better utilise Hamiltonian Monte Carlo (HMC) methods. We find that by combining *in-silico* time-series data from up to 40 simulated live-imaging experiments, we are able to reduce the standard deviation of all inferred parameters by at least a factor of 4 in comparison to single experiments, and several orders of magnitude faster than HMC. By enabling fast and accurate inference of time-series data from a large number of cells, our method opens up new opportunities for identifying kinetic parameters of gene expression that regulate cell fate decisions in biological systems.

Keywords: Parameter inference, Bayesian methods, variational inference, Gene expression oscillations, MCMC, Kalman filters

4.3 Introduction

Oscillations and other dynamic patterns of gene expression are indicators of cell state, and changes in dynamic gene expression have been shown to precede differentiation across multiple organisms, including mice and zebrafish (Imayoshi et al., 2013; Manning et al., 2019; Purvis et al., 2012; Soto et al., 2020). In each of these biological systems, stochastic interactions between transcription factors and DNA may influence the dynamics of gene expression, causing homogeneous populations to diversify into different cell states (McAdams & Adam Arkin, 1997; Raj & van Oudenaarden, 2008; Spudich & Koshland, 1976). Understanding how interactions, or kinetic parameters of gene expression, regulate dynamic gene expression will allow us to uncover critical mechanisms which influence heterogeneity during development. However, simultaneously observing multiple types of molecular interaction at single-cell resolution is technically infeasible, and we are typically limited to capturing the concentration of a few specific proteins over time using live imaging techniques. It is therefore vital to have computational methods for determining kinetic parameters of gene expression from the currently available data.

Experimentally observed gene expression dynamics can be approximated using a generative process, for example, a differential equation model, and then combined with Bayesian inference methods in order to determine kinetic parameters from single-cell time-series data. In order for this approach to be effective, it is essential that the model is able to capture the observed dynamic behaviour.

In a number of biological systems, a protein represses the expression of its own gene and, in combination with delays inherent to biological processes, can exhibit a wide array of dynamic behaviour including ultradian oscillations in both protein and mRNA concentration (Hirata et al., 2002; Jensen et al., 2003; Lewis, 2003; Takebayashi et al., 1994). In order to represent these ultradian oscillations mathematically, models need to account for both stochasticity and delays between different interactions, in the form of stochastic delay differential equations. One mathematical representation of this biological motif is the delay-mediated auto-repressive feedback loop (Burton et al., 2021; Heron et al., 2007; Lewis, 2003; Purvis et al., 2012). In this work we model the delay term as a constant, in contrast to previous approaches to inference in delayed SDE models, e.g. (Calderazzo et al., 2019), that use distributed delays.

In this paper, the parameters of this model are inferred from data using Bayesian methods. In particular, we aim to find the *posterior distribution* of the model parameters $\boldsymbol{\theta}$ given some data $\boldsymbol{y}$,

$$\pi\left(\boldsymbol{\theta} \mid \boldsymbol{y}\right) \propto \pi\left(\boldsymbol{y} \mid \boldsymbol{\theta}\right) \pi(\boldsymbol{\theta}), \tag{4.1}$$

where $\pi(y \mid \theta)$ is the likelihood function, which represents the probability of the data the model parameters, and $\pi(\theta)$ is the prior, a distribution which quantifies our belief in the likely values for each parameter in the absence of data. Prior beliefs about the parameter values can be informed by experiments and reported values in the literature, as well as by physical constraints, for example requiring positive rate parameters. Bayesian methods are used specifically to quantify uncertainty in the inferred parameter estimates and provide a range of reasonable parameter values, as opposed to a point estimate.

Bayesian inference methods for uncovering the rates of e.g. transcription, translation and degradation, using the delay-mediated auto-repressive feedback loop have been developed but are not widely used (Heron et al., 2007). In our previous work (Burton et al., 2021), we adapted a delay-adjusted Kalman filtering approach (Calderazzo et al., 2019) to infer the kinetic parameters of our model using single-cell time-series data. This approach has two main components. First, the true (unobserved) expression of individual genes is approximated from noisy observations by combining a model of gene expression and a measurement model. This is achieved with a delay-adjusted Kalman filter and is illustrated by Figure 4.1A. (see Section 4.4.2 for the derivation). From this procedure, a likelihood function is defined. Second, Markov chain Monte Carlo (MCMC) methods are used to sample from the posterior distribution to identify the most suitable model parameters given the original experimental observations. In particular, the Metropolis-adjusted Langevin algorithm (MALA), a gradient-based sampling algorithm, was used, which takes advantage of the gradient of the likelihood with respect to the model parameters to improve convergence speed (Betancourt, 2017; Girolami & Calderhead, 2011).

For simple problems, i.e. where only one or two parameters are unknown, this approach is effective, and we have shown with *in-silico* experiments that model parameters can be accurately inferred using simulated data (Burton et al., 2021). When extending this approach to infer multiple kinetic parameters of gene regulation simultaneously for experimental data, we found that the uncertainty in our estimates was too high to identify differences between cells with qualitatively different dynamic expression (Burton et al., 2021). One way to overcome this shortcoming is to simply use more data. In typical applications, time series data from multiple cells are collected for a given experimental condition. Cells with similar dynamics can be clustered together (Manning et al., 2019) and assumed to have the same or similar kinetic properties. We have previously shown that in our system of interest, an increase in the length of simulated time series data improves the accuracy of inference (Burton et al., 2021). Increasing the length of time series data is equivalent to combining multiple datasets, under the assumption that they were produced by an equivalent generating process with the same parameter values. We can therefore perform inference on combined time series from a number of cells simultaneously, either by summing the log-likelihoods given by eq. (4.1), or within a hierarchical model, to improve our parameter estimates and avoid overfitting (Gelman et al., 2013). Hierarchical modelling has been extensively applied to combined single cell time series data. For example, to improve estimates of the timing of switching between transcriptional states across a whole tissue, as well as rates of mRNA degradation for a large number of genes (Featherstone et al., 2016; Jenkins et al., 2013). Single cell time series filtering approaches (Calderazzo et al., 2019) have been extended with a spatial hierarchical model that improves parameter estimates for individual cells by ‘borrowing strength’ from combined data (Unosson et al., 2021).

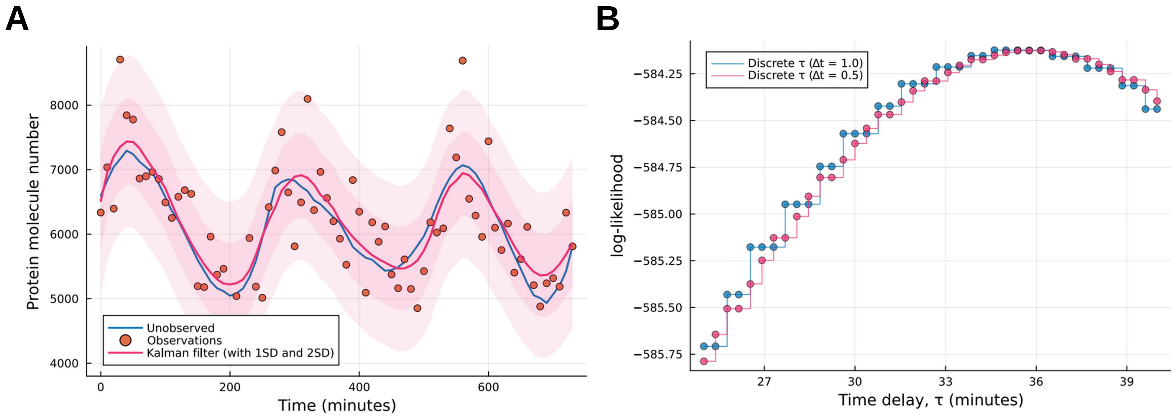


Figure 4.1: **The delay-adjusted Kalman filter accurately recaptures unobserved states from noisy observations. A.** Simulated experimental data. Protein molecule numbers are simulated using the chemical Langevin equation (see Section 4.4.1, blue line). Experimental observations are made every 10 min by adding Gaussian noise to the protein molecule number (orange dots). The Kalman filter recaptures the unobserved state using the experimental observations and parameters used to generate them (pink line). The parameter values used were $P_0 = 3407.99$, $h = 5.17$, $\mu_m = \log(2)/30$, $\mu_p = \log(2)/90$, $\alpha_m = 15.86$, $\alpha_p = 1.27$ and $\Sigma_\epsilon = 382,104$. Confidence regions calculated from the estimated covariance are also shown (pink shaded areas). **B.** Log-likelihood values for a range of time-delay values, with two separate discretisations of the state space. Accuracy improves as Δt decreases, however, there is no gradient information since τ is not continuous.

In our previous implementation of the delay-adjusted Kalman filter (Burton et al., 2021), sampling from the posterior distribution was too slow to deal with data from multiple cells – combining time-series data from a cluster of a typical size of 10–40 cells would require weeks of calculation time, which is unfeasible for day-to-day use. In order to fully utilise the gradient-based sampling approach that we introduced, we have to be able to compute the gradients of the likelihood with respect to *all* model parameters. Since our state-space approximation of the unobserved gene expression dynamics was discretised over time, the gradient of the likelihood with respect to the time delay parameter τ was not defined. Therefore, although we could improve the accuracy of our method with a finer discretisation, we were not able to benefit from the improved performance of gradient-based sampling. Figure 4.1B illustrates this point. The log-likelihood values for varying τ are shown using two different discretisations of the state space approximation (blue and pink lines). Although the accuracy will improve as the discretisation becomes finer, the gradients remain undefined, as indicated by the flat lines between each data point.

Here, we address these drawbacks by introducing a new method that is able to produce accurate, approximate posterior distributions, and is up to two orders of magnitude faster than advanced MCMC for similar problems. This opens up the ability to infer parameters using a cluster of cells at a fraction of the computational cost of our previous approach.

We present a redesigned version of our previously published delay-adjusted Kalman filter (Burton et al., 2021) where the time delay parameter is now continuous, meaning the gradient of the likelihood with respect to τ is defined (see Section 4.4, Methods). This allows us to take full advantage of the variational inference algorithm Pathfinder (Zhang et al., 2022) to sample from an approximation of the target distribution. Pathfinder has been evaluated on a wide range of posterior distributions, outperforms automatic differentiation variational inference (ADVI) and is comparable to short chains of dynamic Hamiltonian Monte Carlo (HMC) (Zhang et al., 2022), a widely used form of MCMC (Betancourt, 2017; M. Hoffman & Ma, 2020; M. D. Hoffman & Gelman, 2014). Further, it requires multiple orders of magnitude fewer log density and gradient evaluations for challenging posteriors (Zhang et al., 2022).

We validate the use of variational methods by comparing posteriors from both Pathfinder and dynamic HMC for test cases, e.g. using synthetic data and only inferring a few parameters. We find that posterior densities from both approaches overlap, indicating very similar distributions. We sample posterior distributions for varying amounts of *in-silico* time-series data simulated using the same parameter values, in order to emulate a cluster of single cells. We show that posterior distributions can be obtained for clusters of up to 40 cells in less than an hour of computation time, and demonstrate a significant reduction in parameter uncertainty as the size of a cluster increases.

4.4 Methods

In this section, we extend the methods introduced in Burton et al. (2021) to account for continuous time delays, in order to enable gradient-based variational inference. First, we briefly re-introduce the delay-adjusted Kalman filter and negative feedback chemical reaction network that models our experimental system. Then, we show how the Kalman filter can be modified so that the time delay is modelled as a continuous parameter. Lastly, we introduce a variational inference method, which enables us to obtain posterior distributions faster than previously possible by several orders of magnitude.

4.4.1 Simulating data from the negative feedback chemical reaction network

We model gene expression dynamics using the negative feedback chemical reaction network, which is defined by

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \alpha_m f(p(t-\tau)) - \mu_m m + \sqrt{\alpha_m f(p(t-\tau)) + \mu_m m}\xi_m, \tag{4.2}$$

$$\frac{\mathrm{d}p}{\mathrm{d}t} = \alpha_p m - \mu_p p + \sqrt{\alpha_p m + \mu_p p}\xi_p, \tag{4.3}$$

where ξ_m and ξ_p are temporally uncorrelated, independent Gaussian white noises, $m(t)$ and $p(t)$ represent concentrations of mRNA and protein at time t and

$$f(p(t-\tau)) = \frac{1}{1 + \left(\frac{p(t-\tau)}{P_0}\right)^h} \tag{4.4}$$

is an activating Hill function. The Hill coefficient h, and repression threshold P_0, both determine the relationship between protein concentration and mRNA production. A higher Hill coefficient implies greater increases in repression of transcription in response to changes in protein concentration. The repression threshold defines the amount of protein needed to reduce transcription to half of the basal rate. The transcriptional time delay parameter, τ, represents the duration of the transcription process. We assume the delay is constant, rather than being defined by a distribution. The degradation rates for mRNA and protein are denoted μ_m and μ_p respectively, α_m is the basal transcription rate in the absence of protein, and α_p is the translation rate.

To simulate experimental time-series data we solve the Chemical Langevin equations (eqs. (4.2) and (4.3)) over a given time span $(0, T)$ with parameters $\boldsymbol{\theta} = (P_0, h, \mu_m, \mu_p, \alpha_m, \alpha_p, \tau)$, discarding the first 1000 minutes so that the time series are independent of the initial condition, which is chosen to be $[30, 500]$. To solve the SDDE we use an explicit Runge-Kutta discretization of the strong order 1.0 Milstein method with adaptive time-stepping (Mil'shtejn, 1975). We then select protein values every $t_{\text{obs}} = 10$ minutes, to obtain an array of *true* protein values $p_{\text{true}} \in \mathbb{R}^N$, where N is the desired number of data points. We set the measurement variance to be $\Sigma_\epsilon = 0.1\hat{x}_{\text{d}}$, where $\hat{x}_{\text{d}}$ is the variance of p_{true}. This is consistent with (Manning et al., 2019), where the measurement variance has been estimated to contribute 10% of the total variance in this system. We construct the synthetic data-set p_{observed} by adding a sample from a normal distribution with zero mean and variance $\Sigma_\epsilon \in \mathbb{R}$ to each data point in p_{true}, e.g.

$$p_{\text{observed}}(i) = p_{\text{true}}(i) + \varepsilon(i),$$

where

$$\varepsilon(i) \sim \text{Normal}\left(0, \Sigma_\epsilon\right),$$

for $i \in 1, \ldots, N$.

4.4.2 The delay-adjusted Kalman filter provides a likelihood function

We are specifically interested in being able to infer the parameters $\boldsymbol{\theta} = (P_0, h, \mu_m, \mu_p, \alpha_m, \alpha_p, \tau)$ of the negative feedback chemical reaction network defined by eqs. (4.2) and (4.3). Formally, we want to use Bayesian inference methods to find the posterior distribution $\pi(\boldsymbol{\theta} \mid \mathbf{y})$. In order to do this we need to define a closed-form likelihood function. This is achieved by the delay-adjusted Kalman filter (Burton et al., 2021; Calderazzo et al., 2019), which approximates the true (*unobserved*) state, $\mathbf{X}(t) \in \mathbb{R}^d$, of a stochastic dynamical system of the form

$$\begin{aligned} d\mathbf{X}(t) &= g(\mathbf{X}(t))dt + f\left(\mathbf{X}(t-\tau)\right)dt \\ &\quad + \sqrt{l(\mathbf{X}(t)) + q\left(\mathbf{X}(t-\tau)\right)}dB(t), \end{aligned} \tag{4.5}$$

where $B(t)$ is a d-dimensional Wiener process, using (potentially noisy) observations $\mathbf{y} = \{y(t_1), \ldots, y(t_n)\}$, linked to the true state by a measurement model

$$y(t_i) = F\mathbf{X}(t_i) + \epsilon_i, \quad i = 1, \ldots, n, \tag{4.6}$$

where $\epsilon_t \sim \mathcal{N}(0, \Sigma_\epsilon)$, $F \in \mathbb{R}^{1\times d}$. In the case of our system of interest, $F = [0, 1]$ and $\mathbf{X}(t_i) = [m(t_i), p(t_i)]^T$. The parameter Σ_ϵ is the measurement variance and represents any noise introduced through the experimental measurement process. For ease of notation, we define $\mathbf{X}(t) = \mathbf{X}_t$ and $\{y(t_1), \ldots, y(t_N)\} = y_{1:N}$.

We initialise the procedure at negative times with the steady-state solutions to the equivalent deterministic model (i.e. where $\xi_m, \xi_p = 0$). The state space mean is initialised with the steady-state solution, and the state space variance is initialised with a multiple of this mean. Specifically, the variance of the mRNA is initialised at 20 times its mean value, and the variance of the protein is initialised at 100 times its mean value. Off-diagonal elements of the state space covariance are initially zero, as in Burton et al. (2021).

The delay-adjusted Kalman filter then approximates the likelihood by iteratively inferring a multivariate Gaussian probability distribution with mean

$$\rho_t = \mathbb{E}\left[\mathbf{X}_t \mid y_{1:t-1}\right]$$

and covariance

$$P_{t,s} = \text{Cov}\left(\mathbf{X}_t, \mathbf{X}_s \mid y_{1:t-1}\right),$$

over the true state at each observation time point, by simulating the dynamical system between consecutive observations (the *prediction step*) and then updating the distribution at each observation (the *update step*).

For the prediction step, since the system state is governed by an SDDE, this requires deriving the delay differential equations that govern both ρ_t and $P_{t,s}$, which is done using the linear noise approximation (Van Kampen, 1992) (see Section 3.4.2 for details):

$$d\rho_t = g(\rho_t)dt + f(\rho_{t-\tau})dt, \tag{4.7}$$

$$dP_t = \left[J_g(\rho_t)P_t + P_t^T J_g(\rho_t)^T\right]dt + \left[J_f(\rho_{t-\tau})P_{t-\tau,t}\right]dt + \left[P_{t,t-\tau}J_f(\rho_{t-\tau})^T\right]dt + A\left(\rho_t, \rho_{t-\tau}\right)dt, \tag{4.8}$$

$$dP_{s,t} = P_{s,t}J_g^T(\rho_t)dt + P_{s,t-\tau}J_f^T(\rho_{t-\tau})dt, \tag{4.9}$$

where $A(\rho_t, \rho_{t-\tau}) = l(\rho_t) + q(\rho_{t-\tau})$. These equations can then be solved with any ODE solver (see Section 4.4.4).

Our prediction step using the ODE solver gives us an *a priori* state estimate since it does not incorporate the state of the system at the current observation time point which has now been reached. In the update step of the Kalman filter, we derive an expression for the mean and variance of the state space distribution $\pi(\mathbf{X}_{t-\tau:t} \mid y_{1:t})$. Crucially, given that our system of interest involves a delay τ, all state space estimates back to time $t-\tau$ are updated at time t. The updated state space mean and variance, denoted $\rho^*_{t-\tau:t}$ and $P^*_{t-\tau:t}$, are given by:

$$\rho^*_{t-\tau:t} = \rho_{t-\tau:t} + C(y - F\rho_t), \tag{4.10}$$

$$P^*_{t-\tau:t} = P^*_{t-\tau:t} - CFP_{t,t-\tau:t}, \tag{4.11}$$

where

$$C = P_{t-\tau:t,t}F^T(FP_tF^T + \Sigma_\epsilon)^{-1},$$

and $P_{t-\tau:t,t} = [P_{t-\tau,t}, P_{t-\tau+\Delta s,t}, \ldots, P_{t,t}]$, where Δs is the off-diagonal step size.

At the end of this filtering procedure, the likelihood can be evaluated by

$$\pi(\mathbf{y} \mid \boldsymbol{\theta}) = \prod_{i=1}^{n} \mathcal{N}(y_i \mid \mu = F\rho_i(\boldsymbol{\theta}),\, \sigma^2 = FP_{i,i}(\boldsymbol{\theta})F^T + \Sigma_\epsilon), \tag{4.12}$$

where $\rho_i(\boldsymbol{\theta})$ and $P_{i,i}(\boldsymbol{\theta})$, which are dependent on $\boldsymbol{\theta}$, are taken directly from the solutions to eqs. (4.7) to (4.9), rather than their updated counterparts.

Our prior distributions for each parameter are outlined in Section B.1

4.4.3 Continuous time delays via ODE interpolation

We solve eqs. (4.7) to (4.9) using the DifferentialEquations.jl Julia package (Rackauckas & Nie, 2017). We use either the forward Euler method with linear interpolation, or the 4th order Runge–Kutta method with 3rd order interpolation (see Section 4.4.4), so that the solution is a continuous function of t. Specifically, interpolation is done automatically via the DifferentialEquations.jl package and allows us to interrogate the solution of our equations at any time t over a given time span.

Solving Equation (4.7) requires access to past values for the state space mean, but is otherwise straightforward. However, calculating $P(t,t)$ using eq. (4.8) at each observation time point for any $\tau > 0$ is more involved. This is due to the fact that Equation (4.8) requires that covariance terms $P(t, t-\tau)$ are known for any t. In other words, we need to be able to interrogate $P(t,s)$ for any t, $s > 0$. In our previous work (Burton et al., 2021) we solved this problem by discretising the state space variance and only allowing values of τ which were multiples of this discretisation value. Here, we define an alternative procedure to solve this problem by employing interpolation methods. Our procedure contains three distinct steps, which we will first explain for $\tau \geq t_{\text{obs}}$, where t_{obs} is the observation time step in our data.

First, solving Equation (4.8) requires that we know $P(t-\tau, t)$ for any $\tau \in (0,t)$. These *off-diagonal* covariances, i.e. $P(s,t)$ where $s \neq t$, are solved over the time-span $t \to t + t_{\text{obs}}$ for $s \in (t-\tau, t)$. We define the off-diagonal step size Δs and partition $(t-\tau, t)$ into $S = [t-\tau, t-\tau+\Delta s, \ldots, t-\Delta s]$, and solve each of the off-diagonal DDE's using eq. (4.9) to obtain a set of solutions

$$\{P(s_i, t) \to P(s_i, t + t_{\text{obs}}) \mid s_i \in S\}\,.$$

Each of these solutions can use the interpolation provided by the DifferentialEquations.jl package. In other words, the second argument of P is continuous. To get $P(s,t)$ for any s we use linear interpolation over the set $\{P(s_i, t) \mid s_i \in S\}$.

Second, we solve the diagonal covariance DDE given by eq. (4.8) up to the next observation time-point to obtain the solution

$$\{P(t,t) \rightarrow P(t+t_{\text{obs}}, t+t_{\text{obs}})\}\,.$$

Lastly, we solve a final set of off-diagonal variances to obtain the set of solutions

$$\{P(s_i,s_i) \rightarrow P(s_i, t+t_{\text{obs}}) \mid s_i \in S\}\,.$$

If $\tau < t_{\text{obs}}$, this three-step procedure is repeated multiple times, solving each DDE over time spans of length τ rather than t_{obs}, until the next observation time point is reached.

We calculate the updated mean ρ_t^*, variance $P^*(t,t)$, and covariance $P^*(s,t)$ at time t given the most recent observation at time t_c via

$$\rho^*(t) = \rho(t) + C(t)(y - F\rho(t_c)), \tag{4.13}$$

$$P^*(s,t) = P(s,t) - C(s)FP(t_c,t) \tag{4.14}$$

where

$$C(s) = P(s,t_c)F^T(FP(t_c,t_c)F^T + \Sigma_\epsilon)^{-1},$$

(see section B.3 for a full derivation). The key advantages of our updated approach are that the time delay parameter τ is a continuous variable and that we can use higher-order solvers which promise increased speed and accuracy.

The implementation of our updated approach is available on GitHub, and online documentation with tutorials are also available at https://burtonjosh.github.io/DelayedKalmanFilter.jl/dev/.

4.4.4 Description of ODE solvers

Within the delay-adjusted Kalman filter, we solve multiple DDEs at each prediction step. To accurately calculate the likelihood and obtain reliable posterior distributions, it is important to ensure that any error introduced in the numerical solution of these equations is minimised. Since DDEs can be thought of as a discontinuous initial value problem, we can choose any ODE solver for our method (ZivariPiran & Enright, 2010). Consider the initial value problem defined by

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t,y), \quad y(t_0) = y_0. \tag{4.15}$$

We want to know y at some time t_{N} in the future. The simplest method for achieving this is the (forward) Euler method. If $y(t_n) = y_n$, then

$$y_{n+1} = y_n + \Delta t f(t_n, y_n), \tag{4.16}$$

where Δt is called the *step size*. Although this method is efficient, the error in the estimate at any given time is proportional to the step size.

If this error is high we can decrease the step size, Δt, but this leads to many evaluations of f. We can use a more accurate solver, which will be more expensive per step but may require fewer steps in total. One popular choice is the 4th order *Runge–Kutta* method. Here we have

$$y_{n+1} = y_n + \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)\Delta t, \tag{4.17}$$

where

$$\begin{aligned}
k_1 &= f(t_n, y_n),\\
k_2 &= f\left(t_n + \frac{\Delta t}{2}, y_n + \Delta t\frac{k_1}{2}\right),\\
k_3 &= f\left(t_n + \frac{\Delta t}{2}, y_n + \Delta t\frac{k_2}{2}\right),\\
k_4 &= f\left(t_n + \Delta t, y_n + \Delta t k_3\right).
\end{aligned}$$

The local truncation error in this estimate is on the order of $\mathcal{O}((\Delta t)^5)$ and the global truncation error is on the order of $\mathcal{O}((\Delta t)^4)$.

4.4.5 Variational inference using Pathfinder

In order to infer the parameters of our model for a combination of datasets we use Pathfinder, a variational inference algorithm which approximates a target posterior distribution (Zhang et al., 2022). The aim of Pathfinder is to locate an approximation to the posterior density along a quasi-Newton optimisation path.

Pathfinder is initialised with a random sample θ_0 from the prior, and then L-BFGS optimisation is used to generate a trajectory $\{\theta_0, \theta_1, \ldots, \theta_L\}$ towards a local maximum of the posterior density. For each θ_l, a local normal approximation of the posterior distribution with mean $\mu^{(l)}$ and variance $\Sigma^{(l)}$ is generated, using the gradient and curvature information collected along the optimisation trajectory.

Pathfinder then selects the approximation which minimises the Kullback-Leibler (KL) divergence to the target density,

$$l^* = \operatorname*{argmin}_l \mathrm{KL}\left[\mathrm{MVNormal}\left(\theta \mid \mu^{(l)}, \Sigma^{(l)}\right) \parallel p(\theta \mid y)\right]. \tag{4.18}$$

To achieve this, it is equivalent to maximise the evidence lower bound (ELBO). With draws from the local approximations, $\phi^{(1)}, \ldots, \phi^{(K)} \sim \mathrm{MVNormal}(\mu^{(l)}, \Sigma^{(l)})$, the ELBO can be approximated with Monte Carlo by

$$\begin{aligned} &\mathrm{ELBO}\left[\mathrm{MVNormal}\left(\mu^{(l)}, \Sigma^{(l)}\right) \parallel p(\theta \mid y)\right] \\ &\qquad \approx \frac{1}{K}\sum_{k=1}^{K} \log p(\phi^{(k)}) - \log(\mathrm{MVNormal}(\phi^{(k)} \mid \mu^{(l)}, \Sigma^{(l)})). \end{aligned} \tag{4.19}$$

Finally, M approximate posterior samples are drawn from the distribution that minimises the KL divergence, $\mathrm{MVNormal}\left(\mu^{(l^*)}, \Sigma^{(l^*)}\right)$.

4.5 Results

We aim to infer kinetic parameters of gene regulation by combining time-series data from multiple cells that show similar dynamics. In this section, we evaluate the performance of our method on *in-silico* data. We simulate collected time series from single cells and use variational inference to infer parameters with combined data from multiple simulated cells. First, we characterize the numerical convergence of our likelihood function using different ODE solvers. We illustrate that higher-order solvers significantly increase the speed of this calculation, and identify an optimal step size for fast and accurate inference. We then validate our updated computational implementation by inferring parameters of the auto-repressive feedback motif using the No-U-Turn sampler (NUTS), a dynamic HMC algorithm. We show that the variational inference algorithm Pathfinder identifies similar posterior distributions to those obtained with MCMC, at a decreased computational cost. Finally, we apply this new approach to the combined data of multiple cells with similar dynamics, which leads to a significant decrease in parameter uncertainty.

4.5.1 Higher order solvers can improve both accuracy and computational efficiency.

There is a trade-off between accuracy and run time in our algorithm (Algorithm 2). We aim to obtain accurate posterior distributions, which means our likelihood calculation needs to be accurate.

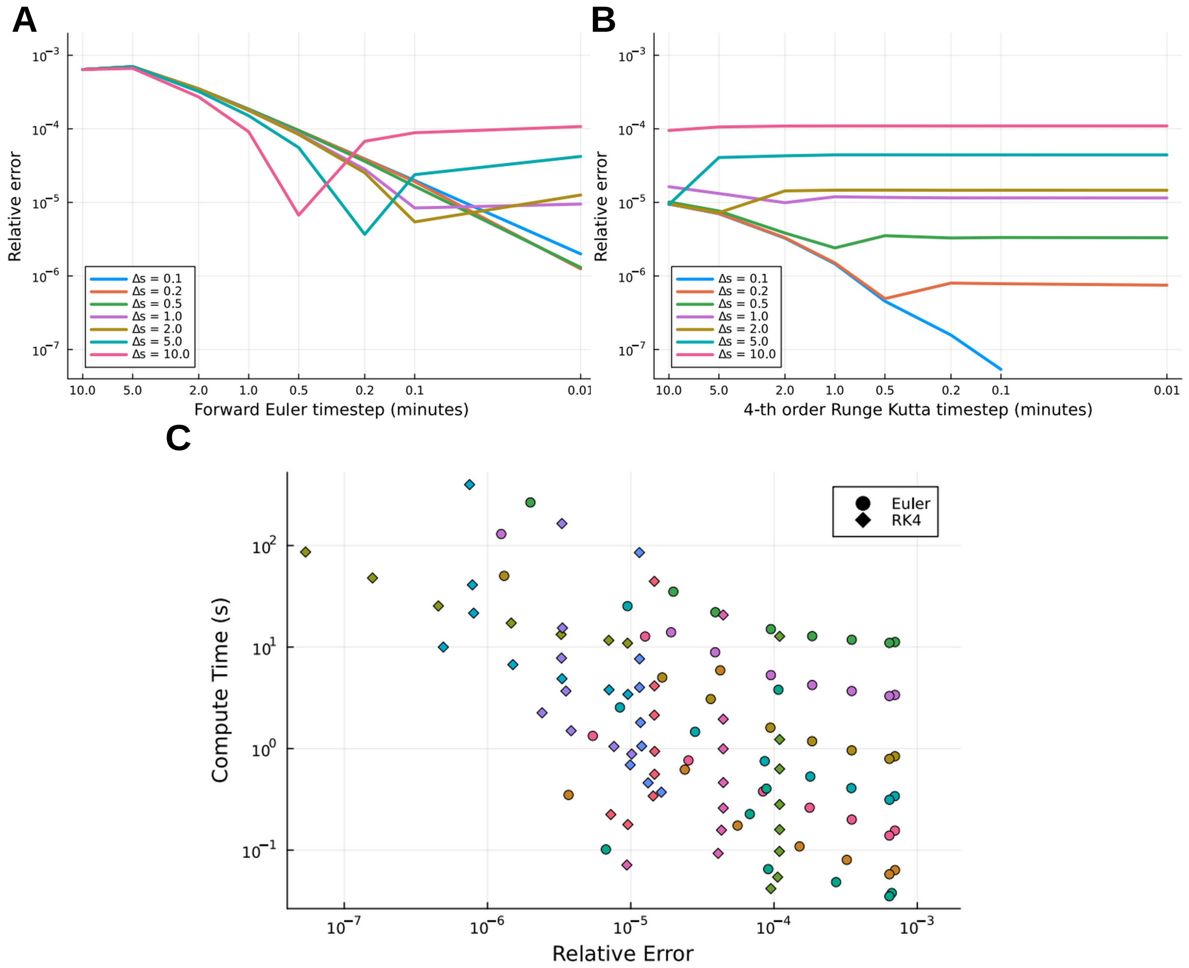


Figure 4.2: **Analysis and comparison of solver configurations allow us to optimise performance. A.** The relative error in the calculated log-likelihood value for a range of solver step sizes Δt, and off-diagonal step sizes Δs, using the forward Euler ODE solver. The error is calculated relative to the likelihood value when using the RK4 solver with $\Delta t = 0.01$, $\Delta s = 0.1$. **B.** Same as A with the 4th order Runge–Kutta solver. In both A and B the accuracy of the likelihood for each value of Δs plateaus when Δt decreases. For the RK4 solver, this happens for much larger values of Δt, allowing for the same accuracy with fewer function evaluations. **C.** A plot of the relative errors (shown in A and B) against the median run-time of each likelihood evaluation in seconds. To obtain a smaller relative error, the step sizes need to be smaller, leading to a longer run-time. Each data point corresponds to a unique solver configuration. The colour scheme is the same as in A and B.

This can be achieved by using higher-order ODE solvers or decreasing the numerical parameters Δs (the *off-diagonal step size*) and Δt (the ODE solver step size) in our algorithm (Algorithm 2). As part of the algorithm, we are required to approximate the covariance of gene expression $P(s,t)$ at arbitrary time combinations s and t. This covariance is related to the correlation of gene expression between the time points s and t. To approximate $P(s,t)$, both arguments of this covariance are discretised into equally sized intervals of length Δs and Δt, respectively. Thus, the accuracy of our approximation depends on the order of the approximation as well as the step size. Further details are available in section Section 4.4.4. In particular, the order of the approximation is determined by the choice of ODE solver inside our method (Forward Euler or Runge-Kutta). Our aim is to identify a solver configuration for our method which optimises both accuracy and computational efficiency.

We first investigated the performance of our method by testing how the off-diagonal step size, Δs, and solver step size, Δt, influence the log-likelihood if different ODE solvers are used. For an accurate representation of the *true* log-likelihood value, we used the RK4 solver with $\Delta s = 0.1$ and $\Delta t = 0.01$. This was the benchmark value for which all of the relative errors were calculated (Figure 4.2A,B).

We found that in general, solutions obtained with the RK4 solver required fewer time steps in total to achieve a given accuracy than the Forward Euler solver. For the forward Euler solver, we found that a step size of Δt greater than 0.1 was typically not sufficient to converge for any Δs (Figure 4.2A). In contrast, the RK4 solver converged typically when $\Delta t = \Delta s$ (Figure 4.2B). For both solvers, decreasing Δt had a limited impact on the total relative error. In order to decrease the relative error, it was also necessary to decrease Δs. We found that the relative error was typically lower when using the RK4 solver. The forward Euler had errors as large as 6×10^{-4}, whereas relative error did not exceed 1×10^{-4} for the RK4 solver.

To assess the trade-off between speed and accuracy we computed the median run times of each of the above configurations and compared them to the relative error (Figure 4.2C). For identical step size combinations, RK4 is slower than Forward Euler, as expected. For example, for an off-diagonal step size of $\Delta s = 0.1$ and solver step size of $\Delta t = 0.01$, the log-likelihood calculation took 301 and 824 seconds for forward Euler and RK4 respectively. In contrast, for an off-diagonal step size of $\Delta s = 10.0$ and solver step size of $\Delta t = 10.0$ the time taken was 0.034 and 0.043 seconds for the forward Euler and RK4 respectively.

To assess the potential speed up for each solver, we calculated the *relative runtime* l_b/l_t of the median time taken for each likelihood calculation l_t, against a benchmark likelihood calculation time l_b, when $\Delta s = 1.0$ and $\Delta t = 1.0$, which was the discretisation we used in our previous implementation (Burton et al., 2021). We found that we were able to achieve up to 15.7 and 25.6 times speedups for the forward Euler and RK4 solvers respectively if larger step sizes were used (Tables 4.1 and 4.2).

Although the fastest configuration using the RK4 was slightly slower than the forward Euler solver using the same combination of Δs and Δt ($\sim 25\%$ slower), it was still very fast (0.043 seconds) and the relative error was an order of magnitude lower. Given these results, we decided to perform further *in-silico* experiments by calculating the likelihood using the RK4 solver, with $\Delta s = \Delta t = 10.0$. This ensured that our calculation run times were short, allowing for faster inference while maintaining high accuracy in our likelihood calculation. To

		Forward Euler step size			
		Δt = 0.01	Δt = 0.1	Δt = 1.0	Δt = 10.0
Off-diagonal step size	Δs = 0.1	0.0017548	0.0140161	0.0386144	0.0465365
	Δs = 1.0	0.0208933	0.189342	1.0	1.67295
	Δs = 10.0	0.143718	1.31861	8.21313	15.6934

Table 4.1: Relative runtimes for a single log-likelihood calculation using the forward Euler method, using the $\Delta s = \Delta t = 1.0$ condition as the benchmark. For example, the calculation is over 15 times faster when $\Delta s = \Delta t = 10.0$.

		Runge–Kutta step size			
		Δt = 0.01	Δt = 0.1	Δt = 1.0	Δt = 10.0
Off-diagonal step size	Δs = 0.1	0.0017548	0.0140161	0.0386144	0.0465365
	Δs = 1.0	0.0134368	0.142397	1.0	2.86899
	Δs = 10.0	0.0876649	0.895599	6.65484	25.6409

Table 4.2: Relative runtimes for a single log-likelihood calculation using the 4th order Runge–Kutta method, using the $\Delta s = \Delta t = 1.0$ condition as the benchmark. For example, the calculation is over 25 times faster when $\Delta s = \Delta t = 10.0$.

assess the consistency of this accuracy, we computed the relative error of the likelihood for 150 different parameter combinations. The relative error was typically less than 1×10^{-3} around the mode of the distribution, i.e. for large log-likelihood values (> -650), and around 1×10^{-2} in the tails (see Figure B.1). Although we sacrifice some accuracy in the tails of the distribution with this configuration, we expect to accurately identify areas of high posterior probability since absolute errors are small around the mode (< 0.65).

4.5.2 Dynamic HMC can be used to perform inference on single datasets for multiple parameters

To confirm that our solver configuration would produce accurate posterior distributions, we tested our new software implementation by inferring parameters for a single synthetic data set. This data set was simulated using the procedure described in Section 4.4.1, with parameter values $P_0 = 3407.99$, $h = 5.17$, $\mu_m = \log(2)/30$, $\mu_p = \log(2)/90$, $\alpha_m = 15.86$, $\alpha_p = 1.27$ and $\Sigma_\epsilon = 382,104$. These parameter values were chosen for the oscillatory dynamics they exhibit in the model. In order to quickly test our approach and easily diagnose

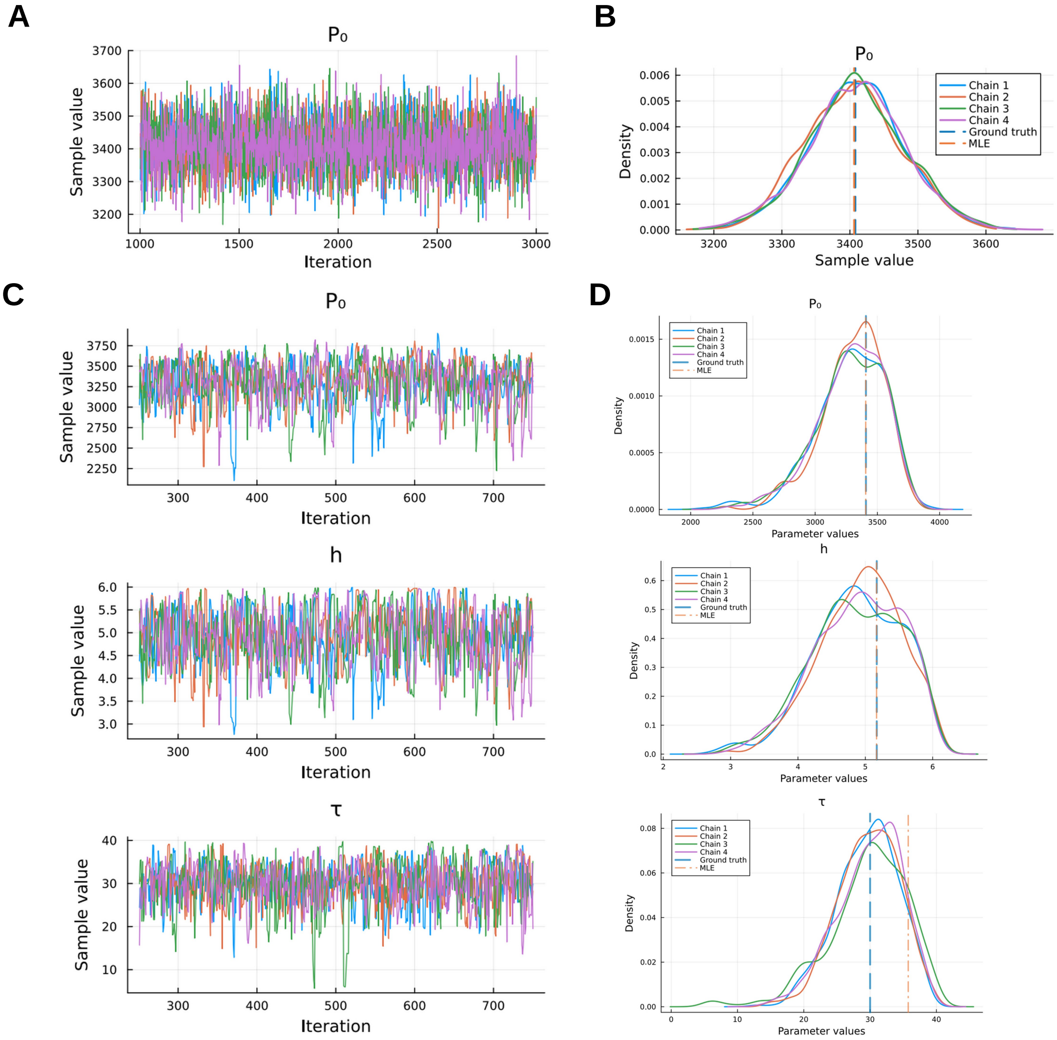


Figure 4.3: **Dynamic HMC can be used to infer multiple parameters for single datasets accurately. A.** For the simulated dataset shown in Figure 4.1A, the repression threshold parameter P_0 is inferred using HMC. The MCMC traceplot is shown here for $2,000$ iterations after warm-up. **B.** A density plot made from the samples shown in (A) (blue). The ground truth (blue line) and maximum likelihood estimate (orange line) are also shown. **C., D.** Inference is done on the same dataset for parameters P_0, h and τ simultaneously. Traceplots are shown for 500 iterations after warm-up, and density plots are shown with the ground truth and maximum likelihood estimates shown (solid pink and dashed orange lines respectively). In each case, we chose the initial value of the Markov chains by performing maximum likelihood estimation with the L-BFGS optimiser and set the target acceptance rate of the NUTS sampler to $\delta = 0.8$. We sampled 8000 posterior draws across 4 chains (3000 draws per chain, discarding the first 1000 draws as burn-in) for A, and we sampled 2000 posterior draws across 4 chains (750 draws per chain, discarding the first 250 draws as burn-in) for C.

any potential problems, we first considered two simple scenarios; where only P_0 is unknown (Figure 4.3A,B), and where the three parameters P_0, h, and τ are unknown (Figure 4.3C,D). These scenarios were chosen to allow for comparisons with the previous implementation (Burton et al., 2021), as well as to see how run time scaled as the number of parameters inferred was increased. We sample 4 chains in parallel from the posterior distribution using the No-U-Turn sampler (NUTS) via the Turing.jl Julia package (Ge et al., 2018; M. D. Hoffman &

Gelman, 2014). The chains are initialised with the maximum likelihood estimate, calculated using the L-BFGS optimiser. The prior distributions used for each parameter are detailed in Section B.1.

The $\hat{R}$ statistic provides the ratio of the between- and within-chain estimates of the parameters. Values close to 1 indicate that these estimates agree and that we are likely to be sampling from the posterior distribution. The effective sample size (ESS) measures the sampling efficiency in the bulk of the distribution, with high values indicating reliable mean and median estimates. Values of $\hat{R} < 1.01$ and ESS > 400 per parameter reassure us that our posterior estimates are reliable (Vehtari et al., 2020). Table 4.3 summarises the 8000 posterior draws obtained using dynamic HMC. We computed an $\hat{R}$ statistic of 1.0007 and an ESS of 2775.32. The inference procedure accurately recaptures the true value of the repression threshold, P_0, with the mean of the posterior distribution being less than 0.015 standard deviations from the true value. The Monte Carlo standard error (MCSE), which estimates the noise in our estimation, is also negligible in comparison to the standard deviation of the Markov chain.

Parameter	**True value**	**MLE**	**Mean**	**SD**	**MCSE**	$\hat{R}$	**ESS**
repression threshold, P_0	3407.99	3406.26	3409.03	71.18	1.50	1.0007	2775

Table 4.3: Summary statistics of the Markov chain shown in Figure 4.3A. The columns refer to the true parameter value, maximum likelihood estimate, posterior mean, posterior standard deviation, Monte Carlo standard error of the Markov chain, $\hat{R}$-statistic and the effective sample size respectively. The true value of P_0 is contained within the 10% highest density interval, indicating the mass of the distribution is close to the ground truth.

To infer three parameters simultaneously (Figure 4.3C,D) we again chose the initial value of the Markov chains by performing maximum likelihood estimation with the L-BFGS optimiser and sampled 2000 posterior draws across 4 chains.

Inferring P_0, h, and τ introduces greater uncertainty (Table 4.4) in our estimates, in comparison to the single parameter case (Table 4.3). This can be seen by the increase in standard deviation for the repression threshold, P_0. However, the mean estimates are still within half a standard deviation for all parameters, and the MCSE is less than 5% of the standard deviation. The $\hat{R}$ and ESS values indicate that our estimates of the posterior distribution are reliable.

Although our chains are initialised with a maximum likelihood estimate, this does not always accurately recapture the true value, i.e. the MLE value for τ is 35.75, whereas the posterior mean is 29.78. Figure B.3A shows positive correlation between τ and h when only considering simulated data from a single cell, which may be the cause of the difference between the ground truth and the maximum likelihood estimate. This highlights the need for us to account for uncertainty in the data, and Bayesian inference methods provide one approach for doing this in a rigorous manner.

Parameter	True value	MLE	Mean	SD	MCSE	$\hat{R}$	ESS
repression threshold, P_0	3407.99	3406.32	3286.63	270.57	8.81	1.0013	676.57
Hill coefficient, h	5.17	5.17	4.90	0.63	0.014	1.0016	681.64
time delay, τ	30.0	35.75	29.78	5.04	0.16	1.0007	940.54

Table 4.4: Summary statistics of the Markov chain shown in Figure 4.3C. The definitions of the columns are the same as in Table 4.3.

Parameter	True value	Mean	SD	MCSE	$\hat{R}$	ESS
repression threshold, P_0	3407.99	3369.8	267.2	3.50	1.0	8137
Hill coefficient, h	5.17	4.89	0.69	0.0077	1.0	8185
time delay, τ	30.0	30.35	5.17	0.058	1.0	7822

Table 4.5: Summary statistics of the samples obtained using the Pathfinder algorithm. The definitions of the columns are the same as in Table 4.3

4.5.3 Pathfinder approximates posteriors with similar accuracy to HMC

The total compute duration to obtain 2000 samples from the posterior distribution using data from a single cell across 4 chains was 5.45 hours per thread, using a laptop with an Intel® Core™ i7-8750H CPU with 16GiB of RAM, running Ubuntu 20.04.5 LTS. In many contexts, it is desirable to sample posterior distributions for a large number of cells or to combine time-series data from multiple cells to further reduce uncertainty in our estimates (Burton et al., 2021). Run-time will scale linearly with the amount of data, meaning for example that pos-

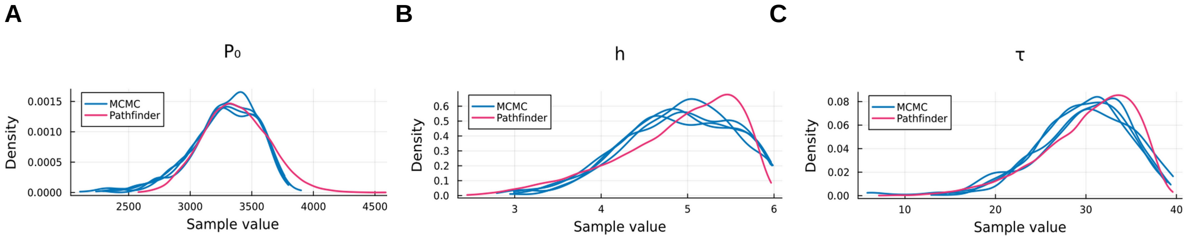


Figure 4.4: **Variational inference allows for fast and accurate inference of multiple parameters. A–C.** Marginal posterior densities from using HMC (blue lines, Figure 4.3D), alongside marginal posterior densities computed using 8000 samples from a single run of Pathfinder (pink lines). The two densities overlap, indicating very similar posterior distributions.

terior inference for a combined cluster of 30 cells would take ∼1 week, and multiple days of computing resources may not always be available.

Variational inference methods are a potential way to speed up inference. In order to evaluate the accuracy and extent of the speed-up available, we used the Pathfinder algorithm to sample from an approximate posterior distribution (Figure 4.4). The summary statistics of the sampled distribution, which are shown in Table 4.5, are comparable to when using NUTS to sample directly. The standard deviations for each parameter are nearly identical, and the means are very similar. The total run-time of this approach on the same laptop was 93.48 seconds on a single thread, a speed-up of 839 times (Table 4.6, first column).

4.5.4 Pathfinder enables fast inference on combined datasets

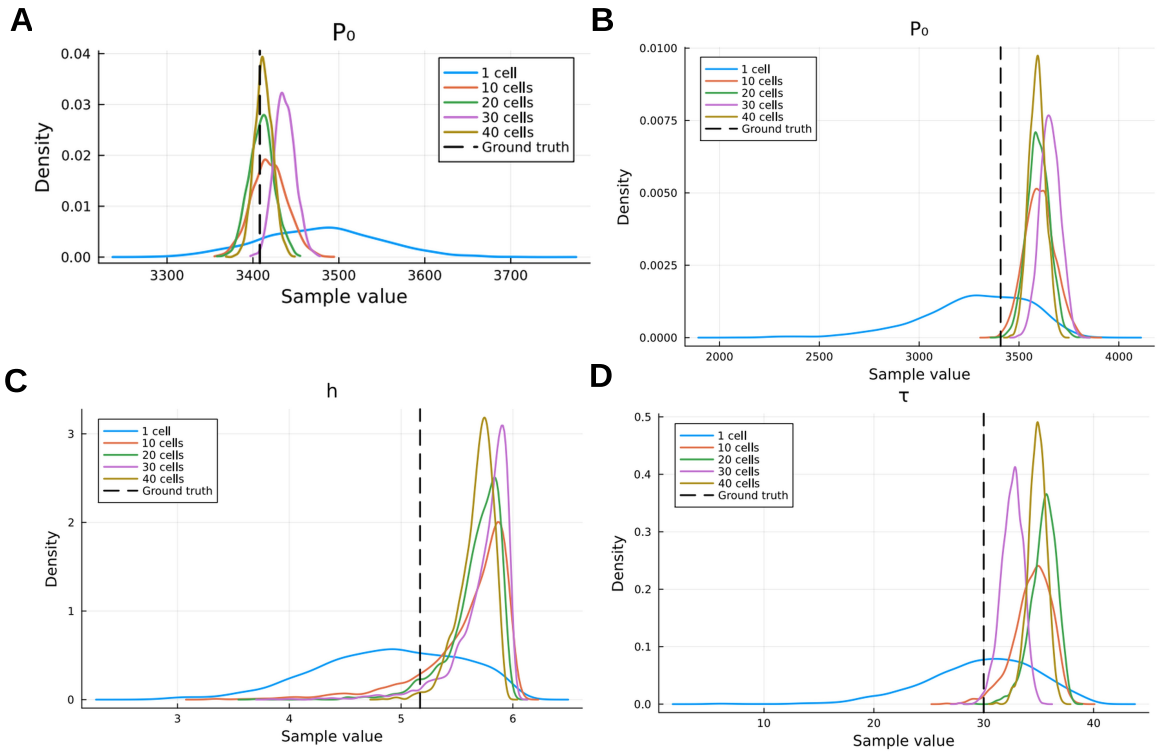


Figure 4.5: **Parameter inference on multiple simulated datasets simultaneously using Pathfinder. A.** The repression threshold is inferred using Pathfinder for varying numbers of cells. As the number of cells increases from 1–40, the standard deviation of the posterior reduces from 69.2 to 10.5. **B–D.** Inferring the same three parameters as in Figure 4.3B–D, but with varying numbers of cells. As the number of cells increases from 1–40, the standard deviation of each parameter is reduced by a factor of (B) 6.51 for the repression threshold P_0, (C) 4.15 for the Hill coefficient h, and (D) 6.05 for the transcriptional time delay τ. The compute duration for performing variational inference over 40 cells was 40 minutes.

In our previous work, we compared posterior distributions from multiple cells in the same cluster and found that for some parameters the uncertainty was too high to identify differences across clusters (Burton et al., 2021). However, given the computational requirements of MCMC sampling, it is not practical to perform inference on combined time-series data from a cluster of 10–40 cells. Therefore, after confirming that variational inference gave similar results to using MCMC directly on simulated data from one cell, we went on to investigate how the posteriors obtained using variational inference changed as simulated time-series data from an increasing number of cells were combined.

To do this, we simulated 40 *in-silico* time-series, using the same approach as previously outlined in Section 4.4.1, with the same parameter values as in Figure 4.3, apart from Σ_ϵ which was chosen to be $360,000$ to simulate noise within a biologically realistic regime (Manning et al., 2019). We calculate the joint likelihood of data from multiple cells by summing each of the likelihoods of the data from the individual cells. This provides us with a framework for interrogating the effect of inference accuracy as the number of cells in a cluster increases.

We first inferred a single parameter, P_0, for 1, 10, 20, 30, and 40 cells respectively (Figure 4.5A). The standard deviation in our estimate when using data from only 1 cell was 69.2, which was reduced to 10.5 when using the combined data from 40 cells. The posterior mean also significantly improved, from 3478.7 when using 1 cell to 3412.3 with 40 cells, where the ground truth was 3407.99.

Having confirmed that variational inference accurately identifies the true parameter in this simple case, we extended our model to infer three parameters simultaneously, P_0, h, and τ (Figure 4.5B–D). The search space for this problem is much larger and introduces additional complexity, given that we have previously identified a strong correlation between P_0 and h (Burton et al., 2021). Despite this, inferring the parameters was extremely fast, taking less than 40 minutes using *in-silico* time-series data from 40 cells (Table 4.6).

We found a drastic improvement in accuracy when incorporating data from multiple simulated cells in this way. For data from 1 cell, the standard deviation for P_0 was 267.2 (Table 4.5), and this decreased to 41.5 with 40 cells. To contextualise the significance of this improvement, the size of the 95% highest posterior density interval (HPDI) for P_0 changes from 995 to 161, a reduction of over 6 times (Figure 4.4B). Similarly, we observe a reduction of 4.1 and 5.8 times in the 95% HPDIs for h and τ respectively (Figure 4.4C,D). Although the posterior means do not seem to converge to the true parameter values as the number of cells increases, their deviation is not large enough that we see different dynamics in our posterior predictive checks (Figure B.2). These checks show that key aspects of the data (period, mean expression) are accurately captured by the inferred parameters, despite a strong correlation between the repression threshold P_0 and Hill coefficient, h (Figure B.3, (Burton et al., 2021))

Number of cells	1	10	20	30	40
Time taken (minutes)	1.55	11.9	33.5	32.4	39.8
RMSE ($\mathbf{P_0}$)	296.5	208.8	196.8	252.5	187.7
SD ($\mathbf{P_0}$)	267.2	74.2	55.8	49.6	41.5

Table 4.6: Time taken to infer P_0, h, and τ simultaneously using variational inference for *in-silico* time-series data from a varying number of cells. The root-mean-square error (Section B.2) for P_0 is also shown. These values are shown for h and τ in Table B.1.

4.6 Discussion

Our main aim in this work was to extend our previously published statistical tool to be able to identify key properties of an auto-negative feedback loop using combined data from a cluster of cells. We have achieved this aim by extending all model parameters to continuous space and employing state-of-the-art variational inference methods. We were able to improve the accuracy and speed of the delay-adjusted Kalman filter by introducing higher-order ODE solvers. We demonstrated that variational inference gave comparable results to advanced HMC methods, and was almost three orders of magnitude faster when inferring three unknown model parameters simultaneously. We were able to use this approach to interrogate how parameter uncertainty changed as the amount of data increased.

Variational inference using the Pathfinder algorithm allows us to sample from an approximation to the true posterior distribution with orders of magnitude fewer log-density and gradient evaluations of the target distribution. Pathfinder has been tested on a diverse set of posteriors, and its accuracy ranges from slightly worse to much better than short chains of HMC (Magnusson et al., 2022). To ensure this approximation was faithful in our model, we compared samples from Pathfinder to samples obtained using dynamic HMC for a simple test case, where one data set was used to infer three parameters simultaneously. We found that when combining the samples from each approach, the resulting $\hat{R}$ and ESS values indicated that the convergence and summary statistics (mean and SD) were unchanged. Given these results, we do not expect our approximation to deviate significantly from the true posterior. We then further investigated the use of Pathfinder by simulating time-series data equivalent to that collected from 40 cells. Assuming the same parameter values, we were able to infer the transcriptional time delay τ, the Hill coefficient h and the repression threshold P_0. These parameters were chosen specifically because they are difficult to measure experimentally, and interact via a Hill function, often making them difficult to separate. Despite this non-linearity, we were able to accurately infer these parameters and recapture the dynamic behaviour using posterior predictive checks.

Although the inferred parameters did not converge to the true values as the number of cells increased (Figure 4.5B–D), the errors in our estimates are not large enough to identify a different dynamic regime (Figure B.2). To reduce unwanted errors in the estimates, Pathfinder can be run I times to produce I normal approximations. Then Pareto-smoothed importance resampling (see Algorithm 5 of Zhang et al. (2022)) from the mixture of these normals can provide a robust approximation to posteriors which are far from normal, e.g. multi-modal posteriors, or which have complex correlation structures. Investigating the effect of this resampling procedure on inference accuracy may be required to prevent bias in posterior distributions. Some amount of bias introduced by a variational inference approach is to be expected. It may be possible to investigate the nature of this bias using simulation based calibration, i.e. sampling from multiple posteriors using simulated data from multiple prior initialisations. The bias in parameter estimates across all of the posteriors can then be analysed, to see if it is consistent across parameter space, or perhaps follows some kind of distribution. Diagnostics to assess variational inference have been proposed using this approach, which may enable us to assess the average bias of a point estimate (Yao et al., 2018).

Although we were able to accurately estimate specific model parameters under the condition that simulated cells share the same parameters, it may be desirable to allow cell-specific parameter values under a hierarchical model. This would relax the current assumption that kinetic parameters within a cluster of cells are identical, as well as the assumption that *qualitatively* similar dynamics reflect similar parameter values. Hierarchical Bayesian modelling of biological regulation processes with delays have been used with improved results over non-hierarchical counterparts (Cortez et al., 2021). Due to the implementation of our inference framework and the use of the Turing.jl MCMC package, extending to a hierarchical model would make for a straightforward next step in this work.

Another potential for inaccuracies in our inference comes from the ODE solver configuration in the Kalman filtering prediction step. In order to achieve the speed-up we have reported, the step size in our solver has been increased. This will inevitably introduce some numerical error, however, we have ensured that the relative error in our log-likelihood calculation is smaller than 10^{-3} around the mode. In our tests on simulated data, we were able to recapture model parameters, and, apart from an increase in run-time, our results were unchanged when we increased the accuracy of our solvers.

We implemented our statistical method with the Julia programming language. The main motivation for this choice was the advanced differential equations ecosystem provided by DifferentialEquations.jl. For the same level of accuracy in our likelihood calculations, we found a speed-up when using higher-order solvers in our Kalman filtering algorithm. A further advantage was the ease of implementation of automatic differentiation methods, which provide the likelihood gradients used for optimisation (Revels et al., 2016). This means that gradients don't have to be derived and implemented for new models and that the size of the code-base was reduced by over 35% compared to directly implementing the gradients, substantially improving readability. We have not been able to identify any other Kalman filtering software packages that apply to SDDE models, and other software packages do not allow the choice of higher-order ODE solvers. To encourage collaboration in the wider open-source scientific programming community, our code is hosted publicly on GitHub, and tutorials and documentation are available at https://burtonjosh.github.io/DelayedKalmanFilter.jl/dev/.

The improved accuracy of our parameter estimates will allow us to determine parameters most likely associated with different dynamic patterns of expression. For example, we may find that a high translation rate, or more stable mRNA, is linked to oscillatory expression that maintains the progenitor state. This improved ability to distinguish these patterns will allow us to make biological predictions which can be validated experimentally. The potential application of this method is broad, applying to any system governed by an auto-repressive negative feedback loop. This approach can also be easily extended to analyse other biological systems that can be modelled using stochastic differential equations.

Data accessibility

All code and data used to produce the figures in this article are freely available on our GitHub repository, https://github.com/burtonjosh/continuous_kf_figures. The DelayedKalmanFilter.jl software package is available at https://github.com/burtonjosh/DelayedKalmanFilter.jl. Online documentation with tutorials for the package are also available at https://burtonjosh.github.io/DelayedKalmanFilter.jl/dev/.

Authors' contributions

J.B. conceived the study and wrote the initial draft of the manuscript. M.R., N.P. and J.K. provided supervision throughout the study and gave comments and suggestions on the manuscript. J.B. developed and implemented the algorithm, ran the simulations, performed the analysis, and produced the figures. J.K. contributed to the design of the Kalman filtering algorithm. All authors gave final approval for publication and agree to be held accountable for the work performed therein.

Competing interests

We declare we have no competing interests.

Funding

This work was supported by a Wellcome Trust Four-Year PhD Studentship in Basic Science to J.B. (219992/Z/19/Z) and a Wellcome Trust Senior Research Fellowship to N.P. (090868/Z/09/Z). M.R.'s work was supported by a Wellcome Trust Investigator Award (204832/B/16/Z).

Chapter 5

Sequential and additive expression of miR-9 precursors control timing of neurogenesis

Ximena Soto, **Joshua Burton**, Cerys S. Manning, Thomas Minchington, Robert Lea, Jessica Lee, Jochen Kursawe, Magnus Rattray and Nancy Papalopulu

5.1 My contribution

This was a collaborative project, where X.S. led the experimental part of the study and I led the theoretical part. I proposed the use of the perfect adaptation model, as well as its extension, via an incoherent feed-forward loop, and suggested this model as a potential mechanism for cell state changes. With helpful discussion and feedback from J.K., X.S., T.M., N.P. and M.R., I designed and performed all analyses of the model and determined parameter values to highlight the suggested potential mechanism. I wrote all code related to both mathematical models and wrote all code to produce Figures C.5, C.6 and 5.6, with the exception of panel A of Figure 5.6 which was provided by X.S.. I contributed to the manuscript by writing Sections 5.4.6 and 5.6.13 and also contributed to the review and editing of the manuscript.

5.2 Publication status

This paper was accepted in *Development* on 26th August 2022 and was published on 3rd October 2022 (Soto et al., 2022).

Abstract

MicroRNAs (miRs) have an important role in tuning dynamic gene expression. However, the mechanism by which they are quantitatively controlled is unknown. We show that the amount of mature miR-9, a key regulator of neuronal development, increases during zebrafish neurogenesis in a sharp stepwise manner. We characterize the spatiotemporal profile of seven distinct microRNA primary transcripts (pri-mir)-9s that produce the same mature miR-9 and show that they are sequentially expressed during hindbrain neurogenesis. Expression of late-onset pri-mir-9-1 is added on to, rather than replacing, the expression of early onset pri-mir-9-4 and -9-5 in single cells. CRISPR/Cas9 mutation of the late-onset pri-mir-9-1 prevents the developmental increase of mature miR-9, reduces late neuronal differentiation and fails to downregulate Her6 at late stages. Mathematical modelling shows that an adaptive network containing Her6 is insensitive to linear increases in miR-9 but responds to stepwise increases of miR-9. We suggest that a sharp stepwise increase of mature miR-9 is created by sequential and additive temporal activation of distinct loci. This may be a strategy to overcome adaptation and facilitate a transition of Her6 to a new dynamic regime or steady state.

Key words: pri-mir-9, miR-9, Neurogenesis, Zebrafish, Temporal control

5.3 Introduction

MicroRNAs (miRs) are a class of small (∼22 nt) regulatory noncoding RNAs, which regulate gene expression at the posttranscriptional level. These small RNAs are processed from large microRNA primary transcripts (pri-mir) into 70∼90 nt precursors (pre-mir) before further splicing into ∼22 nt mature miR. miR-9 is a highly conserved miR that is expressed predominantly in the central nervous system (CNS) of vertebrates and plays a crucial role during CNS development. Specifically, previous work in Xenopus, zebrafish and mice has shown that miR-9 is essential for cell fate transitions during neurogenesis (Bonev et al., 2011; Bonev et al., 2012; Coolen et al., 2013; Shibata et al., 2011). miR-9 post-transcriptionally targets many transcription factors that are involved in neural development such as FoxG1 (Shibata et al., 2008), Tlx (also known as Nr2e1) (Zhao et al., 2009) and members of the Hes/Her helix-loop-helix family of transcription factors, including Hes1 in mouse and Xenopus (Bonev et al., 2011; Bonev et al., 2012) and Her6/Her9 in zebrafish (Coolen et al., 2012; Galant et al., 2016; Leucht et al., 2008; Soto et al., 2020).

The Hes/Her family of proteins is expressed dynamically in an oscillatory manner at the ultradian timescale (Hirata et al., 2002; Shimojo et al., 2008). Hes/Her oscillations are achieved by a negative feedback loop, whereby Hes/Her proteins inhibit their own transcription coupled with a rapid turnover of protein and mRNA. Instability of both protein and mRNA allows for levels of the protein to fall, de-repression to occur and expression to resume, generating a cyclic pattern (Hirata et al., 2002; Novák & Tyson, 2008). Indeed, both mRNAs and proteins of Hes family genes are unstable: for example, in mice, the half-life of Hes1 mRNA is ∼24 min, the Hes1 protein half-life is in the order of 22 min (Hirata et al., 2002) and the Her6 (Hes1 zebrafish orthologue) protein half-life is ∼12 min (Soto et al., 2020).

Instability of mRNA, as well as translation of protein, are partly controlled by miRs. Indeed, our previous work revealed that miR-9 regulation is important for controlling Hes1 mRNA stability and allowing the oscillatory expression of Hes1 to emerge (Bonev et al., 2012; Goodfellow et al., 2014). We have recently shown that in zebrafish, the dynamics of Her6 protein expression switch from noisy to oscillatory and then to downregulation, and that these changes coincide temporally with the onset of miR-9 expression in the hindbrain (Soto et al., 2020). When the influence of miR-9 on *her6* is removed experimentally, Her6 expression does not evolve away from the 'noisy' regime and is not downregulated with a consequent reduction in progenitor differentiation. We have interpreted this to mean that the miR-9 input is necessary to constrain gene expression noise, enabling oscillations to occur and to be decoded by downstream genes, which in turn participate in downregulating Her6 as cells differentiate (Soto et al., 2020).

However, not only the presence of miR-9 but also the amount of miR-9 present is important, as too much or too little miR-9 can lead to dampening of Hes1 oscillations (Bonev et al., 2012; Goodfellow et al., 2014). Indeed, mathematical modelling showed that increasing miR-9 over time drives the Hes1 expression into different states (oscillatory or stable high/low) and that the amount of miR-9 present in the cell determines the length of time for which Hes1 oscillates, effectively timing the transition to differentiation (Goodfellow et al., 2014; Phillips et al., 2016). Together these findings support that Hes/Her dynamics and downregulation are sensitive to the amount of mature miR-9 present in the cell; however, the mechanism by which the miR-9 level is controlled is not known.

This question is complicated by the observation that vertebrates (and some invertebrates) possess multiple copies of the miR-9 gene at distinct loci, which are all capable of producing the same mature miR. For example, both human and mouse contain three copies of miR-9 (Rodriguez-Otero et al., 2011; Shibata et al., 2011) and frogs have four (Walker & Harland, 2008). Due to an additional round of whole-genome duplication (WGD) in teleost fish (Amores et al., 1998; Jaillon et al., 2004), zebrafish have seven paralogues of miR-9 (pri-mir-9-1 to pri-mir-9-7) (P. Y. Chen et al., 2005).

One possibility is that different genomic loci contribute to miR-9 regulation in a qualitative way, with differential temporal and spatial specificity of mature miR-9 expression. Indeed, there is some limited evidence that these discrete copies of miR-9 are expressed differentially during development both temporally and spatially (Nepal et al., 2016; Tambalo et al., 2020). Another, and yet unexplored, possibility is that transcription from different loci may serve to control miR-9 quantitatively, that is to increase the amount of miR-9 in the cell and perhaps do so in a temporally controlled manner, thus contributing to the change of miR-9 levels that is necessary to drive a change in the dynamics of Hes/Her targets.

Here, we undertake a systematic study of pri-mir-9 expression in zebrafish that aims to address the likelihood of these distinct scenarios, with special attention to the possibility of a quantitative control mechanism. We show by *in situ* hybridization that the expression of miR-9 spreads from the forebrain to the hindbrain and increases quantitatively in the hindbrain between 24 and 48 h postfertilisation (hpf). A detailed time course of the expression of all seven pri-mir-9 paralogues shows that they are all transcriptionally active, but exhibit subtle, yet distinct, temporal and spatial profiles. Focusing on a set of early- and late-expressed pri-mir-9s in the hindbrain (pri-mir-9-1, pri-mir-9-4 and pri-mir-9-5) by quantitative single molecule fluorescent *in situ* hybridisation at single cell level, we found that, surprisingly, in many cells, early and late pri-mir-9s were concurrently transcriptionally active such that the expression from late-activated pri-mir-9s is added on to the early ones. This is functionally significant as the specific mutation of the late pri-mir9-1 selectively reduces neurons that normally differentiate late. Our mathematical modelling suggests that the sharp quantitative increase afforded by the deployment of additional transcriptional units, may facilitate the downregulation of Her6 at late time points. We found this to be consistent with a subtle but reproducible failure to downregulate Her6 at late stages when pri-mir-9-1 was specifically mutated. Taken together, although both quantitative and qualitative mechanisms may contribute to the decoding function of mature miR-9s, we found a previously unappreciated quantitative component in the deployment of pri-mir-9s, which is temporally controlled and in turn controls the evolution of Her6 dynamic expression over time.

5.4 Results

5.4.1 Pri-mir-9s are expressed with differed temporal onset

miRs are derived from a duplex precursor and the -5p strand (‘guide’) is preferentially incorporated into an RNA-induced silencing complex (RISC) to exert its regulatory functions, while the complementary -3p strand (‘passenger’) is thought to be rapidly degraded. Indeed, for the mature miR-9 the miR-9-5p is designated as the ‘guide’ strand and its annotation is derived from the mature miR sequence being embedded in the 5’ stem of the miR-9 precursor.

To investigate the expression of the mature miR-9 (-5p strand) in zebrafish embryos, we first performed a whole-mount *in situ* hybridization (WM-ISH) for the mature miR-9 using a locked nucleic acid (LNA) probe. Mature miR-9 was detected only in the forebrain at 24 hpf (Figure 5.1A), but at 30 hpf miR-9 was weakly observed in the midbrain and rhombomere (r) 1 of the hindbrain, maintaining high expression in the forebrain (Figure 5.1A, 30 hpf). As development progressed, miR-9 expression increased in the hindbrain with steady high levels in the forebrain (Figure 5.1A, 35-38 hpf; blue arrow). Later in development, levels in the hindbrain were further increased, while those in the forebrain were decreased (Figure 5.1A, 48 hpf; blue arrow, hindbrain; green arrow, forebrain). These results show a temporally controlled increase of miR-9 expression along the brain/hindbrain axis as previously described in Soto et al. (2020).

To characterize the increase of expression in the hindbrain in a quantitative manner, we used quantitative real-time PCR (RT-qPCR) on dissected hindbrains from stages 25 hpf to 48 hpf. This analysis confirmed that there is upregulation in the time frame analysed. An initial low level of expression at 30 hpf was followed by a sharp upregulation at 37 hpf, which was maintained through to 42 hpf, undergoing a second sharp increase at 48 hpf (Figure 5.1B).

In zebrafish, the mature miR-9 can be produced from seven paralogues of miR-9. The miR-9 paralogues occupy seven unique loci across the genome (GRCz11; Genome Reference Con-

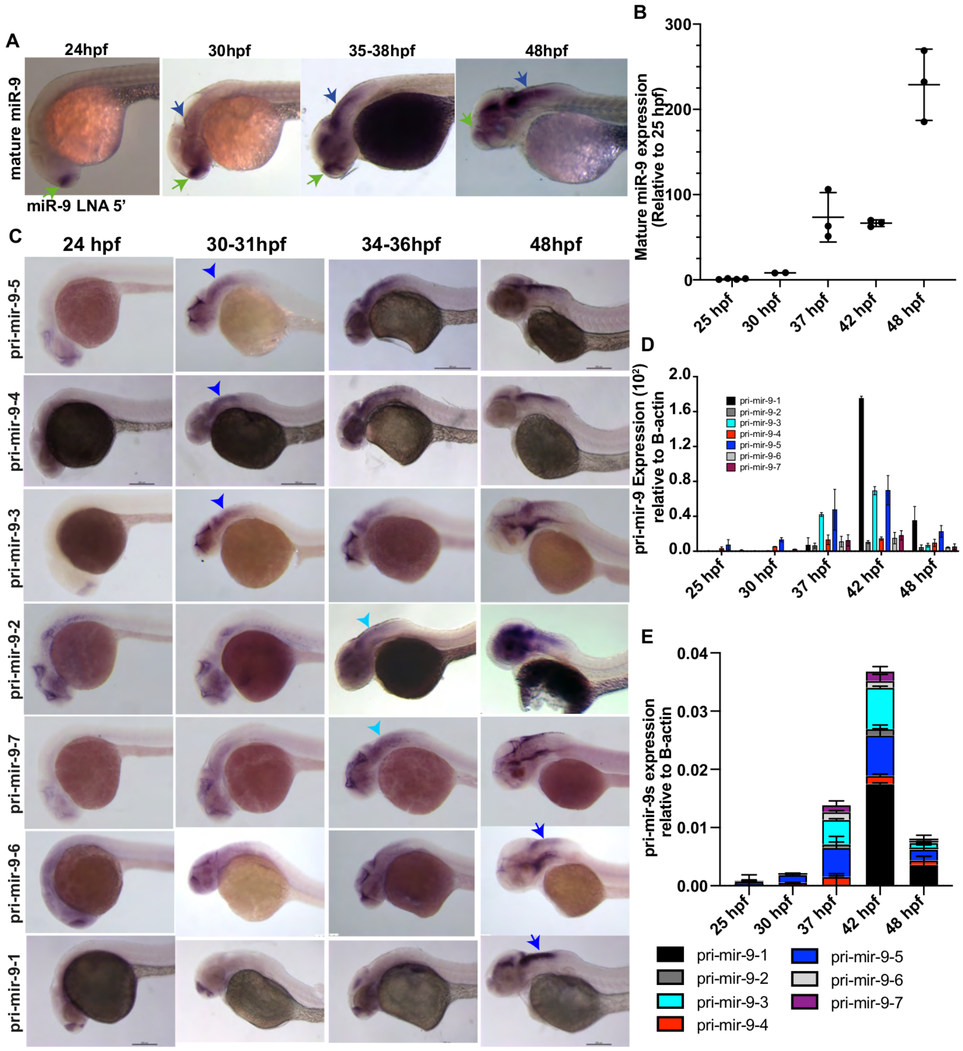


Figure 5.1: **Pri-mir-9 paralogues are expressed with different temporal onset. A.** Representative example of chromogenic whole-mount *in situ* hybridization (WM-ISH) of miR-9 using miR-9 LNA 5'-Dig observed at different stages during development. Similar results can be observed in Soto et al. (2020). Longitudinal view, anterior to the left. Green arrow, forebrain expression; blue arrow, hindbrain expression. **B.** Taqman RT-qPCR of mature miR-9 from dissected hindbrain at different stages of development, relative to 25 hpf. Horizontal bars indicate the median with 95% confidence intervals. **C.** Chromogenic WM-ISH of different pri-mir-9s using specific probes for each paralogue observed at different stages during development. Longitudinal view, anterior to the left. Blue arrowhead, expression in hindbrain at 30-31 hpf; light blue arrowhead, expression in hindbrain at 34-36 hpf; blue arrow, expression in hindbrain at 48 hpf. **D,E.** SYBR green RT-qPCR relative quantification of the seven pri-mir-9s from dissected hindbrains at different stages of development. Quantification was normalised using β-actin. Data are mean±s.d. (B,D,E) $N = 3$, each N contains a pool of 10 hindbrains. Scale bars: 200 μm.

sortium Zebrafish Build 11) (Yates et al., 2020). With the exception of miR9-3, which is located upstream of a long intergenic noncoding RNA (lincRNA), all miR-9 genes are intragenic, overlapping annotations of lincRNAs or proteins (Yates et al., 2020) (Figure C.1A,B). Our in silico analysis of previously published RNA-seq data shows differential temporal expression of six of the seven miR-9 paralogues hosts (White et al., 2017). It is also clear that upregulation of miR-9 host genes (and hence miR-9) coincides with a gradual decline in the expression of Her/Hes family gene expression, consistent with the idea that Her/Hes genes are major targets of miR-9 (Figure C.1C) (Bonev et al., 2011).

Previous work has revealed that the seven miR-9 zebrafish paralogues are expressed in the forebrain at early stages of neurogenesis; however, toward the end of embryonic neuronal differentiation they are also expressed in the hindbrain (Nepal et al., 2016). Little is known about the period spanning the peak of neurogenesis, when miR-9 controls downstream targets such as the ultradian oscillator Her6. To characterize the expression in greater spatiotemporal detail, particularly over regions of the hindbrain area in which Hes/Her target genes are expressed, we investigated the expression of all seven primary transcripts over a time period spanning the peak of neurogenesis, which occurs at 33 hpf (Lyons et al., 2003), using specific probes for each pri-mir-9 (Figure C.2; Section 5.6.3).

We observed that all pri-mir-9s were first expressed in the forebrain (24 hpf) in a regional specific manner, which is not further characterised here. At 48 hpf they are all also expressed in the hindbrain (Figure 5.1C, 24 and 48 hpf; Figure C.3A-C) consistent with previously described results (Nepal et al., 2016). Differential expression was evident in the intermediate stages. Specifically, primir-9-3, -9-4 and -9-5 were expressed ahead of the others in the hindbrain (Figure 5.1C, 30-31 hpf; blue arrowhead). At the peak of hindbrain neurogenesis (34-36 hpf), pri-mir-9-2 and -9-7 were upregulated, joining most of the pri-mir-9s that were highly expressed at this stage (Figure 5.1C, 34-36 hpf). Pri-mir-9-1 and -9-6 were temporally delayed, showing hindbrain expression at 48 hpf, at which point all pri-mir-9 were fully expressed. Quantifying the expression with RT-qPCR confirmed that pri-mir-9-4 and -9-5 were expressed early and that expression of pri-mir-9-1 commenced relatively late, at 42 hpf (Figure 5.1D,E). At 48 hpf, all pri-mir-9s had a lower level of expression, although pri-mir-9-1 continued to be relatively high compared with the other pri-mir-9s (Figure 5.1D,E). Overall, every pri-mir-9 was expressed in the CNS and exhibited a temporal progression.

5.4.2 Expression of pri-mir-9s from distinct loci is additive and sequentially activated

To achieve a more detailed characterisation of expression, we selected three different primary transcripts based on: (1) the onset of their hindbrain temporal expression during development, earliest or latest; and (2) a phylogenetic analysis of sequence based on vertebrate evolutionary relationship performed by Alwin Prem Anand et al. (2018) to select representatives that are widely distributed in the phylogenetic tree. Thus, pri-mir-9-5 was selected as the earliest to be expressed in the hindbrain and belonging to clade I/subgroup I, pri-mir-9-4 as the earliest

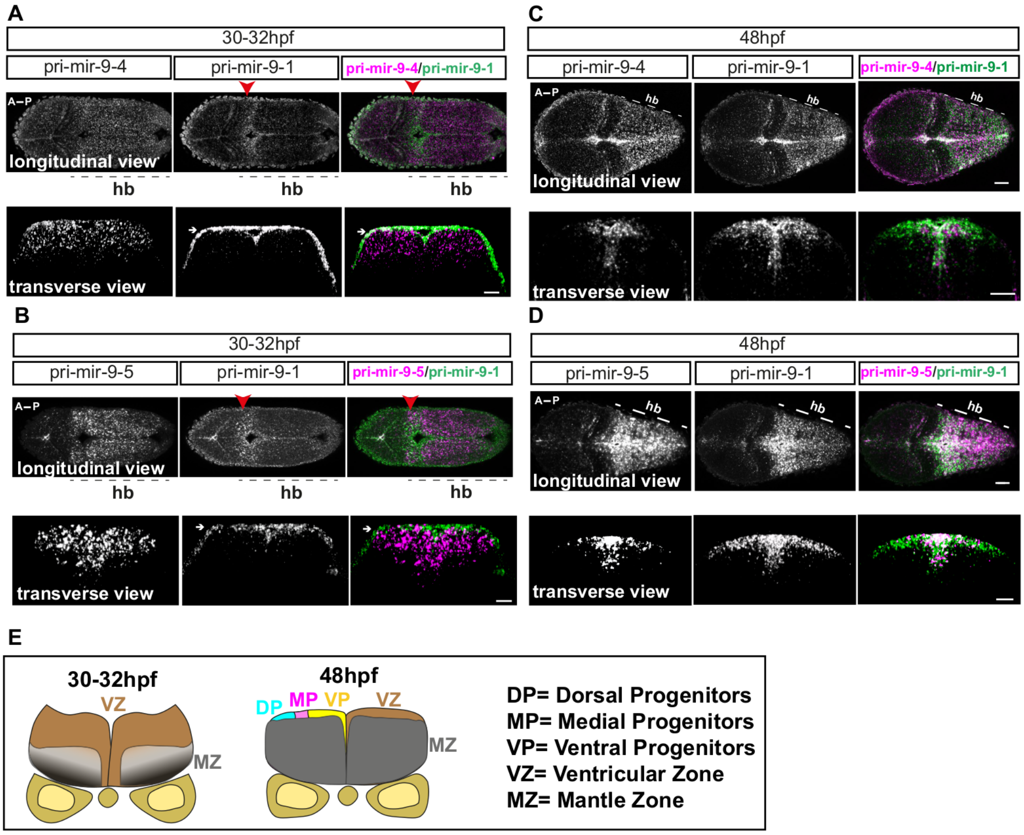


Figure 5.2: **Progressive additive expression of pri-mir-9s during development. A–D.** Representative example of double fluorescent WM-ISH (WM-FISH) labelling of pri-mir-9-1/pri-mir-9-4 (A,C) and pri-mir-9-1/pri-mir-9-5 (B,D) in hindbrain (hb) from wild-type embryo observed at 30-32 hpf (A,B) and at 48 hpf (C,D). Transverse view was collected from hindbrain rhombomere 4/5. Longitudinal view was collected from embryos with anterior to the left and posterior to the right; images are maximum intensity projection; 5 μm thickness for 48 hpf embryos and 10 μm thickness for 30-32 hpf embryos. Merged images indicate pri-mir-9-4 or -9-5 in magenta and pri-mir-9-1 in green. White arrows indicate artefactual signal originated from the amplification step with FITC staining in the WM-FISH; red arrowheads indicate rhombomere 1 of the hindbrain. Pri-mir-9-1/pri-mir-9-4: longitudinal/30-32 hpf, $N = 3$; transverse/30-32 hpf, $N = 3$; longitudinal/48 hpf, $N = 4$; transverse/48 hpf, $N = 8$. Pri-mir-9-1/pri-mir-9-5: longitudinal/30-32 hpf, $N = 3$; transverse/30-32 hpf, $N = 4$; longitudinal/48 hpf, $N = 4$; transverse/48 hpf, $N = 5$. **E.** Schematic of transverse section from zebrafish hindbrain at the level of the otic vesicle for 30-32 hpf and 48 hpf. A, anterior; MZ, mantle zone; P, posterior; VZ, ventricular zone. Within the VZ there are dorsal progenitors (DP), medial progenitors (MP) and ventral progenitors (VP). Scale bars: 30 μm.

and belonging to clade II, and pri-mir-9-1 as the latest and belonging to clade I/subgroup II (Figure C.3D) (Alwin Prem Anand et al., 2018).

Double whole-mount fluorescent *in situ* hybridization (WMFISH) for pri-mir-9-1/pri-mir-9-4 and pri-mir-9-1/pri-mir-9-5 performed on stage 30-32 hpf embryos revealed expression of pri-mir-9-4 and pri-mir-9-5 along the anterior-posterior (A-P) hindbrain axis, whereas the expression of pri-mir-9-1 at this early stage was limited to the region of the anterior hindbrain corresponding to r1 (Figure 5.2A; red arrowhead). A transverse view at mid-hindbrain (r4) reveals expression of pri-mir-9-4 and pri-mir-9-5 within the ventricular zone (VZ; Figure 5.2A,B,E), indicating that pri-mir-9s are expressed in the region where most of the progenitors are found (Lyons et al., 2003; Tambalo et al., 2020). Pri-mir-9-1 staining shows an artefactual surface expression, as indicated with white arrow in transverse view at 30-32 hpf (Figure 5.2A,B); this is because of the WM-FISH detection method.

We repeated this analysis at 48 hpf to examine whether the late expression of pri-mir-9-1 is cumulative with pri-mir-9-4 and primir-9-5 or spatially distinct. Double WM-FISH of pri-mir-9-1 with pri-mir-9-4 or pri-mir-9-5 revealed overlapping expression of the primary transcripts in both longitudinal and transverse views (Figure 5.2C,D). In addition, some distinct expression was observed in transverse views in that pri-mir-9-1 was more broadly expressed toward the dorsal progenitor region (Figure 5.2C-E) when compared with pri-mir-9-4 and pri-mir-9-5.

5.4.3 Mature miR-9 accumulates in single cells by overlapping expression of distinct loci primary transcripts

For overlapping expression to contribute to the total levels of mature miR-9 in a cell, early and late pri-mir-9s would need to be expressed in the same cells. Thus, we investigated pri-mir-9 expression at the single-cell level, using triple WM-smiFISH for pri-mir-9-1, -9-4 smiFISH (single-molecule inexpensive fluorescent *in situ* hybridization) and -9-5 to detect nascent transcription sites, and Phalloidin staining to reveal cell boundaries. At 30 hpf we observed that most cells expressed only one miR-9 primary transcript, pri-mir9-4 or pri-mir-9-5, while a small proportion expressed both and none expressed pri-mir-9-1 (Figure 5.3A,D-F). By contrast, at 36-37 hpf and 48 hpf (Figure 5.3B,C), the number of cells that expressed one pri-mir-9 decreased and, correspondingly, the number that expressed two or three pri-mir-9s increased. The most striking increase was observed in the number of cells that co-express three pri-mir-9s at 36-37 hpf, which was because of the onset of transcription of pri-mir-9-1 in the same cells that expressed pri-mir-9-4 and -9-5. This finding suggests that, in many hindbrain cells, the late expression of pri-mir-9-1 is added to the earlier expression of pri-mir-9-4 and -9-5.

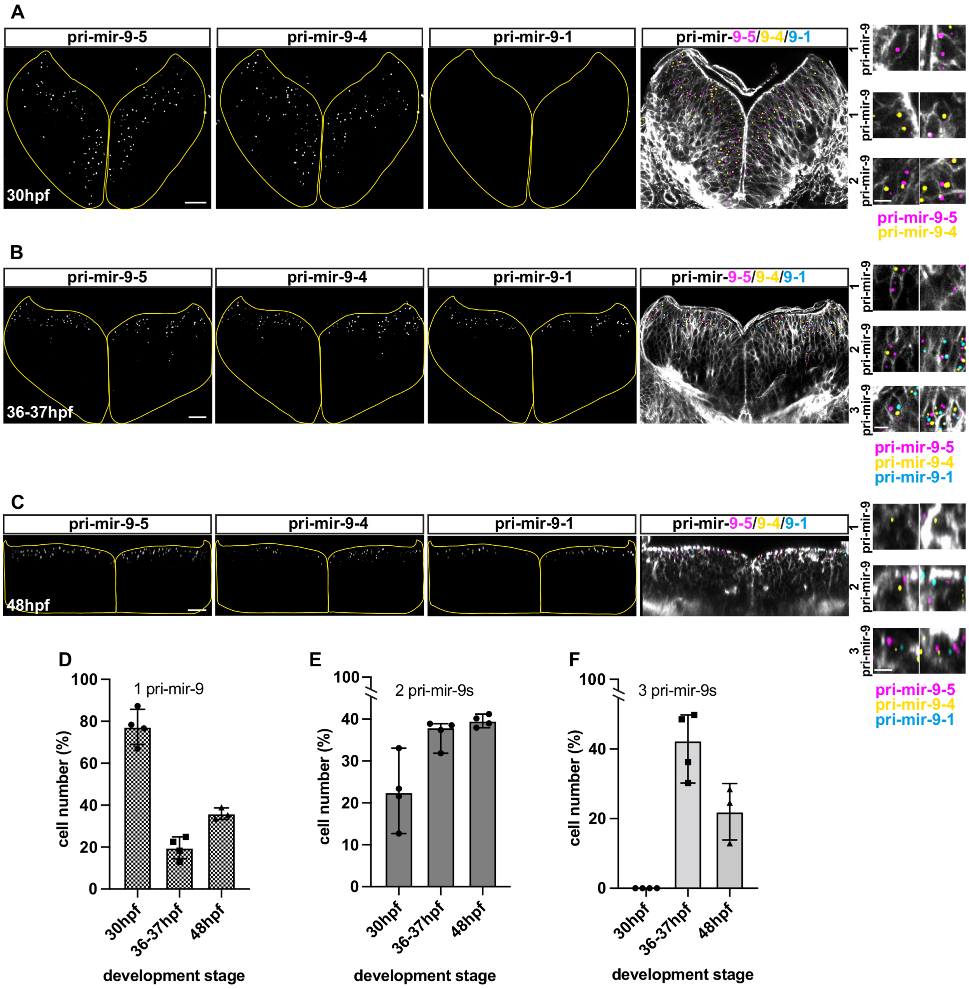


Figure 5.3: **Mature miR-9 expression in a cell is contributed by overlapping activation of distinct miR-9 loci. A–C.** Representative example of the transverse view from triple whole-mount smiFISH, labelling active transcriptional sites for pri-mir-9-5, -9-4 and -9-1 (from left to right) combined with cell boundary staining (Phalloidin-Alexa Fluor 488) in hindbrain from wild-type embryo at 30 hpf (A), 36-37 hpf (B) and 48 hpf (C). Merged images show pri-mir-9-5 in magenta, pri-mir-9-4 in yellow, pri-mir-9-1 in cyan and membrane in grey. **A'–C'.** Increased magnification of representative images to show single cells expressing any single pri-mir-9 (1 pri-mir-9), any two different pri-mir-9 (2 pri-mir-9) and the three different pri-mir-9 (3 pri-mir-9). **D–F.** Percentage of cells expressing any single pri-mir-9 (D), any two different pri-miR-9 (E) and three different pri-mir-9 (F) relative to the total number of cells positive for the precursors (30 hpf, $N = 4$; 37-37 hpf, $N = 4$; 48 hpf, $N = 3$). Data are median with 95% confidence interval. Scale bars: 20 μm (A-C); 5 μm (A'-C').

5.4.4 Medial and dorsal progenitors maintain concurrent expression of miR-9 primary transcripts at late neurogenesis

Based on the smiFISH data presented above, we created a map that depicts transcription in single cells in transverse sections of the hindbrain over development (Figure 5.4A-C). From left to right we observe: (i) the whole transverse section obtained from the Phalloidin staining, (ii) the region in which cells transcribe primir-9-5, (iii) the cells with overlapping transcription

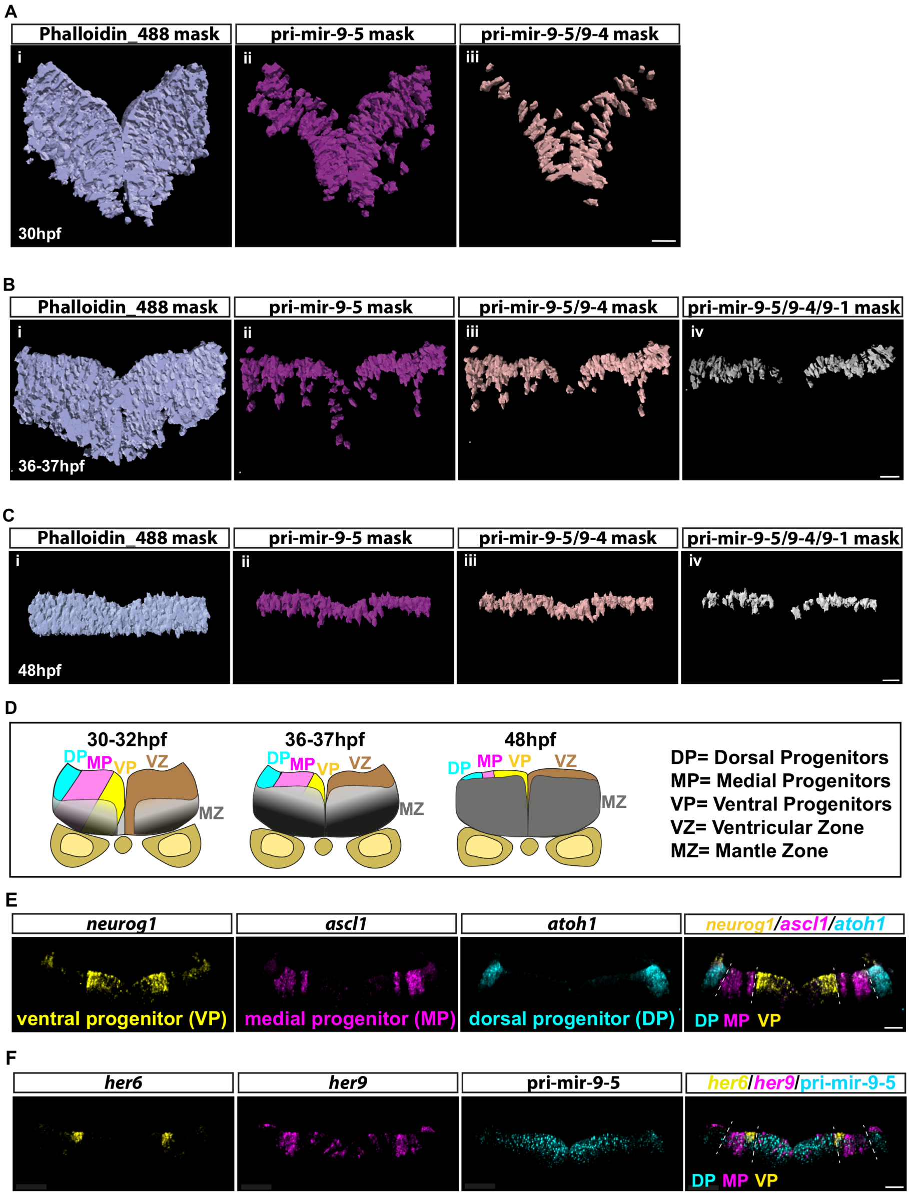


Figure 5.4: **Concurrent expression of miR-9 precursors in dorsal and medial progenitors.**

for pri-mir-9-5/ 9-4 and (iv) the cells in which the three primary transcripts are transcribed (Figure 5.4A-C). At 30 hpf, pri-mir-9-4 and -9-5 are co-expressed in many cells of the VZ (Figure 5.4A). At 36-37 hpf, pri-mir-9-1 is transcriptionally activated in most, but not all, neural progenitors that already express pri-mir-9-4 and -9-5 (Figure 5.4B). At 48 hpf the pattern of triple pri-mir-9 co-expression is similar to that seen in 36-37 hpf (Figure 5.4C). This result supports the concurrent expression of pri-mir-9s at late stages but also shows their expression in the neural progenitor area, which thins during development as cells differentiate (Figure 5.4D). All three paralogues are switched off in differentiating cells located in the marginal zone, suggesting that they are involved in the decision to differentiate rather than in maintain-

Figure 5.4: **A–C.** Mask representing segmented cells obtained from confocal images in Figure 5.3. Using Imaris software, the cell segmentation was performed based on the membrane marker, Phalloidin-AF488, and the spot tool allowed us to count active transcriptional sites for pri-mir-9-5, -9-4 and -9-1. From left to right we visualize the mask showing all segmented cells present in the transverse view of the hindbrain (i, light blue), segmented cells that express pri-mir-9-5 (ii, magenta), both pri-mir-9-5 and -9-4 (iii, light pink) and all three pri-mir-9-5, -9-4 and -9-1 (iv, grey). The study was performed at 30 hpf (A), 36-37 hpf (B) and 48 hpf (C). **D.** Schematic of the transverse section from zebrafish hindbrain at the level of the otic vesicle for 30-32 hpf, 36-37 hpf and 48 hpf. MZ, mantle zone; VZ, ventricular zone. Within the VZ there are dorsal progenitors (DP), medial progenitors (MP) and ventral progenitors (VP). **E.** Representative example of transverse view at 36-37 hpf from triple whole-mount smiFISH labelling *neurog1* as ventral progenitor marker (VP, yellow), *ascl1* as medial progenitor marker (MP, magenta) and *atoh1* as dorsal progenitor marker (DP, cyan). The merge image shows the three progenitor markers in their respective colours, which are expressed in the VZ as described in D. **F.** Representative example of transverse view at 36-37 hpf, from triple whole-mount smiFISH labelling pri-mir-9-5 (cyan) and the zebrafish Hes1 orthologues *her6* (yellow) and *her9* (magenta). The merge image shows pri-mir-9-5 (cyan) co-expressing with *her6* (yellow) and *her9* (magenta). The dashed line indicates a boundary between different progenitor regions (dorsal, medial and ventral progenitor region). Scale bars: 20 μm.

ing the differentiated state (Figure 5.4A-C).

To explore the identity of the triple pri-mir-9 expressing progenitors, we turned our attention to the dorso-ventral (D-V) progenitor axis of the VZ. The everted structure of the zebrafish hindbrain means that dorsal progenitors are located more laterally than medial or ventral ones (Figure 5.4D). We compared the expression to neurog1, ascl1 and atoh1, which are markers for ventral, medial and dorsal progenitors, respectively (Figure 5.4E) (Tambalo et al., 2020). Remarkably, at early stages of development the cells expressing two primary transcripts were mostly localized in the ventral progenitor region of the VZ (Figure 5.4A), whereas at later stages the cells with three primary transcripts excluded the ventral-most domain (Figure 5.4B,C), suggesting that miR-9 high levels are required in medial and dorsal progenitors. Pri-mir-9-5 was expressed throughout the everted D-V axis (Figure 5.4B,D,F) and was coexpressed with *her6* and *her9*, both of which were expressed in the progenitor domain (mainly medial and some dorsal) and are downregulated as cells differentiate. We have previously described Her6 protein expression also in ventral progenitors, which is, however, extremely weak at late development (36-37 hpf) and has not been detected by smiFISH here (Soto et al., 2020). Both *her6* and *her9* contain miR-9 binding sites and are candidates for dynamic regulation by miR-9 (Figure 5.4F) (Coolen et al., 2013; Leucht et al., 2008; Soto et al., 2020).

5.4.5 Knocking out the late pri-mir-1 preferentially affects neuronal differentiation from medial progenitors

The spatial analysis above showed that the expression of pri-mir-9-1 is added onto to pre-existing pri-mir-9-4 and -9-5 expression in medial and dorsal progenitors. To find out whether there is any specificity in deleting pri-mir-9-1, we designed a CRISPR/Cas9based knockdown with guides that were specific to pri-mir-9-1 (Figure 5.5A; Figure C.4A-C). This resulted in reduc-

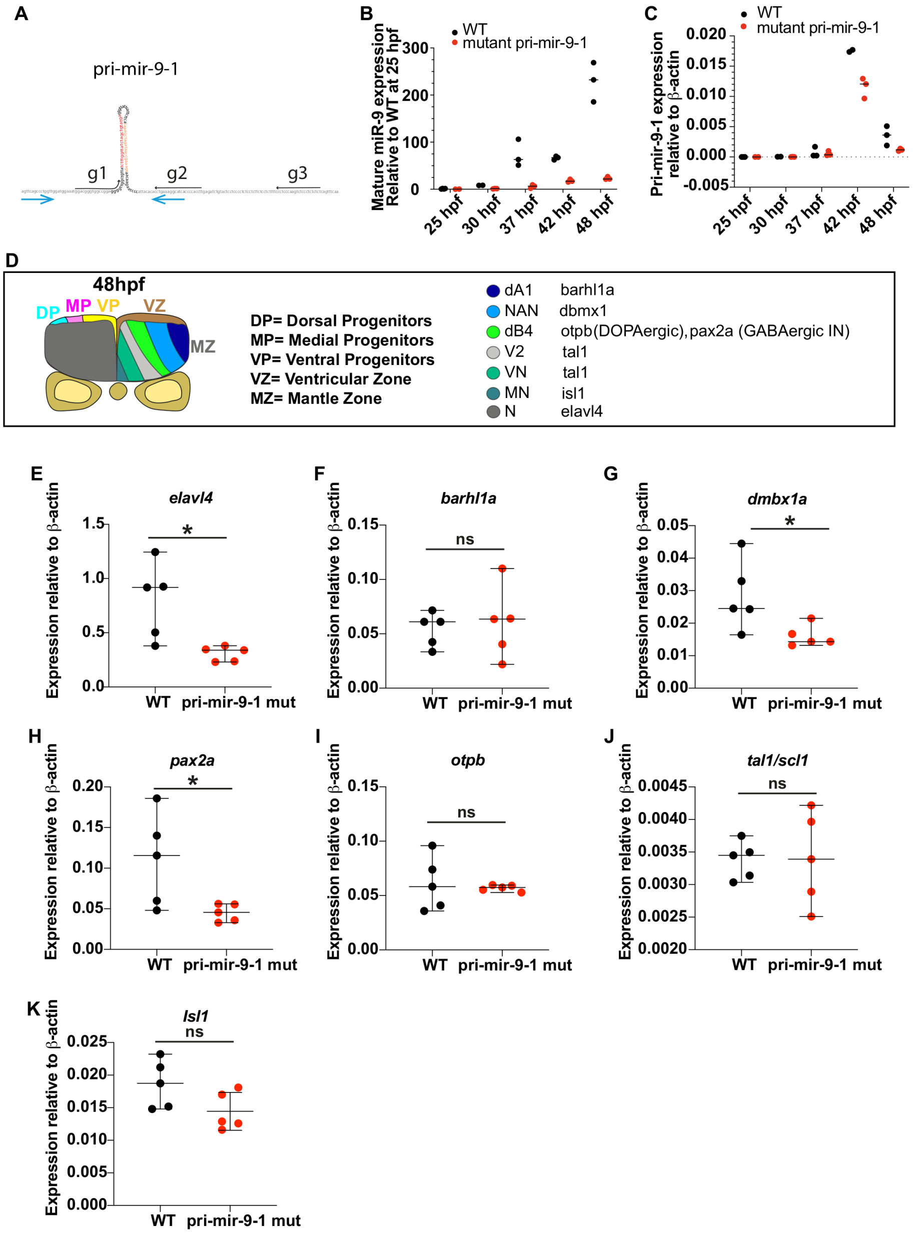


Figure 5.5: **Knocking out the late pri-mir-9-1 preferentially affects neuronal differentiation from medial progenitors.**

tion of mature miR-9 and pri-mir-9-1 from 37 hpf onwards when the endogenous locus was transcribed (Figure 5.5B,C). RT-qPCR was also performed to primir-9-3, -9-4 and -9-5 under mutation of pri-mir-9-1. Some reduction (with high variability between samples) was also observed in pri-mir-9-4 and -9-5, but it was not maintained at later stages of development (Figure C.4E,F, 48 hpf). Pri-mir-9-3 was not affected (Figure C.4D).

Exploring the potential defect further we used a panel of differentiation markers spanning the D-V axis (Figure 5.5D). Injected fish did not show overt abnormalities; however, RT-qPCR analysis at 3 days post-fertilisation (dpf) showed that the differentiation marker elavl4 was

Figure 5.5: **A.** Pri-mir-9-1 hairpin loops with the respective primers used for quantitative PCR annotated as blue arrows (Section 5.6.2; Table C.3). Customized guide RNA to delete specifically pri-mir-9-1 are annotated as g1, g2 and g3. Red sequence, miR-9-5' arm; orange sequence, miR-9-3' arm; black letters, premir-9; grey sequence, partial sequence of pri-mir-9-1. **B.** Taqman RT-qPCR of mature miR-9 from dissected hindbrain at different stages of development, in wild-type conditions (black dots) and deletion of pri-mir-9-1 (red dots), relative to wild-type at 25 hpf. **C.** SYBR green RT-qPCR relative quantification of primir-9-1 from dissected hindbrain at different stages of development, in wild-type conditions (black dots) and deletion of pri-mir-9-1 (red dots). Quantification was normalised using β-actin. **D.** Schematic of transverse section from zebrafish hindbrain at the level of the otic vesicle at 48 hpf. MZ, mantle zone; VZ, ventricular zone. Within the VZ there are dorsal progenitors (DP), medial progenitors (MP) and ventral progenitors (VP). The schematic shows late neuronal markers expressed in different neuronal cell types in the hindbrain: dA1, dorsal neurons expressing *barhl1a*; NAN, noradrenergic neurons expressing *dbmx1*; dB4, GABAergic interneurons expressing *pax2a*; V2, interneurons expressing *tal1*; VN, ventral neurons expressing *tal1*; MN, motor neurons expressing *isl1*; N, pan neuronal zone expressing *elavl4*; *otpb* is localised in the dB4 region but is a marker for dopaminergic neurons. **E–K.** SYBR green relative quantification of *elavl4* (E), *barhl1a* (F), *dmbx1a* (G), *pax2a* (H), *otpb* (I), *tal1/scl1* (J) and *isl1* (K) from dissected hindbrain at 72 hpf, in wild-type conditions (black dots) and deletion of pri-mir-9-1 (red dots). Quantification was normalised using β-actin. Horizontal bars indicate median with 95% confidence intervals. *P<0.05 (Mann-Whitney two-tailed test). (B-C) $N = 3$, each N contain a pool of 10 hindbrain. (E-K) $N = 5$. ns, not significant.

reduced (Figure 5.5E). This analysis also showed that, within the *her6* domain, there was a reduction of noradrenergic neurons (NAN) derived from medial progenitors (*dmbx1a*; Figure 5.5G) and adjacent GABAergic interneurons (*pax2a*; Figure 5.5H), while the more ventral neuronal markers *tal1* and *isl1* (*isl1a*) were not significantly different, neither was the most dorsal marker (*barhl1a*) outside the *her6* domain (Figure 5.5F,I-K).

Medial/dorsal progenitors differentiate later in vertebrate development than ventral ones (Delile et al., 2019), therefore our findings suggest that the late increase of miR-9, afforded by the additional deployment of pri-mir-9-1, is needed for cells to adopt a late neuronal fate.

5.4.6 A miR-9 stepwise increase may be required to overcome adaptation of downstream target expression

Having shown that the increase in miR-9 in development is functionally important for differentiation, we wanted to explore whether the shape of the increase is also important. In other words, whether the way that miR-9 increases in steps can be decoded. This was motivated by the biological evidence obtained from smiFISH and RT-qPCR experiments in which we observed a stepwise sharp increase of the primary transcripts (Figure 5.6A) and the mature miR-9 (Figure 5.1B) over time. We used a mathematical model to ask whether a simple network of gene interactions can differentially respond to a stepwise increase of miR-9 rather than a gradual increase.

Biological systems need to be robust to stochastic fluctuations that are due to low copy numbers, or random perturbations in the surrounding environment. This is referred to as adapta-

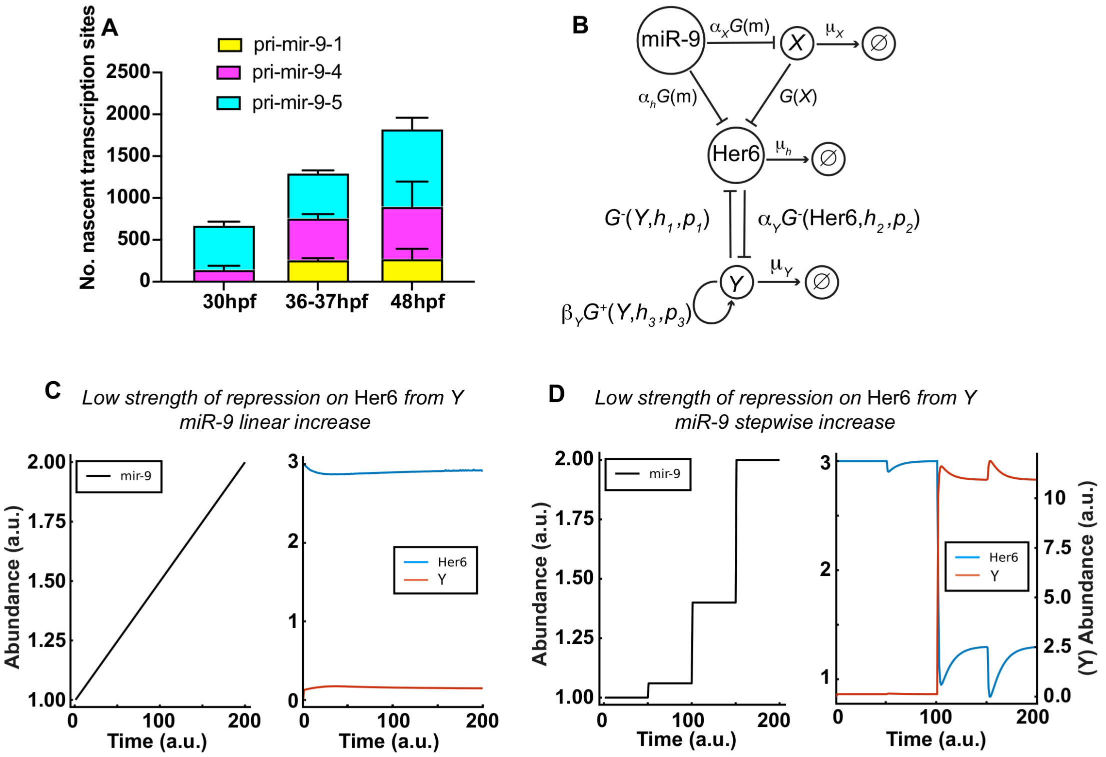


Figure 5.6: **A miR-9 stepwise increase may be required to overcome adaptation of downstream target expression. A.** Graph representing the number of nascent transcription sites for pri-mir-9-1 (yellow), pri-mir-9-4 (magenta) and pri-mir-9-5 (cyan) at 30 hpf, 37-38 hpf and 48 hpf in 25 μm thick transverse sections. 30 hpf, $N = 4$; 37-37 hpf, $N = 4$; 48 hpf, $N = 3$. Data are mean±s.d. **B.** Schematic of the extended mathematical model, which combines an incoherent feedforward loop with an additional mutually repressive self-activating downstream target, Y. The parameters α_h, α_X and α_Y represent the basal production rates of h, X and Y, respectively. μ_h, μ_X and μ_Y represent the degradation rates of h, X and Y, respectively, and β_Y represents the production rate of Y under self-activation. The h_i and p_i are Hill coefficients and repression thresholds, respectively, for each of the Hill functions G^+ and G^-, and $G(x) = 1/x$. The model is described in detail in the Section 5.6.13 (subsection Extended model) and specific parameter values are listed in Table C.14. **C,D.** Dynamics of Her6 in response to different miR-9 expression profiles, for the extended model. (C) A linear miR-9 expression profile leads to a small initial response in Her6 expression levels, which returns to steady state levels owing to the perfect adaptation. (D) Large instantaneous changes in miR-9 can result in a change in steady state for Her6. The initial step change is not sufficient to cause a change in steady state, therefore we introduce a fold change in the stepwise increase of miR-9, which activates Y and represses Her6 into a lower steady state.

tion in the context of a particular output of interest, making the biological system resistant to changes in the input. However, some changes in signals are not simply due to noise or environmental fluctuations, and adaptive systems may therefore also have to respond to specific signals under changing conditions, especially during development, in order to move into a new state. Thus, we explored whether a stepwise change in gene expression can allow a system to move out of an adaptively stable state. As incoherent feed-forward loops (IFFL) are common in biology (Goentoro et al., 2009; Shen-Orr et al., 2002) and have been shown to enable adaptation (Khammash, 2021), we hypothesized the existence of such a network centred around miR-9 as the input and Her6 as the output (Figure C.5A; Section 5.6.13) in which miR-9 affects Her6 negatively (directly) but also positively (indirectly) via repressing a repressor, X. Here, miR-9 directly reduces the rate of production of Her6 protein as well as the rate of production of an intermediate (unknown) species X. Similarly, the production of Her6 is repressed by X (Figure C.5A; parameter values, Table C.14). Mathematically, we say that Her6 adapts perfectly to changes in miR-9 as, in this model, the steady state of Her6 is independent of miR-9 (Section 5.6.13, Steady state calculation of Her6 in the perfect adaptation model). The speed of this adaptation is controlled by the difference in reaction speed of the direct and indirect interactions between miR-9 and Her6. If the direct interaction is much faster than the indirect interaction, Her6 returns to steady state slowly after miR-9 copy numbers are perturbed. However, if the indirect interaction is faster, adaptation occurs quickly.

Such ‘perfect adaptation’ is beneficial because it allows stable mean expression of Her6 in the presence of fluctuations of miR-9 expression (Figure C.5B,C). Conversely, no changes in miR-9, i.e. linear or stepwise, can lead to persistent downregulation of Her6. As Her6 is downregulated in response to increasing miR-9 levels during development, there would need to be an additional mechanism that enables the controlled escape from perfect adaptation. To investigate this, we extended our model to include such a potential mechanism (Figure 5.6B). Specifically, we introduced a downstream target of Her6, named Y, which self-activates and interacts with Her6 through mutual repression (as we have already previously hypothesised in Soto et al. (2020)). The different behaviours of this system can be seen in Figure 5.6C,D. A linear increase in miR-9 leads to an initial repression of Her6, which then proceeds to return to its unperturbed steady state, due to the perfect adaptation (Figure 5.6C; Figure C.6A). However, following a sharp increase of miR-9, the concentration of Her6 decreases more strongly. This is sufficient for Y to overcome the repression from Her6, so that it can self activate and in turn repress Her6 into a new, lower steady state (Figure 5.6D; Figure C.6B). Hence, this extended motif can indeed overcome the built-in adaptation. Importantly, the escape from adaptation is triggered by a step-like change in miR-9 expression and cannot be achieved through gradual changes in miR-9 expression, or small-scale fluctuations.

The qualitative behaviour of the model is not sensitive to different values of the parameter p_1, which regulates the strength of repression of Her6 by Y. The precise choice of p_1 simply modulates the level of the lower state of Her6 expression (Figure 5.6D versus Figure C.6B).

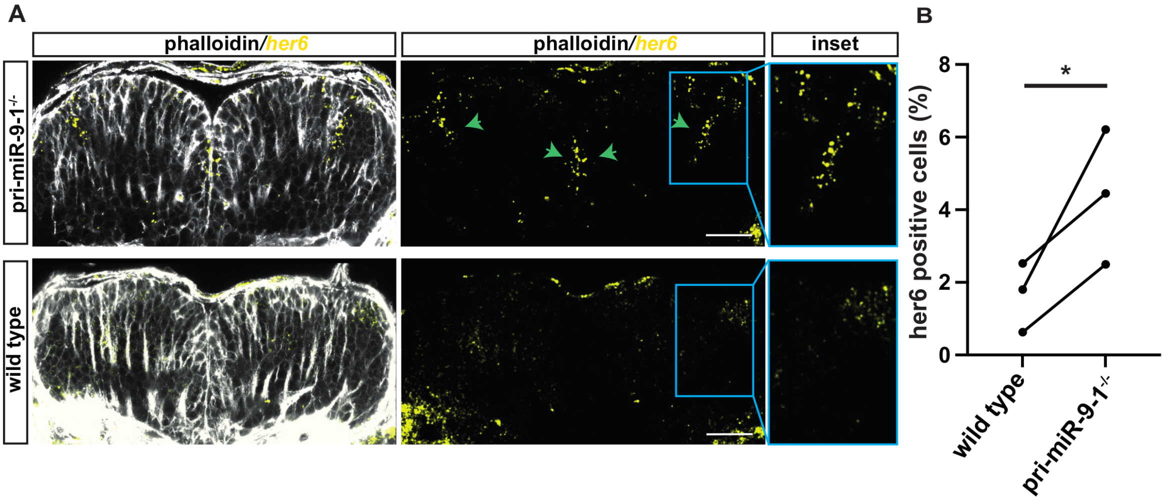


Figure 5.7: **Knocking out the late pri-mir-9-1 impairs *her6* downregulation over the course of development.** **A.** Representative example of transverse view from whole-mount smiFISH labelling her6 transcript (yellow) combined with cell boundary staining, Phalloidin-Alexa Fluor 488 (grey), in hindbrain from pri-mir-9-1 homozygote mutant (pri-mir-9-1$^{-/-}$) (bottom panels) and wild-type (top panels) embryos at 48 hpf. Insets are increased magnification from representative images from the boxed area. The images are maximum projections of three z-stacks, 1.89 mm. Green arrows indicate regions with high her6 expression levels in pri-mir-9-1$^{-/-}$ mutants. **B.** Percentage of cells expressing *her6* relative to total number of cells. Pairwise comparison of *her6*-positive cells; dots indicate mean per experiment from wild-type (two embryos, three embryos, four embryos; three independent experiments) and pri-mir-9-1 homozygote mutant (two embryos, two embryos, three embryos; three independent experiments); *P=0.041 (one-tailed paired t-test). Scale bars: 30 μm.

5.4.7 Knocking out the late pri-mir-1 results in failure to downregulate Her6 in late developmental stages

The prediction from the mathematical model is that a stepwise increase of miR-9 is needed for Her6 protein to transition to a new gene expression state. We have previously shown that, during development, Her6 expression undergoes a transition from noisy to oscillatory to downregulation (Soto et al., 2020). Therefore, we performed her6 smiFISH in transverse sections of the hindbrain to quantify the percentage of *her6*-positive cells in a wild-type and in a pri-mir-9-1 homozygous mutant (pri-mir-9-1$^{-/-}$) stable fish line (Figure C.7A-C). In the pri-mir-9-1 mutants, at 48 hpf *her6* was not downregulated in medial progenitors (Figure 5.7A,B) nor in ventral progenitors where the expression is normally lower (Soto et al., 2020). The lack of *her6* downregulation was confirmed with WM-ISH (Figure C.7D) and with live imaging of homozygous Her6::Venus knock-in zebrafish (Soto et al., 2020), which were also heterozygous for the pri-mir-9-1 mutation (Figure C.7E, pri-mir-9-1$^{+/-}$). Therefore, our findings suggest that the late activation of pri-mir-9-1 contributes to the increase of miR-9 needed to downregulate Her6/Her9 in late neural progenitors so that they can give rise to a spatiotemporally appropriate neuronal fate.

5.5 Discussion

miR-9 is expressed from several genomic loci which, after transcription and processing, produce the same 5' mature form of miR-9 that targets the key neural progenitor transcription factors, Her/Hes. How common is this multi-locus organisation? In humans, only 6.3% of mature miR arms are identical across two or more loci (Kozomara et al., 2019): it is thus not very common, but it is not unique to miR-9. In zebrafish this number rises to around 32.3% (Kozomara et al., 2019). The higher number of miR expressed from multiple loci is possibly due to the teleost-specific WGD. Evidence from rainbow trout also shows that, following the salmonid-specific extra round of WGD, miRs appear to be retained at higher levels than protein-coding genes (Berthelot et al., 2014). This may suggest that extra copies of miR are evolutionarily advantageous. Here, we propose that retention of multiple miR loci could have specific functional advantages for regulatory control of the target gene expression of an organism. By examining in detail the temporal and spatial expression at a single cell level of three selected early and late pri-mir-9s, from across their phylogenetic tree, we offer two possible, not-mutually exclusive, explanations for this multi-site organisation or primary transcripts.

The first explanation involves a qualitative mechanism. In this scenario, distinct pri-mir-9s have a different spatial expression, which allows them to target different, i.e., region-specific, gene expression. Some differences in the spatial expression of pri-mir-9s are easily discernible at low resolution (e.g. differential expression in the forebrain), whereas others are subtle and require posthybridisation sectioning to document, as we have done here. An example of the latter is the expression of pri-mir-9-1 which extends more dorsally in the hindbrain than pri-mir-9-4 at a late stage of development. This correlates well with the expression of Her6 and Her9, which are both miR-9 targets but are expressed adjacent to each other along the D-V axis (Soto et al., 2020).

The second explanation favours a quantitative mechanism. In this scenario, the differential temporal expression, where some primary transcripts commence their expression early while others are only expressed late, results in the simultaneous expression of both (or more) transcriptional loci in the same cells at a particular time in development. In support of this scenario, we have shown using smiFISH that pri-mir-9-1, a late onset primary transcript, is coexpressed in the same cells as the earlier onset pri-mir-9-4 or -9-5. This co-expression may be a strategy to increase the amount of miR-9 available to the cell to a level more than that possible with transcription from one locus alone.

Why would an increase in mature miR-9 over time be needed? One possibility is raised by the recent work from Amin et al. (2021), who demonstrated that miR-dependent phenotypes emerge at particular dose ranges because of hidden regulatory inflection points of their underlying gene networks. This indicates that the miR cellular dose is a major determinant of in vivo neuronal mRNA target selection. A complementary scenario is supported by our previous work where we have shown that the dynamical profile of Hes1 (i.e. oscillatory expression to stable expression of different levels), as well as the amount of time that Hes1 oscillates for, depends on the amount of miR-9 in the cell (Bonev et al., 2012; Goodfellow et al., 2014; Phillips et al., 2016). More recently, we have also shown, using in vivo manipulations, that the input of miR-9 changes the dynamic expression of Her6 from noisy to oscillatory and then to decreasing (Soto et al., 2020).

Taken together, these findings suggest that variations in the dose level of a single miR achieved by additive transcription can exert regulatory effects either by targeting different downstream gene products or by modifying the dynamic expression of the same targets. Both scenarios are compatible with experimental results, whereby mutating the late-onset pri-mir-9-1 preferentially reduced the appearance of markers for neurons that differentiate late. This suggests that the late miR-9 increase is important for late cell fate choices. This is further supported by our previous work reporting a complete repression of neurogenesis along the D-V axis of the hindbrain (Bonev et al., 2011) when total miR-9 is knocked out, whereas in this article pri-mir-9-1 knockout (KO) has more specific effect.

In other cases where multiple paralogues of a miR have been described, differential and non-mutually exclusive qualitative and quantitative regulation may also take place. For example, a recent study found that miR-196 paralogues show both unique and overlapping expression in the mouse (Wong et al., 2015). In this study, single KOs showed some unique phenotypes (qualitative mechanisms) but combinatorial KOs showed better penetrance and additional defects, suggesting an additive role of miR-196 paralogues in establishing vertebral number (quantitative mechanism).

A salient finding from our analysis is that the increase in the amount of miR-9 present in the cell is sharp, as one would perhaps expect by the onset of transcription from additional loci. An exciting possibility, supported by our mathematical modelling, is the existence of gene network motifs that do not respond to slow increases of miR-9 because they are designed to show adaptation, that is, to have steady output in spite of external perturbations. Such network motifs often involve IFFLs, which in turn are very common in biological systems because of their multiple advantages, including fold-change detection and robustness of output (Goentoro et al., 2009; Khammash, 2021). However, in development, cells also need to transition from one state to another in order to diversify cell fates, which is essential for the development of multicellular organisms. Thus, despite the usefulness of adaptation for robustness and homeostasis (Khammash, 2021), a mechanism must exist to be able to over-ride it. We suggest that, in the case of miR-9, a sharp, nonlinear increase may be needed to push a dynamical system into a new state and this may be associated with a cell fate change. In our case, we suggest that the increase of miR-9 during development serves to drive the dynamics of Her6 (and other targets) from one state to another, which may include temporal downregulation, and which in turn is important for the sequential acquisition of cell fates.

At present, our computational model is qualitative, rather than quantitative, and the identities of some interacting genes in the network motif are not known. For example, we postulate the existence of a gene X that lies between miR-9 and Her6. Interestingly, a preliminary bioinformatic screen using transcription factor binding profiles from JASPAR (Castro-Mondragon et al., 2022) and miR target predictions for miR-9 from TargetScanFish (Release 8.0) (McGeary et al., 2019) has identified Onecut, among others, as a potential candidate for factor X, which is a predicted regulator of Her6 and a direct target of miR-9. This is encouraging because Onecut is expressed in the zebrafish hindbrain, is a validated miR-9 target (Bonev et al., 2011; Madelaine et al., 2017) and a temporal factor for mammalian neurogenesis (Sagner et al., 2021).

Despite these limitations, this model was conceptually useful to illustrate the existence of a system that can decode and distinguish between specific upstream signalling profiles. Interestingly, miRs are very commonly involved in transcription factor network motifs, including IFFLs (Tsang et al., 2007). However, the regulation of each pri-mir-9 is presently unknown, but miRs are often involved in reciprocal interactions with transcription factors (Minchington et al., 2020). A fully parameterized model based on experimental evidence and identification of the unknown components/genes would be needed before it can be tested further.

In conclusion, by providing evidence for both a quantitative and qualitative mechanism, we have shed light on the possible roles of organising pri-mir-9s in several distinct genomic loci, which may have led to their evolutionary conservation. An added benefit of our work is that the detailed characterisation we have described here will enable the selection of the correct genomic locus for genetic manipulation of miR-9 production, depending on the precise spatiotemporal expression. It would be interesting to see whether the same mechanism is observed in mammalian species that have three distinct primary miR-9s.

5.6 Materials and Methods

5.6.1 Research animals

Animal experiments were performed under UK Home Office project licences (PFDA14F2D) within the conditions of the Animal (Scientific Procedures) Act 1986. Animals were only handled by personal licence holders.

5.6.2 mRNA extraction and RT-qPCR

miRs and total mRNA were extracted from a pool of ten zebrafish hindbrains using the miR-Vana miRNA Isolation kit and gDNA removed using DNase1 (New England Biolabs). Reverse transcription was performed with either TaqMan MicroRNA Reverse Transcription kit (Applied Biosystems) for mature miR-9 or SuperScript III (Invitrogen) with random hexamers for primirs. Each qPCR reaction was prepared in triplicate in a 96-well plate with the relevant TaqMan MicroRNA assay or using POWER SYBR Green Mastermix (Thermo Fisher Scientific), 0.2 μM each forward and reverse primer (see Table C.3 for respective primers) and 50 ng cDNA. Reactions were run on Step One Plus Real-time PCR System (Applied Biosystems) alongside negative controls. The data for each sample were normalized to the expression level of U6 snRNA for mature miR-9 or b-actin for pri-mir-9s and analysed by the $2^{-\Delta\Delta \mathrm{Ct}}$ method. For each primer pair, the PCR product was examined by gel electrophoresis and its melting curve to ensure a single fragment of the predicted molecular weight.

5.6.3 Molecular cloning

RNA probes for pri-mir-9-1, pri-mir-9-2, pri-mir-9-4, pri-mir-9-5 and pri-mir9-7 were PCR amplified and cloned into pCRII vector using primers described in Table C.1. Except for pri-mir-9-2 probe, they were designed to distinguish the primary transcripts by including sequences, intron and exon, before and after each miR processing, while also covering the sequence corresponding to mature miR-9 (Figure C.2). As the mature miR-9 sequence is conserved between paralogues, to avoid any cross-binding of probes to this sequence we mutated it on each probe using QuikChange II XL Site-Directed Mutagenesis assay. This allowed us to introduce deletions and single nucleotide exchange in specific regions of the mature miR-9 sequence (Table C.2; Figure C.2, sequence highlighted in red). pri-mir-9-3 and pri-mir-9-6 probes were generated from plasmids kindly gifted by Laure Bally-Cuif (Nepal et al., 2016).

5.6.4 Whole-mount chromogenic and fluorescence *in situ* hybridization and sectioning

Chromogenic *in situ* hybridisation was performed as previously described (Thisse & Thisse, 2008) using specific probes for each pri-mir-9 (described in Section 5.6.3) and *her6* (previously used in Soto et al. (2020)). The antibody used to detect the riboprobes was AP-anti-DIG (Roche, 11093274910, 1:1000). Multicolour fluorescence *in situ* hybridisation was modified from Lea et al. (2012) by developing with tyramide amplification (Perkin Elmer) after addition of antisense RNA probes and antibodies conjugated to horseradish peroxidase, anti-DIG-POD (Roche, 11207733910, 1:1000) and anti-FITC-POD (Roche, 11426346910, 1:500) (Lea et al., 2012).

Transverse sections were obtained as described in Dubaissi et al. (2012) with modifications. Embryos were embedded in 25% fish gelatine and 30% sucrose for a minimum of 24 h. We collected 18μm thickness hindbrain sections and transferred them onto superfrost glass slides. The slides were air dried overnight under the fume hood and mounted with Prolong Diamond Antifade.

5.6.5 Imaging

Chromogenic *in situs* were imaged using a Leica M165FC with a DFC7000T camera. Fluorescent *in situ* sections were imaged using Leica TCS SP5 upright confocal with HCX PL APO LU-V-I 20×0.5 water UV lens or Olympus FLUOVIEW FV1000 confocal with UPLSAPO 20× NA:0.75 lens.

5.6.6 smiFISH probe design and synthesis

The smiFISH probes were designed using the probe design tool at http:// www.biosearchtech.com/stellarisdesigner/. The software can assign a varied size of probes, 18–22 nt, therefore we gave a size of 20 nt for all designed probes with the maximum masking level available for zebrafish. Using the respective pri-mir-9 sequence we designed 36 probes for pri-mir-9-1, 35 probes for pri-mir-9-4 and 35 probes for pri-mir-9-5 (Tables C.5 to C.7, respectively). Using the respective gene mature mRNA sequence, we designed 29 probes for *her6*, 33 probes for *her9*, 40 probes for *neurog1*, 39 probes for *atoh1a* and 40 probes for *ascl1a* (Tables C.8 to C.12, respectively). The designed probes were X-FLAP tagged (5'-CCTCCTAAGTTTCGAGCTGGACTCAGTG-3') at the 5' of each gene-specific sequence. The gene-specific probes were ordered from Integrated DNA Technologies (IDT) in a 96-well format in nuclease-free water, 100 μM concentration. Upon arrival, we combined 100 μl of the gene-specific probes together, mixed, split into 100 μl aliquots and stored at -20°C. In addition, we ordered fluo-FLAP sequences (5'-CACTGAGTCCAGCTCGAAACTTAGGAGG-3') from either IDT or Biosearch Technology. These were labelled with either Atto-550, CalFluor-610 or AlexaFluor-647. Each gene-specific probe mix was labelled by mixing 2 μl of the gene-specific X-FLAP probe mix (100 μM), 2.5 μl of fluo-FLAP (100 μM) and 5 μl of 10× NEBuffer 3 (New England Biolabs) in a final volume of 50 μl. The hybridisation cycle was 85°C for 3 min, 65°C for 3 min and 25°C for 3 min. The labelled probe was stored at -20°C.

5.6.7 Whole mount smiFISH

The whole-mount smiFISH protocol for zebrafish embryos was developed by adapting smiFISH protocol from Marra et al. (2019). Embryos were fixed in 4% formaldehyde in 1× PBS. After the smiFISH protocol, embryos were stained with Phalloidin-Alexa Fluor 488 (400× dilution in PBS 1× Tween 0.1%) for 1h at room temperature and followed by three washes with PBSTween. Embryos were embedded in 4% low melting point agarose (SigmaAldrich) to collect 250 μm thickness hindbrain transverse sections.

5.6.8 smiFISH microscopy and deconvolution

smiFISH images were collected using a Leica TCS SP8 upright confocal with HC APO L U-V-I 63×/0.9 water lens, magnification 0.75×. We acquired three-dimensional stacks of 1024×1024 pixels and z-size 0.63 μm, magnification 0.75×, 16 bits per pixel, pinhole of 1 airy unit and scan speed of 200. Channels were sequentially imaged. smiFISH images were collected with frame accuracy 3 and line average 6.

To quantify *her6*-positive cells from smiFISH images we acquired three-dimensional stacks of 1024×1024 pixels and z-stacks 43-51, covering a total of 27-32 μm, that is approximately the size of half to one rhombomere (voxel size x:0.229, y:0.229, z:0.63 μm).

Deconvolution of confocal images was performed using Huygens Professional Software. As pre-processing steps, the images were adjusted for ‘microscopic parameters’ and ‘object stabilizer’ as additional restoration, the latter was used to adjust for any drift during imaging. Following this, we used the deconvolution Wizard tool, the two main factors to adjust during deconvolution were the background values and the signal-to-noise ratio. Background was manually measured for every image and channel, and the optimal signal-to-noise ratio identified for the images was value 3. After deconvolution the images were analysed using Imaris 9.5.

5.6.9 smiFISH segmentation

Segmentation was performed using Phalloidin-AlexaFluor 488 as a membrane marker. Using Imaris 9.5 software we selected the ‘Cells tool’ from which ‘Cells only’ was used as detection type and ‘Cell boundary’ was selected as cell detection type. Automated segmentation was performed, followed by manual curation to identify any incorrectly segmented cells.

To quantify pri-mir-9 nascent transcriptional sites and her6 transcripts we used the ‘Spot tool’. The estimated spot diameter size was xy 1 μm and z 2 μm. We used the default parameters to identify the nascent transcriptional sites and further manual curation was performed to correct for minimal errors carried out by the software. Further on, spots were imported into the segmented cells to identify the cells that contained one, two or three pri-mir-9s.

Following membrane segmentation and quantification of her6 transcripts, the percentage of her6-positive cells was calculated over the total number of cells segmented from the hindbrain transverse section (covering 27-32 μm).

5.6.10 Expression analysis of Hes/Her genes and miR hosts

For the *in silico* analysis of the miR host gene expression we downloaded the time course RNA-seq data (TPM) from White et al. (2017). Here, we used the overlapping host genes as a proxy for the expression of the miR. miR would not show up in standard RNA-seq analysis and there is no current miR time course data. Host genes were identified as those with overlapping annotations with the miR-9 genes. The host genes for each pri-mir-9 are in Table C.4. Pri-mir-9-7 has no overlapping annotation at this time and is thus not reported on in these data.

We filtered the RNA-seq data removing genes which were neither the host genes of the miR or members of the Her family. Three repeats for each stage of development are included in the data and we averaged the expression across the three repeats for each stage. The stages reported in the data are based on standard embryonic stages in zebrafish development. However, we wanted to visualize the expression in terms of hours and the stages were converted accordingly. Finally, before plotting, these data were z-scored to normalize the expression of each of the genes so that we could compare changes in expression over time rather than absolute levels. These data were then plotted using the heatmap.3 package in R.

5.6.11 Deletion of pre-mir-9-1 using CRISPR/Cas9

Preparation of Cas9nls and sgRNAs

For pre-mir-9-1 deletion using CRISPR/Cas9, sgRNA target sites were identified using the CRISPRdirect (http://crispr.dbcls.jp/) and Target Finder (Feng Zhang lab; http://crispr.mit.edu/). sgRNAs were generated following CRISPRscan protocol (Moreno-Mateos et al., 2015) using the oligonucleotides described in Table C.13. Transcription of sgRNA was carried out using MEGAshortscript T7 kit (Ambion/Invitrogen) with 100-400 ng of purified DNA following the manufacturer's instructions. After transcription sgRNA was purified using MEGAclear™ Transcription Clean-Up Kit. The Cas9nls protein was obtained from New England Biolabs (M0646T).

Microinjection and genotyping

One-cell stage wild-type embryos were injected with $\sim$ 1 nl of a solution containing 185 ng/μl Cas9nls protein, 125 ng/μl sgRNA, 40 ng/μl caax-mRFP mRNA in 0.05% Phenol Red. To evaluate if each sgRNA was generating mutation, genomic DNA was extracted from 3-4 dpf embryos using 50 μl NP lysis buffer per embryo (10 mM Tris (pH 8), 1 mM EDTA, 80 mM KCl, 0.3% NP40 and 0.3% Tween) and 0.5 μg/μl Proteinase K (Roche) for 3-4 h at 55°C, 15 min at 95°C and then stored at 4°C. Then, High Resolution Melt (HRM) was performed (Figure C.4A) using the Melt Doc kit (Applied Biosystems) following the manufacturer's instructions. Specific primers were designed to generate an amplicon of 395 bp in wild-type conditions: forward primer 5'-ACAGTTGACTTTCTAATTACAACCC-3' and reverse primer 5'-AGCAGGAGGAGATAATCACAGC-3'.

To analyse the effect of pre-mir-9 deletion in F0 embryos we combined three different sgRNA flanking the region of the mature miR-9 region and they were microinjected as described above. We chose to use three sgRNAs to increase our probability of deleting the mature miR-9 sequence. The embryos were injected with 125 ng/μl of each sgRNA: this low concentration of sgRNA was used to not have overt phenotype at the macroscopic level during the experimental period (24hpf-72 hpf), minimizing the chances of non-specific toxicity. Further on, the amplicons with deletion were identified by agarose gel and sequencing (Figure C.4B,C), as described below.

To identify F1 progeny with germ line transmission (GLT), 3-5 dpf embryos were fin clipped following the protocol described by Robert Wilkinson (R. N. Wilkinson et al., 2013) with modifications. Sylgard (SigmaAldrich, 761028)-coated 10 cm dishes were prepared for dissections. Embryos were placed into Sylgard-coated dishes containing L15 medium with 0.1% Tricaine (Sigma-Aldrich, UK) and 5% fetal bovine serum (Sigma-Aldrich). Once fin clipped, the embryo was rinsed in E3 medium and transferred to a fresh well; the biopsy was transferred to a PCR tube for genomic extraction. Genomic extraction was carried out in 10 μl volume using Phire Animal Tissue Direct PCR kit (Thermo Fisher Scientific, F-140WH). PCR reaction was carried out with 1 μl of the genomic extraction and primers used for HRM. An amplicon of 395 bp indicates a wild-type band and 275 bp indicates a pri-mir-9-1 mutant band. To evaluate the region deleted in the F1 pri-mir-9-1 mutants, PCR was performed per embryo, and the amplicon obtained was cloned into pCRII and transformed into bacteria Top10. Three bacterial colonies were miniprepped and sequenced.

5.6.12 Live imaging of whole developing hindbrain

The F1 adult animals were kept as Her6::Venus$^{+/-}$;pri-mir-9-1$^{+/-}$ and were inbred to obtain and compare offspring such as Her6::Venus$^{+/+}$;pri-mir-91$^{+/-}$ or $^{-/-}$ with Her6::Venus$^{+/+}$;pri-mir-9-1$^{+/+}$. To perform a comparative analysis of overall Her6 expression during hindbrain development on the mixed genotype population, a pool of ten embryos were laterally mounted in 1% low-melting agarose on glass-bottom dishes (MatTek Corporation P50G-1.5-14-F) and imaged using a Zeiss LSM 880 fast Airyscan microscope, followed by genotyping. Only pairs (wild type and mutant) that were found within the same pool were analysed, to allow comparison between similar developmental stages. Parameters used were ×1 zoom; image size x: 425 mm, y: 425 mm, z: 150 mm. Images were subject to 2D maximum projection in FIJI.

5.6.13 Mathematical modelling

Steady state calculation of Her6 in the perfect adaptation model

The perfect adaptation model can be described by a set of differential equations (Figure C.5A; Table C.14):

$$\frac{\mathrm{d}h}{\mathrm{d}t} = \alpha_h G(X)G(m) - \mu_h h, \tag{5.1}$$

$$\frac{\mathrm{d}X}{\mathrm{d}t} = \alpha_X G(m) - \mu_X X, \tag{5.2}$$

where h is Her6, m is miR-9 and α_h, μ_h, α_X and μ_X are positive real constants which represent the production and degradation rates of Her6 and X, respectively. The negative interaction between each of these model components is given by an arbitrary function G. To identify a possible shape of G, we consider the steady state of Equations (5.1) and (5.2), which leads to

$$h^* = \frac{\alpha_h G(X^*)G(m)}{\mu_h},$$

$$X^* = \frac{\alpha_X G(m)}{\mu_X},$$

which combine to give

$$h^* = \frac{\alpha_h G\left(\frac{\alpha_X G(m)}{\mu_X}\right) G(m)}{\mu_h} = \frac{\alpha_h \mu_X}{\alpha_X \mu_h}$$

if G is defined by the negative interaction, $G(m) = 1/m$. Hence, in our chosen model the steady state of Her6, h^*, is independent of miR-9. To achieve this, we made the assumption that G is a nonlinear negative interaction, which agrees with previous models of miR-9 interactions (Goodfellow et al., 2014). In order to explore the adaptation properties of this network, we made certain simplifications over previous models (Goodfellow et al., 2014) such as omitting the *her6* autorepression, transcriptional delays and noise. Thus, this simplified Her6 network does not reproduce the oscillatory expression of Her6 but instead explores the transition between different stable steady states.

Extended model

The extended system can be described by the following set of differential equations (Figure C.6A; Table C.14):

$$\frac{\mathrm{d}h}{\mathrm{d}t} = \alpha_h G(X) G(m) G^-(Y, h_1, p_1) - \mu_h h, \tag{5.3}$$

$$\frac{\mathrm{d}X}{\mathrm{d}t} = \alpha_X G(m) - \mu_X X, \tag{5.4}$$

$$\frac{\mathrm{d}Y}{\mathrm{d}t} = \alpha_Y G^-(h, h_2, p_2) + \beta_Y G^+(Y, h_3, p_3) - \mu_Y Y, \tag{5.5}$$

$$G(x) = 1/x,$$

$$G^-(p, n, p_0) = \frac{1}{1 + \left(\frac{p}{p_0}\right)^n},$$

$$G^+(p, n, p_0) = \frac{1}{1 + \left(\frac{p}{p_0}\right)^{-n}}.$$

We pre-define the profile of miR-9 expression over time to interrogate both stepwise and linear expression, and then solve the system for h, X, and Y. The parameters α_h, α_X and α_Y represent the basal production rates of h, X, and Y, respectively. Similarly, μ_h, μ_X, and μ_Y represent the degradation rates of h, X, and Y, respectively, and β_Y represents the production rate of Y under self-activation. The h_i and p_i are Hill coefficients and repression thresholds, respectively, for each of the Hill functions. For the activating Hill function $G^+(p, n, p_0)$ with arbitrary input parameter p, Hill coefficient n and repression threshold p_0, as p grows much larger than p_0, G^+ tends to 1, and as p goes to 0, G^+ tends to 0. For G^- the limits are reversed, i.e. G^- tends to 1 for small values of p and goes to 0 for $p \gg p_0$. The Hill coefficient n determines the sensitivity of the function to changes in p, i.e. larger n corresponds to higher sensitivity. All parameters introduced here are constants, and their values are listed in Table C.14. These parameters are chosen such that Y is repressed and has no effect on the system when Her6 is at its high steady state h^*.

5.6.14 Statistical testing

Statistical tests were performed in GraphPad Prism 9. Data were tested for normality with D'Agostino–Pearson test. Discrete scatter plots show the median with 95% confidence interval where multiple independent experiments are analysed. Statistical significance between two datasets was tested with Mann–Whitney test (non-parametric). For paired experiments the data was tested for normality with Kolmogorov–Smirnov test followed by a one-tailed paired t-test. Sample sizes, experiment numbers and P-values <0.05 are reported in each figure legend.

Acknowledgements

We are grateful to Dr Tom Pettini for the advice on smiFISH technique and Dr Laure Bally-Cuif for sharing plasmids. The authors also thank the Biological Services Facility, Bioimaging and Systems Microscopy Facilities of the University of Manchester for technical support.

Competing interests

The authors declare no competing or financial interests.

Author contributions

Conceptualization: X.S., N.P.; Methodology: X.S., M.R., N.P.; Software: J.B., T.M., J.K., M.R.; Validation: X.S., T.M., C.S.M., R.L.; Formal analysis: X.S., C.S.M., J.B., T. M.; Investigation: X.S., R.L., J.L., C.S.M., J.B., T.M.; Resources: J.B., J.K., N.P., X.S.; Data curation: X.S., J.B., T.M.; Writing - original draft: X.S., N.P.; Writing - review & editing: N.P., X.S., J.B., J.K., C.S.M., T.M., M.R.; Visualization: X.S., J.B., T.M., R.L.; Supervision: N.P., X.S; Project administration: N.P., X.S.; Funding acquisition: N.P., J.B., C.S.M.

Funding

This work was supported by a Wellcome Trust Senior Research Fellowship (090868/ Z/09/Z) and a Wellcome Trust Investigator Award (224394/Z/21/Z) to N.P. and a Medical Research Council Career Development Award to C.S.M. (MR/V032534/1). J.B. was supported by a Wellcome Trust Four-Year PhD Studentship in Basic Science (219992/Z/19/Z). Open Access funding was provided by The University of Manchester. Deposited in PMC for immediate release.

Data availability

The mathematical modelling of the network motifs was implemented in the Julia programming language. Code for reproducing the computational figures is available at https://github.com/burtonjosh/StepwiseMir9.

Peer review history

The peer review history is available online at https://journals.biologists.com/dev/article-lookup/doi/10.1242/dev.200474.reviewer-comments.pdf

Chapter 6

Discussion

It is by now well established that gene expression is noisy and heterogeneous across a clonal population (Altschuler & Wu, 2010; Elowitz et al., 2002; Huang, 2009; Phillips et al., 2016; Raj & van Oudenaarden, 2008). More recently, evidence that important information is encoded within gene expression, in the form of oscillations, is becoming increasingly available (Goodfellow et al., 2014; Isomura & Kageyama, 2014; Levine et al., 2013; Manning et al., 2019). Of specific interest is that cell fate specification may depend on the dynamic expression of individual transcription factors. For example, murine embryonic neurogenesis depends on the dynamics of HES5 expression (Manning et al., 2019). Decrypting the information contained in these signals requires a mixture of advanced experimental and theoretical techniques. Success in this approach will elucidate the mechanisms which control differentiation and provide a groundwork from which we can investigate the occurrence of developmental disorders.

Mathematical modelling and parameter inference are becoming increasingly important for analysing and understanding detailed single-cell data. A wide selection of different mathematical approaches towards understanding single-cell data have been developed, but a number of open challenges remain. Methods which reduce dynamic data into summary statistics fail to capture time-dependent heterogeneity and thus will perform poorly as predictors of cell fate specification (Cao & Grima, 2019). Other methods which account for fluctuations over time do not account for the essential mechanisms, like transcriptional delay, which underlie the dynamics (Golightly & Wilkinson, 2011), or only simulate mRNA – not considering protein-mRNA interactions (Dattani & Barahona, 2017). Recent developments which account for delays (Calderazzo et al., 2019) have yielded promising results but have not yet been widely applied. So far these methods have taken advantage of Bayesian inference methods to perform parameter inference, in particular using MCMC to sample from posterior distributions. Efficient and fast implementations of these sampling algorithms are required to be more applicable in practice, especially as the amount of data or the model size increases.

The overall aim of this thesis has been to develop and analyse statistical models of gene expression dynamics, with a focus on a stochastic delayed negative feedback model, in order to understand how oscillatory gene expression dynamics emerge and drive cell state specification in murine brain development. First, we extended the existing literature by developing a method for parameter inference via gradient-based MCMC using a delay-adjusted extended Kalman filter to interpret gene expression time-series data. We showed how this method can be used to inform experimental design choices for live imaging, as well as incorporate data from multiple sources. For example, with only a few additional lines of code, we were able to include snapshot data giving information about mRNA concentration in a population of cells at a fixed time point, which greatly improved the accuracy of inference for certain parameters of interest. We found that although we could infer kinetic parameters for single cells, the uncertainty was too high to distinguish cells with qualitatively different dynamics. Subsequently, we significantly improved the computational implementation of our method by accurately accounting for the fact that the time delay parameter is continuous. This gave us access to higher-order ODE solvers with advanced interpolation, improving the accuracy of inference. We further implemented a variational inference algorithm to improve the speed of inference, which allowed us to infer kinetic parameters more accurately by combining time-series data from multiple cells with qualitatively similar dynamics.

We showed that mature miR-9, which regulates Her6 (Soto et al., 2020), increases in a stepwise manner during zebrafish neurogenesis. In light of these experimental observations, we developed a mathematical model to explore potential mechanisms for this behaviour. We proposed a model that was robust to linear increases in miR-9 but could transition into a new steady state with a sharp, nonlinear increase in miR-9 expression. We used this model to provide an explanation for cell state transitions and predict developmental mechanisms in the early zebrafish hindbrain.

In this section, we discuss the benefits and drawbacks of a filtering approach for parameter inference in stochastic delay models. We offer an alternative approach to filtering in order to perform Bayesian inference. Then we briefly consider alternative MCMC sampling algorithms and consider our work in the context of a modern Bayesian workflow. Lastly, we offer a number of suggestions for extending our model.

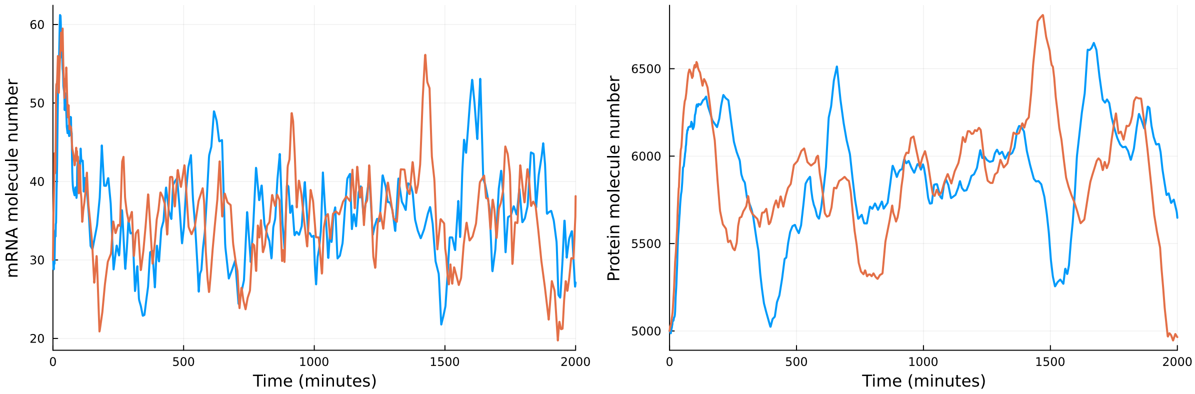
6.1 Simulation and inference in stochastic delay models

Ordinary differential equation models of gene regulation are deterministic – solving the equations with fixed parameters will always give the same result. Conceptually, parameter inference can be performed by simulating data, y_{sim}, using the model for a proposed parameter combination and using, for example, a multivariate normal likelihood of the form

$$p(y_{\text{sim}} \mid \boldsymbol{\theta}) \sim \text{MVNormal}(y_{\text{data}}, \sigma^2 I_d), \tag{6.1}$$

where y_{data} is the observed data, $\boldsymbol{\theta}$ are the unknown model parameters, σ is the (known or unknown) noise term, and I_d is the d-dimensional identity matrix where d is the number of data points.

Conversely, simulating a stochastic model multiple times with the same parameters will yield different results (Figure 6.1). This makes parameter inference much more difficult since there is often no clear way to define the likelihood function. Using the above approach will mean the likelihood will change for the same set of parameters, which will cause problems when doing MCMC sampling. It is also not an accurate mathematical representation of the data, since it fails to account for the intrinsic stochasticity of the model.



(a)

(b)

Figure 6.1: Two simulations of the SDDE defined in Equations (2.21) and (2.22) using the parameter values $(P_0, h, \mu_m, \mu_p, \alpha_m, \alpha_p, \tau) = (5413, 3.17, \log(2)/30, \log(2)/90, 1.86, 1.27, 30)$, showing both (a) mRNA concentration and (b) protein concentration over time.

Since a closed-form likelihood function typically does not exist for stochastic models, a number of methods have been developed for performing inference using either likelihood-free or approximate likelihood approaches (Ghosh et al., 2022; Golightly & Wilkinson, 2011; Manning et al., 2019; C. J. Oates & Mukherjee, 2012; Picchini, 2014; Ryder et al., 2018; Wang & Li, 2018). Amongst these methods, filtering algorithms are the standard choice for inference in SDE's (McKee & Neale, 2019; Mider et al., 2021; Särkkä et al., 2015). In general, filtering algorithms provide an estimate of the state of a time-varying system which is indirectly observed via noisy measurements (Särkkä, 2013). The delay-adjusted Kalman filtering approach presented in Calderazzo et al. (2019) and further developed in this thesis provides an approximate likelihood function for non-linear stochastic delay differential equation (SDDE) models. This has allowed us to infer the parameters of an SDDE using time-dependent data from single cells, which was not previously possible.

However, due to the inclusion of delayed interactions, as model size and complexity increase, the required computational work may become infeasible, even with access to high-performance computing clusters. The emerging field of simulation-based inference methods offers an alternative approach where the underlying model acts as a data simulator (Avecilla et al., 2021; Cranmer et al., 2020; Lueckmann et al., 2021). Neural networks and other machine learning (ML) approaches can then be used to learn the posterior distribution from this simulated data. Once a neural network has been trained for a model, posterior distributions for new data can be evaluated immediately, without the need to redo inference. The key advantage of this approach is that there is no need for a likelihood function, meaning inference can be performed with only the data-generating process. This would be of particular interest in the case of our model, since simulating data is much faster (~ 100 times) than likelihood evaluation via filtering, so the consequences of increased model complexity will not be as severe.

6.2 Inference methods

The results presented in this thesis show that accurate parameter inference for a non-linear delayed stochastic model of gene expression is achievable with a likelihood-based approach, and at the scale of a single cell. In order to achieve these results, we have taken advantage of advanced MCMC sampling algorithms.

The Metropolis random walk algorithm is common due to the simplicity of its implementation. We have shown that using a Kalman filter and random walk algorithm together can allow us to make accurate predictions about specific properties of single cells, such as mean mRNA expression, which agrees with previously reported results (Calderazzo et al., 2019). However, this simplicity presents a trade-off – posterior distributions may reflect complex correlations between model parameters, which makes them difficult to explore with MCMC algorithms. Nonetheless, accurate inference requires a high number of samples. Incorporating geometric properties of the posterior probability space into the inference scheme can result in more efficient sampling (Roberts & Rosenthal, 2001). We took advantage of the Metropolis-adjusted Langevin algorithm in Chapter 3 and the state-of-the-art No-U-Turn sampler in Chapter 4. Both of these algorithms use the gradient of the log-likelihood to guide the exploration of the posterior. In Chapter 3 we found that the use of the gradient based MALA did not appear to speed up the sampling in comparison to RWM. This may, in part, be due to the direct calculation of gradients. A more efficient approach to calculating the gradients is the *adjoint state method*. With this approach, the calculation of the gradient is equivalent to one additional forward model solve. For our direct implementation of the gradients, the scaling was proportional to the number of parameters. Recent work has shown that for small ODE systems (i.e. less than ~ 100 equations) forward-mode automatic differentiation is more efficient than both reverse-mode and continuous adjoint methods (Ma et al., 2021). The Julia programming language, which was used in Chapter 4, has access to these advanced auto-differentiation tools. However, in Chapter 3, the computational experiments were written using the Python programming language, which does not have these approaches readily available. It may be worthwhile to further explore the comparison between the performance of RWM and MALA using this approach. Alternatively, it may be that the lack of gradient information for the transcriptional time delay parameter τ resulted in reduced performance of the MALA sampler. This provided clear motivation for the work in Chapter 4 in order to model τ as a continuous parameter, which resulted in significant performance gains. Extensions to gradient-based MCMC algorithms exist which use Riemann manifolds, eliminating the problem of tuning whilst exploring the space even more efficiently than first-order algorithms (Girolami & Calderhead, 2011). These methods rely on the computation of the Hessian i.e. the second derivative of the log-likelihood function. The added computational cost of these calculations may diminish the efficiency gain in the MCMC sampling. Hence, the additional effort in implementing and potentially deriving closed-form expressions for the second derivative may not be worthwhile.

6.3 Model comparison

Mathematical models should aim to capture key behaviours of a physical process, whilst minimising the number of assumptions made. Model evaluation and comparison are typically done using a number of different information criteria methods (Bayesian, Akaike, deviance, etc.) (Stoica & Selen, 2004). For example, the Akaike information criteria (AIC) is calculated as

$$\text{AIC} = 2k - 2\log(\hat{L}), \tag{6.2}$$

where k is the number of model parameters and $\hat{L}$ is the maximum likelihood value calculated from the posterior samples. For a set of models fit to the same data, the preferred model is the one with the smallest AIC value. This process is highly data-dependent i.e. given a different data set a different model may be selected.

An alternative approach is the iterative model building described by Gelman et al. (2020). This approach advocates starting from a simple model and performing posterior predictive checks to check for misspecification. Additionally, leave-one-out cross-validation (LOO-CV) using Pareto smoothed importance-sampling can be used to identify difficult-to-predict observations, as well as model diagnostics such as under- or over-dispersion (Gelman et al., 2020;

Vehtari et al., 2017; Vehtari et al., 2022).

It is hoped that the open-source implementation of our delay-adjusted Kalman filter will provide a general framework for iterative model development. It is an achievement of our method that it can computationally interface with a number of other open-source Julia packages to perform Bayesian inference, as well as posterior predictive checking and model checking using LOO-CV.

6.4 A number of model extensions are possible

The ability to infer biophysical properties of a cell from live gene expression will allow us to determine how cells make decisions in a wide range of systems. The implications of this work are not limited only to a single gene in a single model system but to any situation involving an auto-repressive feedback motif for which gene expression time course data can be captured. Delay models can be used to understand more general transcriptional-translational feed-forward loops (TTFL), where intermediate interactions are unobserved. For example, Calderazzo et al. (2019) investigated the TTFL that drives the dynamics of the circadian gene *cry1*. A key strength of the work presented in this thesis is the extent of its applicability. In a recent study of a gene interaction network constructed from over 28 human tissues, the auto-negative feedback motif was associated with over 3800 transcription factors (Minchington et al., 2020).

In this work I have defined the delay term as a constant value. Other models that incorporate delay terms represent them as distributions (Calderazzo et al., 2019; Cortez et al., 2021; Unosson et al., 2021). This reflects the natural view that transcription factor molecules are formed and enter the nucleus over an unknown range of delay times. However with this approach a distribution must be chosen for the delay times, and additional hyper-parameters must be inferred which introduces further complexity into the model. One criticism of using a constant delay term is that this assumption might force such a model into a periodic regime (Forger, 2017). In Figure 6 of (Manning et al., 2019), the delayed negative feedback model was analysed for coherence in protein dynamics (a measure of the strength of oscillation where 0 is pure noise and 1 is purely periodic) for a range of parameter values. The model was shown to only exhibit oscillations under very specific conditions, controlled by a bifurcation point between oscillatory and non-oscillatory in the rate of protein degradation which was traversed by noise inherent to the biological system. What was also shown in this figure was that the expected coherence of generated traces was typically between 0.1 and 0.2. In other words, the oscillations that were generated by the model did not tend toward a single dominant frequency when fit to experimental data, but to a range of frequencies that are reflected in the simulated data, and there are parameter regimes in which no oscillations occur despite the presence of a delay term.

Furthermore, it may be the case that the process of transcription and translation are highly regulated and make up the majority of the transcriptional delay time τ, so fluctuations in the transport time of the protein may not have significant or observable downstream effects. In this case the assumption that all TF molecules are formed and enter the nucleus after exactly the same delay may be a simplifying assumption without severe consequences, whilst also allowing for easier inference due to a lower number of model parameters. Posterior predictive checks can also be employed to further investigate whether the model assumptions are sufficient, and this can be done easily with modern probabilistic programming software.

Mathematical models of protein expression dynamics typically interpret transcription as a rate process with Poissonian dynamics. However, promoters can be in on- and off-states, which are associated with different transcription rates. This stochastic pulsatile behaviour has been analysed quantitatively from single-cell reporter imaging data (Harper et al., 2011). The model considered in this thesis does not account for this pulsatile behaviour, and experimental predictions may not provide us with new insights. In this case, our model could incorporate bursty transcription via promoter switching states (Dattani & Barahona, 2017; Gorin & Pachter, 2022; Lipshtat et al., 2006; Thomas et al., 2014) and smFISH snapshot data of mRNA counts could identify dynamics of transcriptional bursting (Halpern et al., 2015). This will improve the accuracy of the method and enable the identification of changes in transcriptional dynamics. Alternatively, methods have been developed which allow for bursting to be simulated via the Langevin equation, which may prove more effective (Yan et al., 2017).

In the example application we focused on in this thesis, HES5 protein expression in murine neural stem cells is not always oscillatory. A number of different dynamics have been identified from experimental data, and single cells can be clustered within a population based on their qualitative gene expression dynamics (Manning et al., 2019). These dynamics correspond to cell fate, therefore inferring parameters from cells exhibiting different dynamics should identify which biophysical properties determine specific cell state transitions. Discovering these properties will allow us to make experimental predictions about which parameters influence differences in gene expression dynamics, and validate the parameter change with experimental techniques appropriate to the prediction.

In HES5 protein expression data, there are often long-term trends present which are not accounted for by the negative feedback model. These trends may be due to interactions with the cell cycle, or the circadian clock. Indeed in ER$^+$ breast cancer cells, oscillations in Hes1 are closely linked with the cell cycle (Sabherwal et al., 2021). In our computational experiments, we detrended experimental data using Gaussian process regression, and only considered data for which there was little to no long term trend. It may be worthwhile to check that the results are not sensitive to the method of trend removal. Furthermore, it may be possible to extend the negative feedback model to account for an additional long term trend, for example by including interactions from micro-RNA's as in Goodfellow et al. (2014), which have been shown to drive long term decreasing trends, as observed in some of the experimental data in Chapter 3 (Manning et al., 2019).

In Chapter 3, we showed that data from a single cell does not provide enough information to detect differences between cells within a population. In Chapter 4, we explored the possibility of performing inference using the combined data from a cluster of cells, which has yielded promising results. Alternative methods exist that measure population variability using data from a whole population of single-cell gene expression (Finkenstädt et al., 2013). Bayesian hierarchical modelling then provides a natural framework to account for the cell-to-cell variability of kinetic parameters and can be used to improve single-cell inference. For example, hierarchical non-stationary models (Monterrubio-Gómez et al., 2020) could be used to inform single-cell parameters using a whole population. More recently, a hierarchical Bayesian approach has been shown to improve estimates of individual cell parameters for a simple delayed birth-death process (Cortez et al., 2021).

In Chapter 5, we developed a perfect adaptation model that focused on the expression of miR-9. Since our aim was to understand the sharp increases in miR-9, we did not explicitly consider interactions which define the Her6 auto-repressive feedback loop. However, miR-9 has been shown to change the dynamic expression of Her6 (Soto et al., 2020), and it may be possible to combine elements of the perfect adaptation model and the auto-repressive feedback loop in order to characterise this behaviour. However, this may require additional experimental data and the identification of the unknown components of this model.

Kinetic parameters of gene expression in single cells do not remain constant, which follows from the fact that key properties of the dynamics, such as expression levels and period of oscillation, tend to change over time. In Figure 6.2 we see gene expression dynamics which are initially aperiodic and then begin to oscillate. The negative feedback model can stochastically change between oscillatory and non-oscillatory dynamics if it is close to a bifurcation point. However, the fact these dynamics have been observed in multiple cells and between multiple experiments suggests a targeted biological mechanism. A natural explanation for this is that one or more of the biophysical properties within the cell have changed. For example, an increase in the protein degradation rate or a decrease in the repression threshold could explain a transition towards oscillatory behaviour. An assumption made by the current method is that these values remain constant throughout the entire time series, which is unlikely to be the case. Extending the method to account for this will allow us to identify the key parameters which influence cell fate decisions.

Methods for inference using regression models with time-varying parameters have been developed with applications to Economics (Dangl & Halling, 2012; Primiceri, 2005), and more recently Gaussian processes with time-varying hyperparameters have been used to estimate non-linear state space models (Liu & Djurić, 2020). These methods would be natural candidates to extend our existing work. This may identify which parameters are changing and also what their time dependency is.

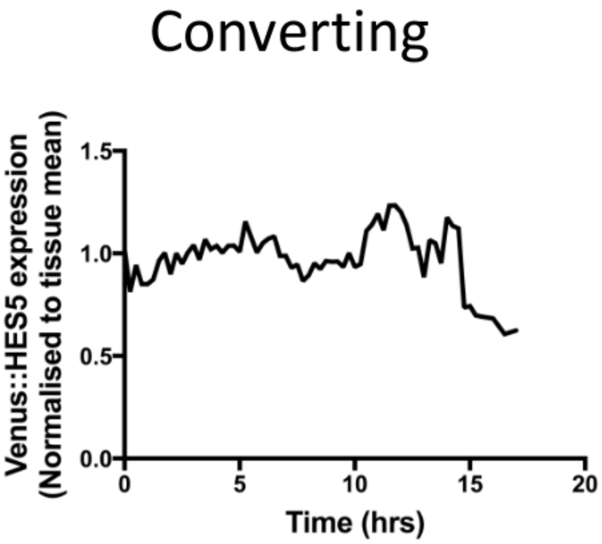


Figure 6.2: An example of a cell exhibiting protein expression dynamics which transition from one regime (aperiodic and noisy) to another (oscillatory).

An approach which may be simpler to implement is to incorporate change-point analysis into the MCMC, which will identify a point in the data at which one or more parameters change, and also determine that change (Eckley et al., 2011). Recent approaches to change-point analysis have been developed that are directly applicable to oscillatory signals (Hadj-Amar et al., 2020).

6.5 Concluding remarks

In this thesis, I have presented novel methods for modelling and analysing gene expression time-series data. The core theme of this work has been to establish a Bayesian framework for the fast inference of kinetic parameters of gene regulation from state-of-the-art time series data, in order to decode the mechanisms driving cell fate transitions during development. Our results have illustrated that methods that capture time-varying behaviour can reveal new information from existing data, making them potentially very powerful tools in studying cell state transitions and kinetics of gene expression. Further, we have demonstrated that our framework can be used to inform the design of live-imaging experiments using simulated data, which will be useful in other contexts. We hope that our framework will pave the way to widespread inference and model comparison on time-series data, leading to novel insights into key mechanisms driving cell fate.

References

Alber, A. B., Paquet, E. R., Biserni, M., Naef, F., & Suter, D. M. (2018). Single Live Cell Monitoring of Protein Turnover Reveals Intercellular Variability and Cell-Cycle Dependence of Degradation Rates [Publisher: Elsevier Inc.]. *Molecular Cell*, *71*(6), 1079–1091.e9. https://doi.org/10.1016/j.molcel.2018.07.023 (cited on p. 55)

Alon, U. (2019). *An Introduction to Systems Biology: Design Principles of Biological Circuits* (2nd ed.) [Pages: 342]. Chapman; Hall/CRC. (Cited on p. 55).

Alonso, D., McKane, A. J., & Pascual, M. (2007). Stochastic amplification in epidemics. *Journal of the Royal Society Interface*, *4*(14), 575–582. https://doi.org/10.1098/rsif.2006.0192 (cited on pp. 21, 189)

Altschuler, S. J., & Wu, L. F. (2010). Cellular Heterogeneity: Do Differences Make a Difference? *Cell*, *141*(4), 559–563. https://doi.org/10.1016/j.cell.2010.04.033 (cited on pp. 41, 137)

Alwin Prem Anand, A., Huber, C., Asnet Mary, J., Gallus, N., Leucht, C., Klafke, R., Hirt, B., & Wizenmann, A. (2018). Expression and function of microRNA-9 in the mid-hindbrain area of embryonic chick. *BMC Developmental Biology*, *18*(1), 3. https://doi.org/10.1186/s12861-017-0159-8 (cited on pp. 114, 115, 205)

Amin, N. D., Senturk, G., Costaguta, G., Driscoll, S., O'Leary, B., Bonanomi, D., & Pfaff, S. L. (2021). A hidden threshold in motor neuron gene networks revealed by modulation of miR-218 dose [Publisher: Elsevier]. *Neuron*, *109*(20), 3252–3267.e6. https://doi.org/10.1016/j.neuron.2021.07.028 (cited on p. 125)

Amores, A., Force, A., Yan, Y.-L., Joly, L., Amemiya, C., Fritz, A., Ho, R. K., Langeland, J., Prince, V., Wang, Y.-L., Westerfield, M., Ekker, M., & Postlethwait, J. H. (1998). Zebrafish hox Clusters and Vertebrate Genome Evolution [Publisher: American Association for the Advancement of Science]. *Science*, *282*(5394), 1711–1714. https://doi.org/10.1126/science.282.5394.1711 (cited on p. 111)

Andrieu, C., Doucet, A., & Holenstein, R. (2010). Particle Markov Chain Monte Carlo Methods. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *72*(3), 269–342. https://doi.org/10.1111/j.1467-9868.2009.00736.x (cited on p. 46)

Andrieu, C., & Thoms, J. (2008). A tutorial on adaptive MCMC. *Statistics and Computing*, *18*(4), 343–373. https://doi.org/10.1007/s11222-008-9110-y (cited on pp. 78, 183)

Artavanis-Tsakonas, S., Rand, M. D., & Lake, R. J. (1999). Notch Signaling: Cell Fate Control and Signal Integration in Development [Publisher: American Association for the Advancement of Science]. *Science*, *284*(5415), 770–776. https://doi.org/10.1126/science.284.5415.770 (cited on p. 20)

Avecilla, G., Chuong, J. N., Li, F., Sherlock, G., Gresham, D., & Ram, Y. (2021). *Simulation-based inference of evolutionary parameters from adaptation dynamics using neural networks* (preprint). Evolutionary Biology. https://doi.org/10.1101/2021.09.30.462581. (Cited on p. 140)

Babtie, A. C., & Stumpf, M. P. (2017). How to deal with parameters for whole-cell modelling. *Journal of the Royal Society Interface*, *14*(133). https://doi.org/10.1098/rsif.2017.0237 (cited on p. 57)

Bansod, S., Kageyama, R., & Ohtsuka, T. (2017). Hes5 regulates the transition timing of neurogenesis and gliogenesis in mammalian neocortical development. *Development*, *144*(17), 3156–3167. https://doi.org/10.1242/dev.147256 (cited on pp. 20, 55)

Barrio, M., Burrage, K., Leier, A., & Tian, T. (2006). Oscillatory Regulation of Hes1: Discrete Stochastic Delay Modelling and Simulation [Publisher: Public Library of Science]. *PLOS Computational Biology*, *2*(9), e117. https://doi.org/10.1371/journal.pcbi.0020117 (cited on p. 23)

Beaumont, M. A., Zhang, W., & Balding, D. J. (2002). Approximate Bayesian computation in population genetics. *Genetics*, *162*(4), 2025–2035. https://doi.org/10.1111/j.1937-2817.2010.tb01236.x (cited on p. 58)

Berthelot, C., Brunet, F., Chalopin, D., Juanchich, A., Bernard, M., Noël, B., Bento, P., Da Silva, C., Labadie, K., Alberti, A., Aury, J.-M., Louis, A., Dehais, P., Bardou, P., Montfort, J., Klopp, C., Cabau, C., Gaspin, C., Thorgaard, G. H., … Guiguen, Y. (2014). The rainbow trout genome provides novel insights into evolution after whole-genome duplication in vertebrates [Number: 1 Publisher: Nature Publishing Group]. *Nature Communications*, *5*(1), 3657. https://doi.org/10.1038/ncomms4657 (cited on p. 125)

Betancourt, M. (2017). A Conceptual Introduction to Hamiltonian Monte Carlo. http://arxiv.org/abs/1701.02434 (cited on pp. 34, 36, 58, 85, 87, 182)

Bonev, B., Pisco, A., & Papalopulu, N. (2011). MicroRNA-9 Reveals Regional Diversity of Neural Progenitors along the Anterior-Posterior Axis [Publisher: Elsevier]. *Developmental Cell*, *20*(1), 19–32. https://doi.org/10.1016/j.devcel.2010.11.018 (cited on pp. 20, 110, 114, 126, 127)

Bonev, B., Stanley, P., & Papalopulu, N. (2012). MicroRNA-9 Modulates Hes1 Ultradian Oscillations by Forming a Double-Negative Feedback Loop [Publisher: Elsevier]. *Cell Reports*, *2*(1), 10–18. https://doi.org/10.1016/j.celrep.2012.05.017 (cited on pp. 22, 77, 110, 111, 126)

Boys, R. J., Wilkinson, D. J., & Kirkwood, T. B. L. (2008). Bayesian inference for a discretely observed stochastic kinetic model. *Statistics and Computing*, *18*(2), 125–135. https://doi.org/10.1007/s11222-007-9043-x (cited on pp. 43, 44)

Brett, T., & Galla, T. (2013). Stochastic processes with distributed delays: Chemical langevin equation and linear-noise approximation. *Physical Review Letters*, *110*(25), 1–5. https://doi.org/10.1103/PhysRevLett.110.250601 (cited on p. 23)

Brett, T., & Galla, T. (2014). Gaussian approximations for stochastic systems with delay: Chemical Langevin equation and application to a Brusselator system. *The Journal of Chemical Physics*, *140*(12), 124112–124112. https://doi.org/10.1063/1.4867786 (cited on pp. 60, 61)

Brock, A., Chang, H., & Huang, S. (2009). Non-genetic heterogeneity a mutation-independent driving force for the somatic evolution of tumours. *Nature Reviews Genetics*, *10*(5), 336–342. https://doi.org/10.1038/nrg2556 (cited on p. 19)

Browning, A. P., Warne, D. J., Burrage, K., Baker, R. E., & Simpson, M. J. (2020). Listen to the noise: Identifiability analysis for stochastic differential equation models in systems biology. *bioRxiv*, 2020.08.10.245233–2020.08.10.245233. https://doi.org/10.1101/2020.08.10.245233 (cited on p. 57)

Burton, J., Manning, C. S., Rattray, M., Papalopulu, N., & Kursawe, J. (2021). Inferring kinetic parameters of oscillatory gene regulation from single cell time-series data. *Journal of The Royal Society Interface*, *18*(182), 20210393. https://doi.org/10.1098/rsif.2021.0393 (cited on pp. 23, 53, 84–87, 89, 91, 96, 98, 100, 102, 201)

Butts, C. L., & Sternberg, E. M. (2009). Flow cytometry as a tool for measurement of steroid hormone receptor protein expression in leukocytes. *Methods in molecular biology (Clifton, N.J.)*, *505*, 35–50. https://doi.org/10.1007/978-1-60327-575-0_3 (cited on p. 40)

Calderazzo, S., Brancaccio, M., & Finkenstädt, B. (2019). Filtering and inference for stochastic oscillators with distributed delays. *Bioinformatics*, *35*(8), 1380–1387. https://doi.org/10.1093/bioinformatics/bty782 (cited on pp. 23, 41, 44, 47, 59, 62, 78, 84–86, 89, 137, 139, 140, 142, 178)

Cao, Z., & Grima, R. (2019). Accuracy of parameter estimation for auto-regulatory transcriptional feedback loops from noisy data. *Journal of the Royal Society Interface*, *16*(153). https://doi.org/10.1098/rsif.2018.0967 (cited on pp. 44, 57, 137)

Castro-Mondragon, J. A., Riudavets-Puig, R., Rauluseviciute, I., Berhanu Lemma, R., Turchi, L., Blanc-Mathieu, R., Lucas, J., Boddie, P., Khan, A., Manosalva Pérez, N., Fornes, O., Leung, T. Y., Aguirre, A., Hammal, F., Schmelter, D., Baranasic, D., Ballester, B., Sandelin, A., Lenhard, B., … Mathelier, A. (2022). JASPAR 2022: The 9th release of the open-access database of transcription factor binding profiles. *Nucleic Acids Research*, *50*(D1), D165–D173. https://doi.org/10.1093/nar/gkab1113 (cited on p. 127)

Chang, P. J., Hsiao, Y. L., Tien, A. C., Li, Y. C., & Pi, H. (2008). Negative-feedback regulation of proneural proteins controls the timing of neural precursor division. *Development*, *135*(18), 3021–3030. https://doi.org/10.1242/dev.021923 (cited on p. 56)

Chen, L., Wang, R., Zhou, T., & Aihara, K. (2005). Noise-induced cooperative behavior in a multicell system. *Bioinformatics*, *21*(11), 2722–2729. https://doi.org/10.1093/bioinformatics/bti392 (cited on p. 21)

Chen, P. Y., Manninga, H., Slanchev, K., Chien, M., Russo, J. J., Ju, J., Sheridan, R., John, B., Marks, D. S., Gaidatzis, D., Sander, C., Zavolan, M., & Tuschl, T. (2005). The developmental miRNA profiles of zebrafish as determined by small RNA cloning [Company: Cold Spring Harbor Laboratory Press Distributor: Cold Spring Harbor Laboratory Press Institution: Cold Spring Harbor Laboratory Press Label: Cold Spring Harbor Laboratory Press Publisher: Cold Spring Harbor Lab]. *Genes & Development*, *19*(11), 1288–1293. https://doi.org/10.1101/gad.1310605 (cited on p. 111)

Cheng, Y.-C., Chiang, M.-C., Shih, H.-Y., Ma, T.-L., Yeh, T.-H., Huang, Y.-C., Lin, C.-Y., & Lin, S.-J. (2015). The transcription factor hairy/E(spl)-related 2 induces proliferation of neural progenitors and regulates neurogenesis and gliogenesis. *Developmental Biology*, *397*(1), 116–128. https://doi.org/10.1016/j.ydbio.2014.10.018 (cited on p. 18)

Choorapoikayil, S., Willems, B., Ströhle, P., & Gajewski, M. (2012). Analysis of her1 and her7 Mutants Reveals a Spatio Temporal Separation of the Somite Clock Module [Publisher: Public Library of Science]. *PLOS ONE*, *7*(6), e39073. https://doi.org/10.1371/journal.pone.0039073 (cited on p. 19)

Chou, J., Provot, S., & Werb, Z. (2010). GATA3 in development and cancer differentiation: Cells GATA have it! *Journal of Cellular Physiology*, *222*(1), 42–49. https://doi.org/10.1002/jcp.21943 (cited on pp. 18, 56)

Clerx, M., Robinson, M., Lambert, B., Lei, C. L., Ghosh, S., Mirams, G. R., & Gavaghan, D. J. (2019). Probabilistic Inference on Noisy Time Series (PINTS). *Journal of Open Research Software*, *7*(1). https://doi.org/10.5334/jors.252 (cited on p. 58)

Coolen, M., Katz, S., & Bally-Cuif, L. (2013). miR-9: A versatile regulator of neurogenesis. *Frontiers in Cellular Neuroscience*, *7*. Retrieved March 4, 2023, from https://www.frontiersin.org/articles/10.3389/fncel.2013.00220 (cited on pp. 110, 119)

Coolen, M., Thieffry, D., Drivenes, O., Becker, T. S., & Bally-Cuif, L. (2012). miR-9 Controls the Timing of Neurogenesis through the Direct Inhibition of Antagonistic Factors [Publisher: Elsevier]. *Developmental Cell*, *22*(5), 1052–1064. https://doi.org/10.1016/j.devcel.2012.03.003 (cited on pp. 22, 110)

Cortez, M. J., Hong, H., Choi, B., Kim, J. K., & Josić, K. (2021). Hierarchical Bayesian models of transcriptional and translational regulation processes with delays. *Bioinformatics*, *38*(1), 187–195. https://doi.org/10.1093/bioinformatics/btab618 (cited on pp. 104, 142, 144)

Cranmer, K., Brehmer, J., & Louppe, G. (2020). The frontier of simulation-based inference [Publisher: Proceedings of the National Academy of Sciences]. *Proceedings of the National Academy of Sciences*, *117*(48), 30055–30062. https://doi.org/10.1073/pnas.1912789117 (cited on p. 140)

Dangl, T., & Halling, M. (2012). Predictive regressions with time-varying coefficients. *Journal of Financial Economics*, *106*(1), 157–181. https://doi.org/10.1016/j.jfineco.2012.04.003 (cited on p. 145)

Dattani, J., & Barahona, M. (2017). Stochastic models of gene transcription with upstream drives: Exact solution and sample path characterization [Publisher: Royal Society]. *Journal of The Royal Society Interface*, *14*(126), 20160833. https://doi.org/10.1098/rsif.2016.0833 (cited on pp. 137, 143)

Delile, J., Rayon, T., Melchionda, M., Edwards, A., Briscoe, J., & Sagner, A. (2019). Single cell transcriptomics reveals spatial and temporal dynamics of gene expression in the developing mouse spinal cord (A. Klein & B. Treutlein, Eds.). *Development*, *146*(12), dev173807. https://doi.org/10.1242/dev.173807 (cited on p. 121)

Diaconis, P., Holmes, S., & Montgomery, R. (2007). Dynamical Bias in the Coin Toss. *SIAM Review*, *49*(2), 211–235. https://doi.org/10.1137/S0036144504446436 (cited on p. 26)

Doucet, A., & Johansen, A. M. (2011). A tutorial on particle filtering and smoothing : Fiteen years later [Num Pages: 1063]. In D. Crisan & B. Rozovskii (Eds.). Oxford University Press. Retrieved July 23, 2023, from http://webcat.warwick.ac.uk/record=b2490036~S1. (Cited on p. 46)

Duan, J., Shi, H., Liu, D., & Yu, H. (2016). Square Root Cubature Kalman Filter-Kalman Filter Algorithm for Intelligent Vehicle Position Estimate. *Procedia Engineering*, *137*, 267–276. https://doi.org/10.1016/j.proeng.2016.01.258 (cited on p. 45)

Duane, S., Kennedy, A. D., Pendleton, B. J., & Roweth, D. (1987). Hybrid Monte Carlo. *Physics Letters B*, *195*(2), 216–222. https://doi.org/10.1016/0370-2693(87)91197-X (cited on pp. 36, 58)

Dubaissi, E., Panagiotaki, N., Papalopulu, N., & Vize, P. D. (2012). Antibody Development and Use in Chromogenic and Fluorescent Immunostaining. In S. HOPPLER & P. D. Vize (Eds.), *Xenopus Protocols: Post-Genomic Approaches* (pp. 411–429). Humana Press. https://doi.org/10.1007/978-1-61779-992-1_23. (Cited on p. 129)

Dublanche, Y., Michalodimitrakis, K., Kümmerer, N., Foglierini, M., & Serrano, L. (2006). Noise in transcription negative feedback loops: Simulation and experimental analysis. *Molecular Systems Biology*, *2*, 1–12. https://doi.org/10.1038/msb4100081 (cited on p. 21)

Durmus, A., Roberts, G. O., Vilmart, G., & Zygalakis, K. C. (2017). Fast langevin based algorithm for MCMC in high dimensions. *Annals of Applied Probability*, *27*(4), 2195–2237. https://doi.org/10.1214/16-AAP1257 (cited on pp. 34, 182)

D'Urso, A., & Brickner, J. H. (2017). Epigenetic transcriptional memory [Publisher: Springer Berlin Heidelberg]. *Current Genetics*, *63*(3), 435–439. https://doi.org/10.1007/s00294-016-0661-8 (cited on p. 79)

Eckley, I. A., Fearnhead, P., & Killick, R. (2011). Analysis of changepoint models. In A. T. Cemgil, D. Barber, & S. Chiappa (Eds.), *Bayesian Time Series Models* (pp. 205–224). Cambridge University Press. https://doi.org/10.1017/CBO9780511984679.011. (Cited on p. 145)

Eldar, A., & Elowitz, M. B. (2010). Functional roles for noise in genetic circuits [Number: 7312 Publisher: Nature Publishing Group]. *Nature*, *467*(7312), 167–173. https://doi.org/10.1038/nature09326 (cited on p. 21)

Elerian, O., Chib, S., & Shephard, N. (2001). Likelihood Inference for Discretely Observed Nonlinear Diffusions [_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1468-0262.00226]. *Econometrica*, *69*(4), 959–993. https://doi.org/10.1111/1468-0262.00226 (cited on p. 44)

Elowitz, M. B., Levine, A. J., Siggia, E. D., & Swain, P. S. (2002). Stochastic gene expression in a single cell. *Science*, *297*(5584), 1183–1186. https://doi.org/10.1126/science.1070919 (cited on pp. 19, 21, 79, 137)

Ernst, C. (2016). Proliferation and Differentiation Deficits are a Major Convergence Point for Neurodevelopmental Disorders [Publisher: Elsevier Ltd]. *Trends in Neurosciences*, *39*(5), 290–299. https://doi.org/10.1016/j.tins.2016.03.001 (cited on p. 18)

Evensen, G. (2003). The Ensemble Kalman Filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, *53*(4), 343–367. https://doi.org/10.1007/s10236-003-0036-9 (cited on p. 46)

Ezer, D., Moignard, V., Göttgens, B., & Adryan, B. (2016). Determining Physical Mechanisms of Gene Expression Regulation from Single Cell Gene Expression Data. *PLoS Computational Biology*, *12*(8), 1–28. https://doi.org/10.1371/journal.pcbi.1005072 (cited on p. 21)

Fan, M., Kuwahara, H., Wang, X., Wang, S., & Gao, X. (2015). Parameter estimation methods for gene circuit modeling from time-series mRNA data: A comparative study. *Briefings in Bioinformatics*, *16*(6), 987–999. https://doi.org/10.1093/bib/bbv015 (cited on p. 58)

Farahi, F., & Yazdi, H. S. (2020). Probabilistic Kalman filter for moving object tracking. *Signal Processing: Image Communication*, *82*, 115751. https://doi.org/10.1016/j.image.2019.115751 (cited on p. 45)

Fearnhead, P., Giagos, V., & Sherlock, C. (2014). Inference for reaction networks using the linear noise approximation [_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/biom.12152]. *Biometrics*, *70*(2), 457–466. https://doi.org/10.1111/biom.12152 (cited on pp. 43, 44)

Fearnhead, P., & Prangle, D. (2012). Constructing summary statistics for approximate Bayesian computation: Semi-automatic approximate Bayesian computation. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, *74*(3), 419–474. https://doi.org/10.1111/j.1467-9868.2011.01010.x (cited on p. 58)

Featherstone, K., Hey, K., Momiji, H., McNamara, A. V., Patist, A. L., Woodburn, J., Spiller, D. G., Christian, H. C., McNeilly, A. S., Mullins, J. J., Finkenstädt, B. F., Rand, D. A., White, M. R., & Davis, J. R. (2016). Spatially coordinated dynamic gene transcription in living pituitary tissue (R. H. Singer, Ed.) [Publisher: eLife Sciences Publications, Ltd]. *eLife*, *5*, e08494. https://doi.org/10.7554/eLife.08494 (cited on p. 86)

Femino, A. M., Fay, F. S., Fogarty, K., & Singer, R. H. (1998). Visualization of single RNA transcripts in situ. *Science*, *280*(5363), 585–590. https://doi.org/10.1126/science.280.5363.585 (cited on pp. 39, 73, 190)

Finkenstädt, B., Woodcock, D. J., Komorowski, M., Harper, C. V., Davis, J. R., White, M. R., & Rand, D. A. (2013). Quantifying intrinsic and extrinsic noise in gene transcription using the linear noise approximation: An application to single cell data. *Annals of Applied Statistics*, *7*(4), 1960–1982. https://doi.org/10.1214/13-AOAS669 (cited on pp. 42, 57, 144)

Forger, D. B. (2017). *Biological Clocks, Rhythms, and Oscillations*. MIT Press. (Cited on p. 142).

Galant, S., Furlan, G., Coolen, M., Dirian, L., Foucher, I., & Bally-Cuif, L. (2016). Embryonic origin and lineage hierarchies of the neural progenitor subtypes building the zebrafish adult midbrain. *Developmental Biology*, *420*(1), 120–135. https://doi.org/10.1016/j.ydbio.2016.09.022 (cited on pp. 22, 110)

Galla, T. (2009). Intrinsic fluctuations in stochastic delay systems: Theoretical description and application to a simple model of gene regulation. *Physical Review E - Statistical, Non-*

linear, and Soft Matter Physics, *80*(2), 1–9. https://doi.org/10.1103/PhysRevE.80.021909 (cited on pp. 23, 60, 61, 68, 71)

Ge, H., Xu, K., & Ghahramani, Z. (2018). Turing: A Language for Flexible Probabilistic Inference [ISSN: 2640-3498]. *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, 1682–1690. Retrieved January 16, 2023, from https://proceedings.mlr.press/v84/ge18b.html (cited on pp. 36, 98, 197)

Gelman, A., B. Carlin, J., S. Stern, H., B. Dunson, D., Vehtari, A., & B. Rubin, D. (2013). *Bayesian Data Analysis* (3rd) [Pages: 675]. CRC Press. (Cited on p. 86).

Gelman, A., Rubin, D. B., Gelman, A., & Rubin, D. B. (1992). Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, *7*(4), 457–472 (cited on p. 188).

Gelman, A., Vehtari, A., Simpson, D., Margossian, C. C., Carpenter, B., Yao, Y., Kennedy, L., Gabry, J., Bürkner, P.-C., & Modrák, M. (2020). Bayesian Workflow [arXiv: 2011.01808]. *arXiv:2011.01808 [stat]*. Retrieved March 21, 2022, from http://arxiv.org/abs/2011.01808 (cited on pp. 79, 141)

Ghosh, S., Birrell, P. J., & Angelis, D. D. (2022). Differentiable Bayesian inference of SDE parameters using a pathwise series expansion of Brownian motion [ISSN: 2640-3498]. *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, 10982–10998. Retrieved March 1, 2023, from https://proceedings.mlr.press/v151/ghosh22a.html (cited on p. 139)

Gilks, W. R., Richardson, S., & Spiegelhalter, D. (Eds.). (1995). *Markov Chain Monte Carlo in Practice*. Chapman; Hall/CRC. https://doi.org/10.1201/b14835. (Cited on pp. 29, 30)

Gillespie, D. T. (2000). The chemical Langevin equation. *The Journal of Chemical Physics*, *113*(1), 297–306. https://doi.org/10.1063/1.481811 (cited on pp. 60, 61)

Gillespie, D. T. (2007). Stochastic Simulation of Chemical Kinetics. *Annual Review of Physical Chemistry*, *58*(1), 35–55. https://doi.org/10.1146/annurev.physchem.58.032806.104637 (cited on pp. 43, 77)

Girolami, M., & Calderhead, B. (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society*, *73*, 123–214. http://onlinelibrary.wiley.com/doi/10.1111/j.1467-9868.2010.00765.x/full%5Cnpapers2://publication/uuid/83554943-4606-4196-9EC9-E2CA01AA5358 (cited on pp. 58, 78, 85, 141)

Giuggioli, L., & Neu, Z. (2019). Fokker-Planck representations of non-Markov Langevin equations: Application to delayed systems. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *377*(2153). https://doi.org/10.1098/rsta.2018.0131 (cited on p. 21)

Godwin, S., Ward, D., Pedone, E., Homer, M., Fletcher, A. G., & Marucci, L. (2017). An extended model for culture-dependent heterogenous gene expression and proliferation dynamics in mouse embryonic stem cells [Publisher: Springer US]. *npj Systems Biology and Applications*, *3*(1), 1–12. https://doi.org/10.1038/s41540-017-0020-5 (cited on p. 79)

Goentoro, L., Shoval, O., Kirschner, M. W., & Alon, U. (2009). The Incoherent Feedforward Loop Can Provide Fold-Change Detection in Gene Regulation [Publisher: Elsevier]. *Molecular Cell*, *36*(5), 894–899. https://doi.org/10.1016/j.molcel.2009.11.018 (cited on pp. 122, 126)

Golightly, A., & Wilkinson, D. J. (2005). Bayesian Inference for Stochastic Kinetic Models Using a Diffusion Approximation [Publisher: [Wiley, International Biometric Society]]. *Biometrics*, *61*(3), 781–788. Retrieved July 23, 2023, from https://www.jstor.org/stable/3695607 (cited on pp. 43, 44)

Golightly, A., & Wilkinson, D. J. (2011). Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo [Publisher: Royal Society]. *Interface Focus*, *1*(6), 807–820. https://doi.org/10.1098/rsfs.2011.0047 (cited on pp. 23, 43, 46, 137, 139)

Gómez-Schiavon, M., Chen, L. F., West, A. E., & Buchler, N. E. (2017). BayFish: Bayesian inference of transcription dynamics from population snapshots of single-molecule RNA FISH in single cells [Publisher: Genome Biology]. *Genome Biology*, *18*(1), 1–12. https://doi.org/10.1186/s13059-017-1297-9 (cited on pp. 21, 41, 57)

Goodfellow, M., Phillips, N. E., Manning, C., Galla, T., & Papalopulu, N. (2014). microRNA input into a neural ultradian oscillator controls emergence and timing of alternative cell states [Number: 1 Publisher: Nature Publishing Group]. *Nature Communications*, *5*(1), 3399. https://doi.org/10.1038/ncomms4399 (cited on pp. 22, 56, 61, 68, 77, 110, 111, 126, 134, 137, 144)

Goodwin, B. C. (1965). Oscillatory behavior in enzymatic control processes. *Advances in Enzyme Regulation*, *3*(100). https://doi.org/10.1016/0065-2571(65)90067-1 (cited on p. 19)

Gorin, G., & Pachter, L. (2022). Modeling bursty transcription and splicing with the chemical master equation. *Biophysical Journal*, *121*(6), 1056–1069. https://doi.org/10.1016/j.bpj.2022.02.004 (cited on p. 143)

Gruenheit, N., Parkinson, K., Brimson, C. A., Kuwana, S., Johnson, E. J., Nagayama, K., Llewellyn, J., Salvidge, W. M., Stewart, B., Keller, T., Zon, W. v., Cotter, S. L., & Thompson, C. R. L. (2018). Cell Cycle Heterogeneity Can Generate Robust Cell Type Proportioning [Publisher: Elsevier]. *Developmental Cell*, *47*(4), 494–508.e4. https://doi.org/10.1016/j.devcel.2018.09.023 (cited on p. 21)

Haario, H., Saksman, E., & Tamminen, J. (2001). An adaptive Metropolis algorithm. *Bernoulli*, *7*(2), 223–242 (cited on p. 182).

Hadj-Amar, B., Rand, B. F., Fiecas, M., Lévi, F., & Huckstepp, R. (2020). Bayesian Model Search for Nonstationary Periodic Time Series. *Journal of the American Statistical Association*, *115*(531), 1320–1335. https://doi.org/10.1080/01621459.2019.1623043 (cited on p. 145)

Halpern, K. B., Tanami, S., Landen, S., Chapal, M., Szlak, L., Hutzler, A., Nizhberg, A., & Itzkovitz, S. (2015). Bursty Gene Expression in the Intact Mammalian Liver [Publisher: Elsevier]. *Molecular Cell*, *58*(1), 147–156. https://doi.org/10.1016/j.molcel.2015.01.027 (cited on p. 143)

Hargreaves, J. C., Annan, J. D., Edwards, N. R., & Marsh, R. (2004). An efficient climate forecasting method using an intermediate complexity Earth System Model and the ensemble Kalman filter. *Climate Dynamics*, *23*(7), 745–760. https://doi.org/10.1007/s00382-004-0471-4 (cited on p. 45)

Harper, C. V., Finkenstädt, B., Woodcock, D. J., Friedrichsen, S., Semprini, S., Ashall, L., Spiller, D. G., Mullins, J. J., Rand, D. A., Davis, J. R. E., & White, M. R. H. (2011). Dynamic Analysis of Stochastic Transcription Cycles (A. Levchenko, Ed.). *PLoS Biology*, *9*(4), e1000607. https://doi.org/10.1371/journal.pbio.1000607 (cited on p. 143)

Hastings, W. K. (1970). Monte carlo sampling methods using Markov chains and their applications. *Biometrika*, *57*(1), 97–109. https://doi.org/10.1093/biomet/57.1.97 (cited on p. 58)

Hatakeyama, J. (2004). Hes genes regulate size, shape and histogenesis of the nervous system by control of the timing of neural stem cell differentiation. *Development*, *131*(22), 5539–5550. https://doi.org/10.1242/dev.01436 (cited on pp. 18, 20, 60)

Heron, E. A., Finkenstädt, B., & Rand, D. A. (2007). Bayesian inference for dynamic transcriptional regulation; the Hes1 system as a case study. *Bioinformatics*, *23*(19), 2596–2603. https://doi.org/10.1093/bioinformatics/btm367 (cited on pp. 23, 44, 84, 85)

Hey, K. L., Momiji, H., Featherstone, K., Davis, J. R. E., White, M. R. H., Rand, D. A., & Finkenstädt, B. (2015). A stochastic transcriptional switch model for single cell imaging data. *Biostatistics (Oxford, England)*, *16*(4), 655–669. https://doi.org/10.1093/biostatistics/kxv010 (cited on p. 44)

Hirata, H., Ohtsuka, T., Bessho, Y., & Kageyama, R. (2000). Generation of structurally and functionally distinct factors from the basic helix-loop-helix gene Hes3 by alternative first exons. *Journal of Biological Chemistry*, *275*(25), 19083–19089. https://doi.org/10.1074/jbc.M001075200 (cited on p. 20)

Hirata, H., Yoshiura, S., Ohtsuka, T., Bessho, Y., Harada, T., Yoshikawa, K., & Kageyama, R. (2002). Oscillatory Expression of the bHLH Factor Hes1 Regulated by a Negative Feedback Loop [Publisher: American Association for the Advancement of Science]. *Science*, *298*(5594), 840–843. https://doi.org/10.1126/science.1074560 (cited on pp. 20, 60, 77, 84, 110)

Hoffman, M., & Ma, Y. (2020). Black-Box Variational Inference as a Parametric Approximation to Langevin Dynamics [ISSN: 2640-3498]. *Proceedings of the 37th International Conference on Machine Learning*, 4324–4341. Retrieved January 10, 2023, from https://proceedings.mlr.press/v119/hoffman20a.html (cited on p. 87)

Hoffman, M. D., & Gelman, A. (2014). The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, *15*, 1593–1623 (cited on pp. 36, 58, 87, 98, 182, 197).

Honkela, A., Peltonen, J., Topa, H., Charapitsa, I., Matarese, F., Grote, K., Stunnenberg, H. G., Reid, G., Lawrence, N. D., & Rattray, M. (2015). Genome-wide modeling of transcription kinetics reveals patterns of RNA production delays. *Proceedings of the National*

Academy of Sciences of the United States of America, *112*(42), 13115–13120. https://doi.org/10.1073/pnas.1420404112 (cited on pp. 56, 57)

Huang, S. (2009). Non-genetic heterogeneity of cells in development: More than just noise. *Development*, *136*(23), 3853–3862. https://doi.org/10.1242/dev.035139 (cited on pp. 18, 39, 137)

Imayoshi, I., Isomura, A., Harima, Y., Kawaguchi, K., Kori, H., Miyachi, H., Fujiwara, T., Ishidate, F., & Kageyama, R. (2013). Oscillatory control of factors determining multipotency and fate in mouse neural progenitors. *Science*, *342*(6163), 1203–1208. https://doi.org/10.1126/science.1242366 (cited on pp. 18–20, 55, 84, 191)

Isomura, A., & Kageyama, R. (2014). Ultradian oscillations and pulses: Coordinating cellular responses and cell fate decisions. *Development (Cambridge)*, *141*(19), 3627–3636. https://doi.org/10.1242/dev.104497 (cited on p. 137)

Iwasaki, T., Takiguchi, R., Hiraiwa, T., Yamada, T. G., Yamazaki, K., Hiroi, N. F., & Funahashi, A. (2020). Neural Differentiation Dynamics Controlled by Multiple Feedback Loops in a Comprehensive Molecular Interaction Network. *Processes*, *8*(166) (cited on p. 56).

Jaillon, O., Aury, J.-M., Brunet, F., Petit, J.-L., Stange-Thomann, N., Mauceli, E., Bouneau, L., Fischer, C., Ozouf-Costaz, C., Bernot, A., Nicaud, S., Jaffe, D., Fisher, S., Lutfalla, G., Dossat, C., Segurens, B., Dasilva, C., Salanoubat, M., Levy, M., … Roest Crollius, H. (2004). Genome duplication in the teleost fish Tetraodon nigroviridis reveals the early vertebrate proto-karyotype [Number: 7011 Publisher: Nature Publishing Group]. *Nature*, *431*(7011), 946–957. https://doi.org/10.1038/nature03025 (cited on p. 111)

Jenkins, D. J., Finkenstädt, B., & Rand, D. A. (2013). A temporal switch model for estimating transcriptional activity in gene expression. *Bioinformatics (Oxford, England)*, *29*(9), 1158–1165. https://doi.org/10.1093/bioinformatics/btt111 (cited on pp. 42, 57, 86)

Jensen, M., Sneppen, K., & Tiana, G. (2003). Sustained oscillations and time delays in gene expression of protein Hes1. *FEBS Letters*, *541*(1-3), 176–177. https://doi.org/10.1016/S0014-5793(03)00279-5 (cited on p. 84)

Jin, L., & Lloyd, R. V. (1997). In situ hybridization: Methods and applications. *Journal of Clinical Laboratory Analysis*, *11*(1), 2–9. https://doi.org/10.1002/(SICI)1098-2825(1997)11:1<2::AID-JCLA2>3.0.CO;2-F (cited on p. 39)

Julier, S., & Uhlmann, J. (2004). Unscented filtering and nonlinear estimation [Conference Name: Proceedings of the IEEE]. *Proceedings of the IEEE*, *92*(3), 401–422. https://doi.org/10.1109/JPROC.2003.823141 (cited on p. 46)

Kageyama, R., Niwa, Y., Isomura, A., González, A., & Harima, Y. (2012). Oscillatory gene expression and somitogenesis. *WIREs Developmental Biology*, *1*(5), 629–641. https://doi.org/10.1002/wdev.46 (cited on p. 18)

Kageyama, R., Ohtsuka, T., Shimojo, H., & Imayoshi, I. (2008). Dynamic Notch signaling in neural progenitor cells and a revised view of lateral inhibition. *Nature Neuroscience*, *11*(11), 1247–1251. https://doi.org/10.1038/nn.2208 (cited on p. 18)

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering, Transactions of the ASME*, *82*(1), 35–45. https://doi.org/10.1115/1.3662552 (cited on pp. 44, 45)

Kelly, A. (1994). *A 3D State Space Formulation of a Navigation Kalman Filter for Autonomous Vehicles:* (tech. rep.). Defense Technical Information Center. Fort Belvoir, VA. https://doi.org/10.21236/ADA282853. (Cited on p. 45)

Khammash, M. H. (2021). Perfect adaptation in biology [Publisher: Elsevier]. *Cell Systems*, *12*(6), 509–521. https://doi.org/10.1016/j.cels.2021.05.020 (cited on pp. 123, 126)

Khodarahmi, M., & Maihami, V. (2023). A Review on Kalman Filter Models. *Archives of Computational Methods in Engineering*, *30*(1), 727–747. https://doi.org/10.1007/s11831-022-09815-7 (cited on p. 46)

Kicheva, A., Bollenbach, T., Ribeiro, A., Pérez Valle, H., Lovell-Badge, R., Episkopou, V., & Briscoe, J. (2014). Coordination of progenitor specification and growth in mouse and chick spinal cord. *Science*, *345*(6204). https://doi.org/10.1126/science.1254927 (cited on p. 20)

Klein, A. M., Mazutis, L., Akartuna, I., Tallapragada, N., Veres, A., Li, V., Peshkin, L., Weitz, D. A., & Kirschner, M. W. (2015). Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells [Publisher: Elsevier Inc.]. *Cell*, *161*(5), 1187–1201. https://doi.org/10.1016/j.cell.2015.04.044 (cited on p. 40)

Kobayashi, T., & Kageyama, R. (2014). *Expression dynamics and functions of hes factors in development and diseases* (1st ed., Vol. 110) [Pages: 283 Publication Title: Current Topics in Developmental Biology]. Elsevier Inc. https://doi.org/10.1016/B978-0-12-405943-6.00007-5. (Cited on pp. 19, 20, 55)

Kojima, K., Fujita, A., Shimamura, T., Imoto, S., & Miyano, S. (2008). Estimation of nonlinear gene regulatory networks via L1 regularized NVAR from time series gene expression data. *Genome informatics. International Conference on Genome Informatics*, *20*, 37–51. https://doi.org/10.1142/9781848163003_0004 (cited on p. 58)

Kozomara, A., Birgaoanu, M., & Griffiths-Jones, S. (2019). miRBase: From microRNA sequences to function. *Nucleic Acids Research*, *47*(D1), D155–D162. https://doi.org/10.1093/nar/gky1141 (cited on p. 125)

Lea, R., Bonev, B., Dubaissi, E., Vize, P. D., & Papalopulu, N. (2012). Multicolor Fluorescent In Situ mRNA Hybridization (FISH) on Whole Mounts and Sections. In S. HOPPLER & P. D. Vize (Eds.), *Xenopus Protocols: Post-Genomic Approaches* (pp. 431–444). Humana Press. https://doi.org/10.1007/978-1-61779-992-1_24. (Cited on p. 128)

Leimkuhler, B., & Reich, S. (2005). *Simulating Hamiltonian Dynamics*. Cambridge University Press. https://doi.org/10.1017/CBO9780511614118. (Cited on p. 36)

Leucht, C., Stigloher, C., Wizenmann, A., Klafke, R., Folchert, A., & Bally-Cuif, L. (2008). MicroRNA-9 directs late organizer activity of the midbrain-hindbrain boundary [Number: 6 Publisher: Nature Publishing Group]. *Nature Neuroscience*, *11*(6), 641–648. https://doi.org/10.1038/nn.2115 (cited on pp. 22, 110, 119)

Levi, F., & Schibler, U. (2007). Circadian Rhythms: Mechanisms and Therapeutic Implications. *Annual Review of Pharmacology and Toxicology*, *47*(1), 593–628. https://doi.org/10.1146/annurev.pharmtox.47.120505.105208 (cited on p. 19)

Levine, J. H., Lin, Y., & Elowitz, M. B. (2013). Functional roles of pulsing in genetic circuits. *Science*, *342*(6163), 1193–1200. https://doi.org/10.1126/science.1239999 (cited on p. 137)

Lewis, J. (2003). Autoinhibition with transcriptional delay: A simple mechanism for the zebrafish somitogenesis oscillator. *Current Biology*, *13*(16), 1398–1408. https://doi.org/10.1016/S0960-9822(03)00534-7 (cited on pp. 19–21, 56, 60, 61, 77, 84)

Liepe, J., Kirk, P., Filippi, S., Toni, T., Barnes, C. P., & Stumpf, M. P. (2014). A framework for parameter estimation and model selection from experimental data in systems biology using approximate Bayesian computation [Publisher: Nature Publishing Group]. *Nature Protocols*, *9*(2), 439–456. https://doi.org/10.1038/nprot.2014.025 (cited on p. 57)

Lillacci, G., & Khammash, M. (2010). Parameter estimation and model selection in computational biology. *PLoS Computational Biology*, *6*(3). https://doi.org/10.1371/journal.pcbi.1000696 (cited on pp. 19, 58)

Lipshtat, A., Loinger, A., Balaban, N. Q., & Biham, O. (2006). Genetic Toggle Switch without Cooperative Binding [Publisher: American Physical Society]. *Physical Review Letters*, *96*(18), 188101. https://doi.org/10.1103/PhysRevLett.96.188101 (cited on p. 143)

Liu, Y., & Djurić, P. M. (2020). Gaussian Process State-Space Models with Time-Varying Parameters and Inducing Points [Publisher: NIH Public Access]. *Proceedings of the ... European Signal Processing Conference (EUSIPCO). EUSIPCO (Conference)*, *2020*, 1462. https://doi.org/10.23919/Eusipco47968.2020.9287481 (cited on p. 145)

Longo, D., & Hasty, J. (2006). Dynamics of single-cell gene expression. *Molecular Systems Biology*, *2*. https://doi.org/10.1038/msb4100110 (cited on pp. 21, 55)

Loos, C. (2016). *Analysis of Single-Cell Data* [Publication Title: Analysis of Single-Cell Data]. https://doi.org/10.1007/978-3-658-13234-7. (Cited on p. 21)

Lueckmann, J.-M., Boelts, J., Greenberg, D. S., Gonçalves, P. J., & Macke, J. H. (2021). Benchmarking Simulation-Based Inference [arXiv:2101.04653 [cs, stat]]. Retrieved July 6, 2022, from http://arxiv.org/abs/2101.04653. (Cited on p. 140)

Lyons, D. A., Guy, A. T., & Clarke, J. D. W. (2003). Monitoring neural progenitor fate through multiple rounds of division in an intact vertebrate brain. *Development*, *130*(15), 3427–3436. https://doi.org/10.1242/dev.00569 (cited on pp. 114, 116)

M. Suter, D., Molina, N., Naef, F., & Schibler, U. (2011). Origins and consequences of transcriptional discontinuity. *Current Opinion in Cell Biology*, *23*, 657–662. https://doi.org/10.1016/j.ceb.2011.09.004 (cited on p. 187)

Ma, Y., Dixit, V., Innes, M., Guo, X., & Rackauckas, C. (2021). A Comparison of Automatic Differentiation and Continuous Sensitivity Analysis for Derivatives of Differential Equation Solutions [arXiv:1812.01892 [cs]]. https://doi.org/10.48550/arXiv.1812.01892. (Cited on p. 141)

Macosko, E. Z., Basu, A., Satija, R., Nemesh, J., Shekhar, K., Goldman, M., Tirosh, I., Bialas, A. R., Kamitaki, N., Martersteck, E. M., Trombetta, J. J., Weitz, D. A., Sanes, J. R., Shalek, A. K., Regev, A., & McCarroll, S. A. (2015). Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets [Publisher: Elsevier]. *Cell*, *161*(5), 1202–1214. https://doi.org/10.1016/j.cell.2015.05.002 (cited on p. 40)

Madelaine, R., Sloan, S. A., Huber, N., Notwell, J. H., Leung, L. C., Skariah, G., Halluin, C., Paşca, S. P., Bejerano, G., Krasnow, M. A., Barres, B. A., & Mourrain, P. (2017). MicroRNA-9 Couples Brain Neurogenesis and Angiogenesis [Publisher: Elsevier]. *Cell Reports*, *20*(7), 1533–1542. https://doi.org/10.1016/j.celrep.2017.07.051 (cited on p. 127)

Magnusson, M., Bürkner, P., & Vehtari, A. (2022). Posteriordb: A set of posteriors for Bayesian inference and probabilistic programming. https://github.com/stan-dev/posteriordb. (Cited on p. 103)

Mahaffy, J. M., & Pao, C. V. (1984). Models of genetic control by repression with time delays and spatial effects. *Journal of Mathematical Biology*, *20*(1), 39–57. https://doi.org/10.1007/BF00275860 (cited on p. 19)

Manning, C. S., Biga, V., Boyd, J., Kursawe, J., Ymisson, B., Spiller, D. G., Sanderson, C. M., Galla, T., Rattray, M., & Papalopulu, N. (2019). Quantitative single-cell live imaging links HES5 dynamics with cell-state and fate in murine neurogenesis [Publisher: Springer US]. *Nature communications*, (In Press). https://doi.org/10.1038/s41467-019-10734-8 (cited on pp. 18–20, 22, 23, 39, 55, 56, 58, 61, 66, 68–70, 78, 79, 84, 85, 88, 102, 137, 139, 142–144, 187, 191)

Marinopoulou, E., Biga, V., Sabherwal, N., Miller, A., Desai, J., Adamson, A. D., & Papalopulu, N. (2021). HES1 protein oscillations are necessary for neural stem cells to exit from quiescence. *iScience*, *24*(10), 103198. https://doi.org/10.1016/j.isci.2021.103198 (cited on p. 18)

Marra, A. N., Chambers, B. E., Chambers, J. M., Drummond, B. E., Adeeb, B. D., Wesselman, H. M., Morales, E. E., Handa, N., Pettini, T., Ronshaugen, M., & Wingert, R. A. (2019). Chapter 11 - Visualizing gene expression during zebrafish pronephros development and regeneration. In T. Weimbs (Ed.), *Methods in Cell Biology* (pp. 183–215). Academic Press. https://doi.org/10.1016/bs.mcb.2019.06.003. (Cited on p. 130)

Mbalawata, I. S., Särkkä, S., & Haario, H. (2013). Parameter estimation in stochastic differential equations with Markov chain Monte Carlo and non-linear Kalman filtering. *Computational Statistics*, *28*(3), 1195–1223. https://doi.org/10.1007/s00180-012-0352-y (cited on pp. 47, 58, 183)

McAdams, H. H., & Adam Arkin. (1997). Stochastic mechanisms in gene expression [Publisher: Proceedings of the National Academy of Sciences]. *Proceedings of the National Academy of Sciences*, *94*(3), 814–819. https://doi.org/10.1073/pnas.94.3.814 (cited on p. 84)

McAdams, H. H., & Arkin, A. (1999). It's a noisy business! Genetic regulation at the nanomolar scale. *Trends in Genetics*, *15*(2), 65–69. https://doi.org/10.1016/S0168-9525(98)01659-X (cited on pp. 43, 57)

McGeary, S. E., Lin, K. S., Shi, C. Y., Pham, T. M., Bisaria, N., Kelley, G. M., & Bartel, D. P. (2019). The biochemical basis of microRNA targeting efficacy [Publisher: American Association for the Advancement of Science]. *Science*, *366*(6472), eaav1741. https://doi.org/10.1126/science.aav1741 (cited on p. 127)

McKee, K. L., & Neale, M. C. (2019). Direct estimation of the parameters of a delayed, intermittent activation feedback model of postural sway during quiet standing (J. D. Touboul, Ed.). *PLOS ONE*, *14*(9), e0222664. https://doi.org/10.1371/journal.pone.0222664 (cited on p. 139)

Michor, F., & Polyak, K. (2010). The origins and implications of intratumor heterogeneity. *Cancer Prevention Research*, *3*(11), 1361–1364. https://doi.org/10.1158/1940-6207.CAPR-10-0234 (cited on p. 21)

Mider, M., Schauer, M., & van der Meulen, F. (2021). Continuous-discrete smoothing of diffusions [arXiv:1712.03807 [stat]]. Retrieved March 1, 2023, from http://arxiv.org/abs/1712.03807. (Cited on p. 139)

Milner, P., Gillespie, C. S., & Wilkinson, D. J. (2013). Moment closure based parameter inference of stochastic kinetic models. *Statistics and Computing*, *23*(2), 287–295. https://doi.org/10.1007/s11222-011-9310-8 (cited on p. 43)

Mil'shtejn, G. N. (1975). Approximate Integration of Stochastic Differential Equations [Publisher: Society for Industrial and Applied Mathematics]. *Theory of Probability & Its Applications*, *19*(3), 557–562. https://doi.org/10.1137/1119062 (cited on p. 88)

Minas, G., Momiji, H., Jenkins, D. J., Costa, M. J., Rand, D. A., & Finkenstädt, B. (2017). ReTrOS: A MATLAB toolbox for reconstructing transcriptional activity from gene and protein expression data [Publisher: BMC Bioinformatics]. *BMC Bioinformatics*, *18*(1), 1–11. https://doi.org/10.1186/s12859-017-1695-8 (cited on pp. 21, 41, 42, 58)

Minchington, T. G., Griffiths-Jones, S., & Papalopulu, N. (2020). Dynamical gene regulatory networks are tuned by transcriptional autoregulation with microRNA feedback [Number: 1 Publisher: Nature Publishing Group]. *Scientific Reports*, *10*(1), 12960. https://doi.org/10.1038/s41598-020-69791-5 (cited on pp. 127, 142)

Mirzaei, F. M., & Roumeliotis, S. I. (2008). A Kalman Filter-Based Algorithm for IMU-Camera Calibration: Observability Analysis and Performance Evaluation [Conference Name: IEEE Transactions on Robotics]. *IEEE Transactions on Robotics*, *24*(5), 1143–1156. https://doi.org/10.1109/TRO.2008.2004486 (cited on p. 45)

Moles, C. G., Mendes, P., & Banga, J. R. (2003). Parameter estimation in biochemical pathways: A comparison of global optimization methods. *Genome Research*, *13*(11), 2467–2474. https://doi.org/10.1101/gr.1262503 (cited on p. 58)

Momiji, H., & Monk, N. A. (2008). Dissecting the dynamics of the Hes1 genetic oscillator. *Journal of Theoretical Biology*, *254*(4), 784–798. https://doi.org/10.1016/j.jtbi.2008.07.013 (cited on pp. 71, 77)

Monk, N. A. (2003). Oscillatory expression of Hes1, p53, and NF-kB driven by transcriptional time delays. *Current Biology*, *13*(16), 1409–1413. https://doi.org/10.1016/S0960-9822(03)00494-9 (cited on pp. 20, 23, 56, 60, 61, 68, 77, 187)

Mönke, G., Sorgenfrei, F. A., Schmal, C., & Granada, A. E. (2020). Optimal time frequency analysis for biological data - pyBOAT. *bioRxiv*, 27–27. https://doi.org/10.1101/2020.04.29.067744 (cited on p. 78)

Monterrubio-Gómez, K., Roininen, L., Wade, S., Damoulas, T., & Girolami, M. (2020). Posterior inference for sparse hierarchical non-stationary models. *Computational Statistics and Data Analysis*, *148*. https://doi.org/10.1016/j.csda.2020.106954 (cited on p. 144)

Moreno-Mateos, M. A., Vejnar, C. E., Beaudoin, J.-D., Fernandez, J. P., Mis, E. K., Khokha, M. K., & Giraldez, A. J. (2015). CRISPRscan: Designing highly efficient sgRNAs for CRISPR-Cas9 targeting in vivo [Number: 10 Publisher: Nature Publishing Group]. *Nature Methods*, *12*(10), 982–988. https://doi.org/10.1038/nmeth.3543 (cited on p. 131)

Munsky, B., Neuert, G., & Van Oudenaarden, A. (2012). Using gene expression noise to understand gene regulation. *Science*, *336*(6078), 183–187. https://doi.org/10.1126/science.1216379 (cited on p. 57)

Nepal, C., Coolen, M., Hadzhiev, Y., Cussigh, D., Mydel, P., Steen, V. M., Carninci, P., Andersen, J. B., Bally-Cuif, L., Müller, F., & Lenhard, B. (2016). Transcriptional, post-transcriptional and chromatin-associated regulation of pri-miRNAs, pre-miRNAs and moRNAs. *Nucleic Acids Research*, *44*(7), 3070–3081. https://doi.org/10.1093/nar/gkv1354 (cited on pp. 111, 114, 128, 204)

Novák, B., & Tyson, J. J. (2008). Design principles of biochemical oscillators [Number: 12 Publisher: Nature Publishing Group]. *Nature Reviews Molecular Cell Biology*, *9*(12), 981–991. https://doi.org/10.1038/nrm2530 (cited on pp. 55, 110)

Oates, A. C., & Ho, R. K. (2002). Hairy/E(spl)-related (Her) genes are central components of the segmentation oscillator and display redundancy with the Delta/Notch signaling pathway in the formation of anterior segmental boundaries in the zebrafish. *Development*, *129*(12), 2929–2946. https://doi.org/10.1242/dev.129.12.2929 (cited on p. 19)

Oates, C. J., & Mukherjee, S. (2012). Network inference and biological dynamics [arXiv:1112.1047 [stat]]. *The Annals of Applied Statistics*, *6*(3). https://doi.org/10.1214/11-AOAS532 (cited on p. 139)

Ochi, S., Imaizumi, Y., Shimojo, H., Miyachi, H., & Kageyama, R. (2020). Oscillatory expression of Hes1 regulates cell proliferation and neuronal differentiation in the embryonic brain. *Development*, *147*(4), dev182204. https://doi.org/10.1242/dev.182204 (cited on p. 18)

Ohtsuka, T., Ishibashi, M., Gradwohl, G., Nakanishi, S., Guillemot, F., & Kageyama, R. (1999). Hes1 and Hes5 as Notch effectors in mammalian neuronal differentiation. *The EMBO Journal*, *18*(8), 2196–2207. https://doi.org/10.1093/emboj/18.8.2196 (cited on p. 20)

Pedregosa, F., Grisel, O., Weiss, R., Passos, A., Brucher, M., Varoquax, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., & Brucher, M. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830 (cited on p. 63).

Pepe-Mooney, B. J., Dill, M. T., Alemany, A., Ordovas-Montanes, J., Matsushita, Y., Rao, A., Sen, A., Miyazaki, M., Anakk, S., Dawson, P. A., Ono, N., Shalek, A. K., van Oudenaarden, A., & Camargo, F. D. (2019). Single-Cell Analysis of the Liver Epithelium Reveals Dynamic Heterogeneity and an Essential Role for YAP in Homeostasis and

Regeneration [Publisher: Elsevier Inc.]. *Cell Stem Cell*, *25*(1), 23–38.e8. https://doi.org/10.1016/j.stem.2019.04.004 (cited on p. 55)

Phillips, N. E., Manning, C., Papalopulu, N., & Rattray, M. (2017). Identifying stochastic oscillations in single-cell live imaging time series using Gaussian processes. *PLoS Computational Biology*, *13*(5), 1–30. https://doi.org/10.1371/journal.pcbi.1005479 (cited on pp. 20, 78)

Phillips, N. E., Manning, C. S., Pettini, T., Biga, V., Marinopoulou, E., Stanley, P., Boyd, J., Bagnall, J., Paszek, P., Spiller, D. G., White, M. R., Goodfellow, M., Galla, T., Rattray, M., & Papalopulu, N. (2016). Stochasticity in the miR-9/Hes1 oscillatory network can account for clonal heterogeneity in the timing of differentiation (A. Sánchez Alvarado, Ed.) [Publisher: eLife Sciences Publications, Ltd]. *eLife*, *5*, e16118. https://doi.org/10.7554/eLife.16118 (cited on pp. 22, 23, 55, 56, 61, 77, 111, 126, 137, 187, 189)

Picchini, U. (2014). Inference for SDE Models via Approximate Bayesian Computation. *Journal of Computational and Graphical Statistics*, *23*(4), 1080–1100. https://doi.org/10.1080/10618600.2013.866048 (cited on p. 139)

Picot, J., Guerin, C. L., Le Van Kim, C., & Boulanger, C. M. (2012). Flow cytometry: Retrospective, fundamentals and recent instrumentation. *Cytotechnology*, *64*(2), 109–130. https://doi.org/10.1007/s10616-011-9415-0 (cited on p. 40)

Politi, A. Z., Cai, Y., Walther, N., Hossain, M. J., Koch, B., Wachsmuth, M., & Ellenberg, J. (2018). Quantitative mapping of fluorescently tagged cellular proteins using FCS-calibrated four-dimensional imaging [Publisher: Nature Publishing Group]. *Nature Protocols*, *13*(6), 1445–1464. https://doi.org/10.1038/nprot.2018.040 (cited on p. 40)

Pollard, S. M. (2013). In vitro expansion of fetal neural progenitors as adherent cell lines. *Methods in Molecular Biology*, *1059*, 13–24. https://doi.org/10.1007/978-1-62703-574-3_2 (cited on p. 191)

Powell, M. J. D. (1964). An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, *7*, 155–162 (cited on p. 181).

Primiceri, G. E. (2005). Time Varying Structural Vector Autoregressions and Monetary Policy. *The Review of Economic Studies*, *72*(3), 821–852. https://doi.org/10.1111/j.1467-937X.2005.00353.x (cited on p. 145)

Purvis, J. E., Karhohs, K. W., Mock, C., Batchelor, E., Loewer, A., & Lahav, G. (2012). P53 Dynamics Control Cell Fate. *Science*, *336*(6087), 1440–1444. https://doi.org/10.1126/science.1218351 (cited on pp. 18, 55, 56, 84)

Pyne, S., Hu, X., Wang, K., Rossin, E., Lin, T.-i., Maier, L. M., Baecher-allan, C., Mclachlan, G. J., Tamayo, P., Hafler, D. A., Jager, P. L. D., & Mesirov, J. P. (2009). Automated high-dimensional flow cytometric data analysis. *PNAS*, *106*(21), 8519–8524 (cited on p. 39).

Rackauckas, C., & Nie, Q. (2017). DifferentialEquations.jl – A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia [Number: 1 Publisher: Ubiquity Press]. *Journal of Open Research Software*, *5*(1), 15. https://doi.org/10.5334/jors.151 (cited on p. 91)

Raj, A., van den Bogaard, P., Rifkin, S. A., van Oudenaarden, A., & Tyagi, S. (2008). Imaging individual mRNA molecules using multiple singly labeled probes. *Nature Methods*, *5*(10), 877–879. https://doi.org/10.1038/nmeth.1253 (cited on pp. 39, 73, 190)

Raj, A., & van Oudenaarden, A. (2008). Nature, Nurture, or Chance: Stochastic Gene Expression and Its Consequences. *Cell*, *135*(2), 216–226. https://doi.org/10.1016/j.cell.2008.09.050 (cited on pp. 18, 19, 21, 84, 137)

Ramachandra Murthy, K., Singh, S., Tuck, D., & Varadan, V. (2019). Chapter 8 - Bayesian Item Response Theory for Cancer Biomarker Discovery. In A. S. R. Srinivasa Rao & C. R. Rao (Eds.), *Handbook of Statistics* (pp. 355–404). Elsevier. https://doi.org/10.1016/bs.host.2018.09.005. (Cited on p. 58)

Raser, J. M., & O'Shea, E. K. (2005). Noise in gene expression: Origins, consequences, and control. *Science*, *309*(5743), 2010–2013. https://doi.org/10.1126/science.1105891 (cited on p. 21)

Rasmussen, C. E., & Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning*. https://doi.org/10.7551/mitpress/3206.001.0001. (Cited on p. 63)

Revels, J., Lubin, M., & Papamarkou, T. (2016). Forward-Mode Automatic Differentiation in Julia [arXiv:1607.07892 [cs]]. https://doi.org/10.48550/arXiv.1607.07892. (Cited on p. 105)

Roberts, G. O., Stramer, O., & Science, A. (2003). Langevin Diffusions and Metropolis-Hastings Algorithms. *Computing*, 337–357 (cited on p. 182).

Roberts, G. O., & Rosenthal, J. S. (2001). Optimal Scaling for Various Metropolis-Hastings Algorithms. *Statistical Science*, *16*(4), 351–367. https://doi.org/10.1214/ss/1015346320 (cited on pp. 32, 33, 78, 140, 183)

Roberts, G. O., & Tweedie, R. L. (1996). Exponential Convergence of Langevin Distributions and Their Discrete Approximations. *Bernoulli*, *2*(4), 341–341. https://doi.org/10.2307/3318418 (cited on pp. 33, 58)

Rodriguez-Otero, P., Román-Gómez, J., Vilas-Zornoza, A., José-Eneriz, E. S., Martín-Palanco, V., Rifón, J., Torres, A., Calasanz, M. J., Agirre, X., & Prosper, F. (2011). Deregulation of FGFR1 and CDK6 oncogenic pathways in acute lymphoblastic leukaemia harbouring epigenetic modifications of the MIR9 family. *British Journal of Haematology*, *155*(1), 73–83. https://doi.org/10.1111/j.1365-2141.2011.08812.x (cited on p. 111)

Roesch, E., & Stumpf, M. P. (2019). Parameter inference in dynamical systems with co-dimension 1 bifurcations. *Royal Society Open Science*, *6*(10). https://doi.org/10.1101/623413 (cited on p. 71)

Rudkin, G. T., & Stollar, B. D. (1977). High resolution detection of DNA-RNA hybrids in situ by indirect immunofluorescence [29]. *Nature*, *265*(5593), 472–473. https://doi.org/10.1038/265472a0 (cited on p. 39)

Ryder, T., Golightly, A., McGough, A. S., & Prangle, D. (2018). Black-Box Variational Inference for Stochastic Differential Equations [ISSN: 2640-3498]. *Proceedings of the 35th International Conference on Machine Learning*, 4423–4432. Retrieved March 1, 2023, from https://proceedings.mlr.press/v80/ryder18a.html (cited on p. 139)

Sabherwal, N., Rowntree, A., Marinopoulou, E., Pettini, T., Hourihane, S., Thomas, R., Soto, X., Kursawe, J., & Papalopulu, N. (2021). Differential phase register of Hes1 oscillations with mitoses underlies cell-cycle heterogeneity in ER+ breast cancer cells [Publisher: Proceedings of the National Academy of Sciences]. *Proceedings of the National Academy of Sciences*, *118*(45), e2113527118. https://doi.org/10.1073/pnas.2113527118 (cited on p. 143)

Sagner, A., Zhang, I., Watson, T., Lazaro, J., Melchionda, M., & Briscoe, J. (2021). A shared transcriptional code orchestrates temporal patterning of the central nervous system [Publisher: Public Library of Science]. *PLOS Biology*, *19*(11), e3001450. https://doi.org/10.1371/journal.pbio.3001450 (cited on p. 127)

Salvatier, J., Wiecki, T. V., & Fonnesbeck, C. (2016). Probabilistic programming in Python using PyMC3 [Publisher: PeerJ Inc.]. *PeerJ Computer Science*, *2*, e55. https://doi.org/10.7717/peerj-cs.55 (cited on p. 36)

Särkkä, S. (2013). *Bayesian Filtering and Smoothing*. Cambridge University Press. https://doi.org/10.1017/CBO9781139344203. (Cited on pp. 45–47, 61, 139, 181)

Särkkä, S., Hartikainen, J., Mbalawata, I. S., & Haario, H. (2015). Posterior inference on parameters of stochastic differential equations via non-linear Gaussian filtering and adaptive MCMC. *Statistics and Computing*, *25*(2), 427–437. https://doi.org/10.1007/s11222-013-9441-1 (cited on p. 139)

Schroeder, T. (2011). Long-term single-cell imaging of mammalian stem cells. *Nature Methods*, *8*(4), S30–S35. https://doi.org/10.1038/nmeth.1577 (cited on p. 39)

Schwanhüusser, B., Busse, D., Li, N., Dittmar, G., Schuchhardt, J., Wolf, J., Chen, W., & Selbach, M. (2011). Global quantification of mammalian gene expression control. *Nature*, *473*(7347), 337–342. https://doi.org/10.1038/nature10098 (cited on pp. 56, 187)

Shen, B., Zhang, W., Zhang, J., Zhou, J., Wang, J., Chen, L., Wang, L., Hodgkins, A., Iyer, V., Huang, X., & Skarnes, W. C. (2014). Efficient genome modification by CRISPR-Cas9 nickase with minimal off-target effects. *Nature Methods*, *11*(4), 399–402. https://doi.org/10.1038/nmeth.2857 (cited on p. 40)

Shen-Orr, S. S., Milo, R., Mangan, S., & Alon, U. (2002). Network motifs in the transcriptional regulation network of Escherichia coli [Number: 1 Publisher: Nature Publishing Group]. *Nature Genetics*, *31*(1), 64–68. https://doi.org/10.1038/ng881 (cited on p. 122)

Shibata, M., Kurokawa, D., Nakao, H., Ohmura, T., & Aizawa, S. (2008). MicroRNA-9 Modulates Cajal–Retzius Cell Differentiation by Suppressing Foxg1 Expression in Mouse Medial Pallium [Publisher: Society for Neuroscience Section: Brief Communications]. *Journal of Neuroscience*, *28*(41), 10415–10421. https://doi.org/10.1523/JNEUROSCI.3219-08.2008 (cited on p. 110)

Shibata, M., Nakao, H., Kiyonari, H., Abe, T., & Aizawa, S. (2011). MicroRNA-9 Regulates Neurogenesis in Mouse Telencephalon by Targeting Multiple Transcription Factors [Publisher: Society for Neuroscience Section: Articles]. *Journal of Neuroscience*, *31*(9), 3407–3422. https://doi.org/10.1523/JNEUROSCI.5085-10.2011 (cited on pp. 110, 111)

Shimojo, H., Ohtsuka, T., & Kageyama, R. (2008). Oscillations in Notch Signaling Regulate Maintenance of Neural Progenitors [Publisher: Elsevier]. *Neuron*, *58*(1), 52–64. https://doi.org/10.1016/j.neuron.2008.02.014 (cited on pp. 19, 20, 110)

Singer, H. (2002). Parameter estimation of nonlinear stochastic differential equations: Simulated maximum likelihood versus extended kalman filter and itô-taylor expansion. *Journal of Computational and Graphical Statistics*, *11*(4), 972–995. https://doi.org/10.1198/106186002808 (cited on pp. 46, 61)

Singer, H. (2006). Continuous-Discrete Unscented Kalman Filtering. (January 2006), 1–28 (cited on p. 47).

Singh, A. (2011). Negative feedback through mRNA provides the best control of gene-expression noise [Publisher: IEEE]. *IEEE Transactions on Nanobioscience*, *10*(3), 194–200. https://doi.org/10.1109/TNB.2011.2168826 (cited on p. 21)

Singh, J., & Padgett, R. A. (2009). Rates of in situ transcription and splicing in large human genes [Number: 11 Publisher: Nature Publishing Group]. *Nature Structural & Molecular Biology*, *16*(11), 1128–1133. https://doi.org/10.1038/nsmb.1666 (cited on p. 187)

Soroldoni, D., & Oates, A. C. (2011). Live transgenic reporters of the vertebrate embryo's Segmentation Clock [Publisher: Elsevier Ltd]. *Current Opinion in Genetics and Development*, *21*(5), 600–605. https://doi.org/10.1016/j.gde.2011.09.006 (cited on pp. 18, 55, 56)

Soto, X., Biga, V., Kursawe, J., Lea, R., Doostdar, P., & Papalopulu, N. (2019). miR-9 mediated noise optimization of the her6 oscillator is needed for cell state progression in the Zebrafish hindbrain. *bioRxiv*. https://doi.org/10.1101/608604 (cited on pp. 18, 39)

Soto, X., Biga, V., Kursawe, J., Lea, R., Doostdar, P., Thomas, R., & Papalopulu, N. (2020). Dynamic properties of noise and Her6 levels are optimized by miR-9, allowing the decoding of the Her6 oscillator [Publisher: John Wiley & Sons, Ltd]. *The EMBO Journal*, *39*(12), e103558. https://doi.org/10.15252/embj.2019103558 (cited on pp. 19, 22, 55, 56, 77, 79, 84, 110, 112, 113, 119, 123–126, 128, 138, 144)

Soto, X., Burton, J., Manning, C. S., Minchington, T., Lea, R., Lee, J., Kursawe, J., Rattray, M., & Papalopulu, N. (2022). Sequential and additive expression of miR-9 precursors control timing of neurogenesis. *Development*, *149*(19), dev200474. https://doi.org/10.1242/dev.200474 (cited on pp. 24, 108)

Spudich, J. L., & Koshland, D. E. (1976). Non-genetic individuality: Chance in the single cell [Number: 5568 Publisher: Nature Publishing Group]. *Nature*, *262*(5568), 467–471. https://doi.org/10.1038/262467a0 (cited on pp. 21, 84)

Stan Development Team. (2023). Stan Modeling Language Users Guide and Reference Manual. https://mc-stan.org. (Cited on p. 36)

Stoica, P., & Selen, Y. (2004). Model-order selection: A review of information criterion rules [Conference Name: IEEE Signal Processing Magazine]. *IEEE Signal Processing Magazine*, *21*(4), 36–47. https://doi.org/10.1109/MSP.2004.1311138 (cited on p. 141)

Swain, P. S., Elowitz, M. B., & Siggia, E. D. (2002). Intrinsic and extrinsic contributions to stochasticity in gene expression. *Proceedings of the National Academy of Sciences of the United States of America*, *99*(20), 12795–12800. https://doi.org/10.1073/pnas.162041399 (cited on p. 21)

Takebayashi, K., Sasai, Y., Sakai, Y., Watanabe, T., Nakanishi, S., & Kageyama, R. (1994). Structure, chromosomal locus, and promoter analysis of the gene encoding the mouse helix-loop-helix factor HES-1. Negative autoregulation through the multiple N box elements. *Journal of Biological Chemistry*, *269*(7), 5150–5156. https://doi.org/10.1016/S0021-9258(17)37668-8 (cited on p. 84)

Tambalo, M., Mitter, R., & Wilkinson, D. G. (2020). A single cell transcriptome atlas of the developing zebrafish hindbrain. *Development*, *147*(6), dev184143. https://doi.org/10.1242/dev.184143 (cited on pp. 111, 116, 119)

Tan, S. L., Ohtsuka, T., González, A., & Kageyama, R. (2012). MicroRNA9 regulates neural stem cell differentiation by controlling Hes1 expression dynamics in the developing brain. *Genes to Cells*, *17*(12), 952–961. https://doi.org/10.1111/gtc.12009 (cited on p. 55)

Thisse, C., & Thisse, B. (2008). High-resolution in situ hybridization to whole-mount zebrafish embryos [Number: 1 Publisher: Nature Publishing Group]. *Nature Protocols*, *3*(1), 59–69. https://doi.org/10.1038/nprot.2007.514 (cited on p. 128)

Thomas, P., Popović, N., & Grima, R. (2014). Phenotypic switching in gene regulatory networks [Publisher: Proceedings of the National Academy of Sciences]. *Proceedings of the National Academy of Sciences*, *111*(19), 6994–6999. https://doi.org/10.1073/pnas.1400049111 (cited on p. 143)

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., & Stumpf, M. P. (2009). Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, *6*(31), 187–202. https://doi.org/10.1098/rsif.2008.0172 (cited on p. 57)

Torres-Padilla, M. E., & Chambers, I. (2014). Transcription factor heterogeneity in pluripotent stem cells: A stochastic advantage. *Development (Cambridge)*, *141*(11), 2173–2181. https://doi.org/10.1242/dev.102624 (cited on p. 21)

Tsang, J., Zhu, J., & Oudenaarden, A. v. (2007). MicroRNA-Mediated Feedback and Feedforward Loops Are Recurrent Network Motifs in Mammals [Publisher: Elsevier]. *Molecular Cell*, *26*(5), 753–767. https://doi.org/10.1016/j.molcel.2007.05.018 (cited on p. 127)

Tsutsumi, S., Matsuda, S., Akaike, A., Okano, H., Aburatani, H., Matsuzaki, Y., Okano, H. J., Saga, Y., Kuwako, K.-i., & Sugimoto, H. (2012). Sox21 Promotes Hippocampal Adult Neurogenesis via the Transcriptional Repression of the Hes5 Gene. *Journal of Neuroscience*, *32*(36), 12543–12557. https://doi.org/10.1523/jneurosci.5803-11.2012 (cited on p. 20)

Tunnacliffe, E., & Chubb, J. R. (2020). What Is a Transcriptional Burst? [Publisher: Elsevier Ltd]. *Trends in Genetics*, *36*(4), 288–297. https://doi.org/10.1016/j.tig.2020.01.003 (cited on p. 77)

Unosson, M., Brancaccio, M., Hastings, M., Johansen, A. M., & Finkenstädt, B. (2021). A spatio-temporal model to reveal oscillator phenotypes in molecular clocks: Parameter estimation elucidates circadian gene transcription dynamics in single-cells (A. Csikász-Nagy, Ed.). *PLOS Computational Biology*, *17*(12). https://doi.org/10.1371/journal.pcbi.1009698 (cited on pp. 86, 142)

Vallejos, C. A., Richardson, S., & Marioni, J. C. (2016). Beyond comparisons of means: Understanding changes in gene expression at the single-cell level [Publisher: Genome Biology]. *Genome Biology*, *17*(1), 1–14. https://doi.org/10.1186/s13059-016-0930-3 (cited on p. 57)

Van der Merwe, R., & Wan, E. (2001). The square-root unscented Kalman filter for state and parameter-estimation [ISSN: 1520-6149]. *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, *6*, 3461–3464 vol.6. https://doi.org/10.1109/ICASSP.2001.940586 (cited on p. 46)

Van Kampen, N. G. (1992). *Stochastic Processes in Physics and Chemistry* (1st). Elsevier. (Cited on p. 90).

Vehtari, A., Gelman, A., & Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC [arXiv:1507.04544 [stat]]. *Statistics and Computing*, *27*(5), 1413–1432. https://doi.org/10.1007/s11222-016-9696-4 (cited on p. 141)

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., & Bürkner, P.-C. (2020). Rank-Normalization, Folding, and Localization: An Improved R for Assessing Convergence of MCMC. *Bayesian Analysis*, 1–27. https://doi.org/10.1214/20-ba1221 (cited on pp. 38, 99, 188)

Vehtari, A., Simpson, D., Gelman, A., Yao, Y., & Gabry, J. (2022). Pareto Smoothed Importance Sampling [arXiv:1507.02646 [stat]]. https://doi.org/10.48550/arXiv.1507.02646. (Cited on p. 142)

Walker, J. C., & Harland, R. M. (2008). Expression of microRNAs during embryonic development of Xenopus tropicalis. *Gene Expression Patterns*, *8*(6), 452–456. https://doi.org/10.1016/j.gep.2008.03.002 (cited on p. 111)

Wang, H., & Li, J. (2018). Adaptive Gaussian process approximation for Bayesian inference with expensive likelihood functions [arXiv:1703.09930 [stat]]. Retrieved March 1, 2023, from http://arxiv.org/abs/1703.09930. (Cited on p. 139)

Warne, D. J., Baker, R. E., & Simpson, M. J. (2019). Simulation and inference algorithms for stochastic biochemical reaction networks: From basic concepts to state-of-the-art. *Journal of the Royal Society Interface*, *16*(151). https://doi.org/10.1098/rsif.2018.0943 (cited on pp. 57, 58)

Weinreb, C., Wolock, S., Tusi, B. K., Socolovsky, M., & Klein, A. M. (2018). Fundamental limits on dynamic inference from single-cell snapshots. *Proceedings of the National Academy of Sciences of the United States of America*, *115*(10), E2467–E2476. https://doi.org/10.1073/pnas.1714723115 (cited on p. 41)

Weiss, J. N. (1997). The Hill equation revisited: Uses and misuses. *The FASEB Journal*, *11*(11), 835–841. https://doi.org/10.1096/fasebj.11.11.9285481 (cited on p. 61)

White, R. J., Collins, J. E., Sealy, I. M., Wali, N., Dooley, C. M., Digby, Z., Stemple, D. L., Murphy, D. N., Billis, K., Hourlier, T., Füllgrabe, A., Davis, M. P., Enright, A. J., & Busch-Nentwich, E. M. (2017). A high-resolution mRNA expression time course of embryonic development in zebrafish (D. Y. Stainier, Ed.) [Publisher: eLife Sciences

Publications, Ltd]. *eLife*, *6*, e30860. https://doi.org/10.7554/eLife.30860 (cited on pp. 114, 131, 203)

Wilkinson, D. J. (2018). *Stochastic Modelling for Systems Biology, Third Edition* (3rd ed.). Chapman; Hall/CRC. https://doi.org/10.1201/9781351000918. (Cited on p. 43)

Wilkinson, R. N., Elworthy, S., Ingham, P. W., & van Eeden, F. J. (2013). A method for high-throughput PCR-based genotyping of larval zebrafish tail biopsies [Publisher: Future Science]. *BioTechniques*, *55*(6), 314–316. https://doi.org/10.2144/000114116 (cited on p. 132)

Wolpert, D. M., & Ghahramani, Z. (2000). Computational principles of movement neuroscience. *Nature Neuroscience*, *3*(S11), 1212–1217. https://doi.org/10.1038/81497 (cited on p. 45)

Wong, S. F. L., Agarwal, V., Mansfield, J. H., Denans, N., Schwartz, M. G., Prosser, H. M., Pourquié, O., Bartel, D. P., Tabin, C. J., & McGlinn, E. (2015). Independent regulation of vertebral number and vertebral identity by microRNA-196 paralogs [Publisher: Proceedings of the National Academy of Sciences]. *Proceedings of the National Academy of Sciences*, *112*(35), E4884–E4893. https://doi.org/10.1073/pnas.1512655112 (cited on p. 126)

Xiong, K., Wei, C. L., & Liu, L. D. (2015). Robust multiple model adaptive estimation for spacecraft autonomous navigation. *Aerospace Science and Technology*, *42*, 249–258. https://doi.org/10.1016/j.ast.2015.01.021 (cited on p. 45)

Yan, C. C. S., Chepyala, S. R., Yen, C. M., & Hsu, C. P. (2017). Efficient and flexible implementation of Langevin simulation for gene burst production [Publisher: Springer US]. *Scientific Reports*, *7*(1), 1–16. https://doi.org/10.1038/s41598-017-16835-y (cited on p. 143)

Yang, N., Smyllie, N. J., Morris, H., Gonçalves, C. F., Dudek, M., Pathiranage, D., Chesham, J. E., Adamson, A., Spiller, D., Zindy, E., Bagnall, J., Humphreys, N., Hoyland, J., Loudon, A. S., Hastings, M. H., & Meng, Q. J. (2020). Quantitative live imaging of Venus::BMAL1 in a mouse model reveals complex dynamics of the master circadian clock regulator. *PLoS Genetics*, *16*(4), 1–24. https://doi.org/10.1371/journal.pgen.1008729 (cited on p. 55)

Yao, Y., Vehtari, A., Simpson, D., & Gelman, A. (2018). Yes, but Did It Work?: Evaluating Variational Inference [arXiv:1802.02538 [stat]]. https://doi.org/10.48550/arXiv.1802.02538. (Cited on p. 104)

Yates, A. D., Achuthan, P., Akanni, W., Allen, J., Allen, J., Alvarez-Jarreta, J., Amode, M. R., Armean, I. M., Azov, A. G., Bennett, R., Bhai, J., Billis, K., Boddu, S., Marugán, J. C., Cummins, C., Davidson, C., Dodiya, K., Fatima, R., Gall, A., … Flicek, P. (2020). Ensembl 2020. *Nucleic Acids Research*, *48*(D1), D682–D688. https://doi.org/10.1093/nar/gkz966 (cited on p. 113)

Zechner, C., Ruess, J., Krenn, P., Pelet, S., Peter, M., Lygeros, J., & Koeppl, H. (2012). Moment-based inference predicts bimodality in transient gene expression [Publisher: Proceedings of the National Academy of Sciences]. *Proceedings of the National Academy of Sciences*, *109*(21), 8340–8345. https://doi.org/10.1073/pnas.1200161109 (cited on p. 43)

Zhang, L., Carpenter, B., Gelman, A., & Vehtari, A. (2022). Pathfinder: Parallel quasi-Newton variational inference [arXiv:2108.03782 [cs, stat]]. Retrieved January 10, 2023, from http://arxiv.org/abs/2108.03782. (Cited on pp. 87, 93, 104)

Zhao, C., Sun, G., Li, S., & Shi, Y. (2009). A feedback regulatory loop involving microRNA-9 and nuclear receptor TLX in neural stem cell fate determination [Number: 4 Publisher: Nature Publishing Group]. *Nature Structural & Molecular Biology*, *16*(4), 365–371. https://doi.org/10.1038/nsmb.1576 (cited on p. 110)

Zhuang, K., & Zhu, H. (2010). Stability and bifurcation analysis in a model of Hes1 selfregulation with time delay. *World Academy of Science, Engineering and Technology*, *43*(7), 469–471 (cited on p. 71).

ZivariPiran, H., & Enright, W. H. (2010). An efficient unified approach for the numerical solution of delay differential equations. *Numerical Algorithms*, *53*(2), 397–417. https://doi.org/10.1007/s11075-009-9331-y (cited on p. 92)

Appendices

Appendix A

Inferring kinetic parameters of oscillatory gene regulation from single cell time-series data supplementary

Here we provide full derivations for the prediction and update step in the non-linear, delay-adapted Kalman filter. We detail the MCMC sampling algorithms discussed in the main text, as well as our chosen method for adaptive step size selection within the MCMC. Then, we show how the gradient of the likelihood can be computed within the Kalman filter. We further describe our process for choosing prior distributions, and for assessing the convergence of our Markov chains, before discussing how the oscillation quality of time series data is quantified, and how the likelihood function is extended to account for data on mRNA copy numbers. Finally, we give details about the data collection and analysis. Supplementary figures are provided at the end.

A.1 Kalman filter prediction and update step

In order to predict the probability distribution over the state space $\mathbf{X}$ at observation k, it is necessary to calculate ρ_t and P_t as defined in the main manuscript (Section 3.4.2). Here, we show how we derive delay differential equations for both quantities, closely following Calderazzo et al. (2019).

A.1.1 Prediction Step

We first rewrite eqs. (3.2) and (3.3) in the general form of a vector-valued, delayed stochastic differential equation:

$$\begin{aligned} d\mathbf{X}(t) = g(\mathbf{X}(t))dt + f\left(\mathbf{X}(t-\tau)\right)dt \\ + \sqrt{l(\mathbf{X}(t)) + q\left(\mathbf{X}(t-\tau)\right)}dB(t), \end{aligned} \tag{A.1}$$

where $B(t)$ is a 2-dimensional Wiener process, and

$$\begin{aligned} g(\mathbf{X}(t)) &= \begin{bmatrix} -\mu_m m(t) \\ \alpha_p m(t) - \mu_p p(t) \end{bmatrix}, \\ f(\mathbf{X}(t-\tau)) &= \begin{bmatrix} \alpha_m G(p(t-\tau)) \\ 0 \end{bmatrix}, \\ l(\mathbf{X}(t)) &= \begin{bmatrix} \mu_m m(t) & 0 \\ 0 & \alpha_p m(t) + \mu_p p(t) \end{bmatrix}, \\ q(\mathbf{X}(t-\tau)) &= \begin{bmatrix} \alpha_m G(p(t-\tau)) & 0 \\ 0 & 0 \end{bmatrix}. \end{aligned}$$

Next, we aim to use this definition to derive delay differential equations for ρ and P. To do so, we employ Taylor expansions. The first order Taylor expansions of $g(\cdot)$ and $f(\cdot)$ about ρ_t are

$$g(X_t) \approx g(\rho_t) + J_g(\rho_t) \cdot (X_t - \rho_t),$$

and

$$f(X_{t-\tau}) \approx f(\rho_{t-\tau}) + J_f(\rho_{t-\tau}) \cdot (X_{t-\tau} - \rho_{t-\tau}).$$

We can hence write

$$X_{t+\delta_t} = X_t + g(X_t)\delta_t + f(X_{t-\tau})\delta_t + \sqrt{l(X_t) + q(X_{t-\tau})}\mathcal{N}(0, \delta_t) + \mathcal{O}(\delta_t^2), \tag{A.2}$$

where $\mathcal{N}(0, \delta_t)$ is a Gaussian distribution with mean zero and variance δ_t. Inserting this into the definition of ρ_t, we find

$$d\rho_t = g(\rho_t)dt + f(\rho_{t-\tau})dt. \tag{A.3}$$

Similarly, we insert eq. (A.2) into the definition of P to find

$$dP_t = \left[J_g(\rho_t)P_t + P_t^T J_g(\rho_t)^T\right] dt$$

$$
\begin{aligned}
&+ \left[J_f(\rho_{t-\tau})P_{t-\tau,t}\right] dt \\
&+ \left[P_{t,t-\tau}J_f(\rho_{t-\tau})^T\right] dt \\
&+ A\left(\rho_t, \rho_{t-\tau}\right) dt,
\end{aligned}
\tag{A.4}
$$

where $A(\rho_t, \rho_{t-\tau}) = l(\rho_t) + q(\rho_{t-\tau})$ and $P_{t,s} = \mathrm{Cov}(\mathbf{X}(t), \mathbf{X}(s) \mid y_{0:t-1})$.

Lastly, we have

$$
dP_{s,t} = P_{s,t}J_g^T(\rho_t)dt + P_{s,t-\tau}J_f^T(\rho_{t-\tau})dt. \tag{A.5}
$$

We can then use a standard forward Euler method with a time step which represents the absolute time difference between consecutive hidden states, to numerically integrate these delay differential equations between observations. Before the first prediction step can be executed, it is necessary to define initial and past conditions for these delay differential equations. The state space mean is initialised at negative times with the steady-state solutions to the deterministic approximation of the model, i.e. eqs. (3.2) and (3.3) of the main manuscript without their respective noise terms. At steady state, we have

$$
p^* = \frac{\alpha_m \alpha_p}{\mu_m \mu_p} G(p^*) \tag{A.6}
$$

and

$$
m^* = \frac{\mu_p}{\alpha_p} p^*. \tag{A.7}
$$

The state space variance is initialised with a multiple of this mean. Specifically, the variance of the mRNA is initialised at 20 times its mean value, and the variance of the protein is initialised at 100 times its mean value. These are chosen to reflect the variance of protein and mRNA expression from the Chemical Langevin Equations. For simplicity, off-diagonal elements of the state space covariance are initialised at zero. We expect the influence of these initial and past conditions to diminish after the first update step has been applied and as the number of data points increases.

A.1.2 Update Step

For our update step, we derive an expression for the mean and variance of the state space distribution $\pi(x_{t-\tau:t} \mid y_{0:t})$, denoted $\rho^*_{t-\tau:t}$ and $P^*_{t-\tau:t}$ respectively. That is, the likelihood of our state space estimates from the past time $t - \tau$ to the current time, t, given all of our current observations. This is necessary in order to accurately predict the state space distribution at the next observation time point, $\pi(x_{t+\Delta t} \mid y_{0:t})$, as past states can affect future states due to the presence of delays.

Assuming approximate Gaussianity of the joint distribution $\pi(x_{t+\Delta t}, y_{t+\Delta t} \mid y_{0:t})$, following Särkkä (2013), we can obtain the following expression for the updated state space mean and variance:

$$\rho^*_{t-\tau:t} = \rho_{t-\tau:t} + K_{t-\tau:t}\left(y_t - F\rho_t\right), \tag{A.8}$$

$$P^*_{t-\tau:t} = P_{t-\tau:t} - K_{t-\tau:t}S_t K^T_{t-\tau:t}, \tag{A.9}$$

$$K_{t-\tau:t} = P_{t-\tau:t,t}F^T S_t^{-1}, \tag{A.10}$$

where $S_t = FP_tF^T + \Sigma_\epsilon$.

A.2 Markov Chain Monte Carlo sampling algorithms

For completeness, we provide here the algorithms for the Metropolis-Hasting random walk, and the Metropolis-adjusted Langevin Algorithm.

A.2.1 Metropolis-Hastings Random Walk

We initialise the algorithm with a set of the model parameters, $\boldsymbol{\theta}_0$, chosen by minimising the log-likelihood function using Powell's method (Powell, 1964). In the case of our model, each parameter vector is composed as $\boldsymbol{\theta} = [P_0, h, \mu_m, \mu_p, \alpha_m, \alpha_p, \tau]^T$. At each iteration, i, we propose a combination of parameters, $\boldsymbol{\theta}'$, from a symmetric distribution, $q(\cdot)$, centered around the current parameters, $\boldsymbol{\theta}_i$. In this paper we use the normal distribution $\mathcal{N}(\boldsymbol{\theta}, \epsilon\Sigma)$, where ϵ is called the step size, and Σ is the proposal covariance matrix. An acceptance ratio, $\alpha(\boldsymbol{\theta}_i, \boldsymbol{\theta}')$, defined by

$$\begin{aligned}\alpha(\boldsymbol{\theta}_i, \boldsymbol{\theta}') &= \min\left\{1, \frac{\pi(\boldsymbol{\theta}' \mid \mathbf{y})}{\pi(\boldsymbol{\theta}_m \mid \mathbf{y})}\right\} \\ &= \min\left\{1, \frac{\pi(\mathbf{y} \mid \boldsymbol{\theta}')\pi(\boldsymbol{\theta}')}{\pi(\mathbf{y} \mid \boldsymbol{\theta}_i)\pi(\boldsymbol{\theta}_m)}\right\}\end{aligned}$$

is calculated, and a uniform random number, u, is generated on $[0, 1]$.

1. If $u \leq \alpha(\boldsymbol{\theta}_i, \boldsymbol{\theta}')$, we *accept* the proposal $\boldsymbol{\theta}'$ and set $\boldsymbol{\theta}' = \boldsymbol{\theta}_{i+1}$.
2. If $u \geq \alpha(\boldsymbol{\theta}_i, \boldsymbol{\theta}')$, we *reject* the proposal $\boldsymbol{\theta}'$ and set $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{i+1}$.

The distribution of random draws $\{\boldsymbol{\theta}_i \mid i \in \{0, 1, \ldots, N_s\}\}$ converges to our posterior distribution, $\pi(\boldsymbol{\theta} \mid \mathbf{y})$, as $N_s \to \infty$. In practice, the number of iterations necessary for accurate inference depends on a number of factors, including the complexity of the problem, the number of parameters to be inferred, and the relationship between the posterior and proposal distributions. We outline our choice of N_s in Section A.3

A.2.2 Metropolis-adjusted Langevin Algorithm (MALA)

Whilst the MH algorithm is relatively simple to understand and implement, it can fail to identify the posterior if it has a non-trivial, curved correlation structure (Haario et al., 2001), or if it is multi-modal (Roberts et al., 2003). It can also become slow when the dimensionality is high, i.e. when there are a large number of parameters to infer. In these cases, the algorithm typically requires a very large number of samples, which is time-consuming, especially if the likelihood calculation is slow. One way to address this is to use a sampling algorithm that uses the gradient of the posterior distribution, as well as its absolute values, to draw new samples. The gradient can help guide the search towards the modes of the posterior distribution, making for a more efficient algorithm which typically needs fewer samples. A number of different MCMC approaches utilise the gradient of the distribution, such as MALA, and the NO-U-Turn sampler (M. D. Hoffman & Gelman, 2014), which is an extension of the original Hamiltonian Monte Carlo (HMC) (Betancourt, 2017).

We use MALA, since it is well-established that it has better convergence properties than the MH algorithm (Durmus et al., 2017). Our algorithm for MALA iterates over the same steps as the MH algorithm, with an adjusted expression for the proposed parameter combination. Where MH uses the proposal

$$\boldsymbol{\theta}' = \boldsymbol{\theta} + \sqrt{\epsilon}\Sigma^{1/2}\xi, \tag{A.11}$$

MALA instead uses

$$\boldsymbol{\theta}' = \boldsymbol{\theta} - (\epsilon/2)\Sigma U(\boldsymbol{\theta}) + \sqrt{\epsilon}\Sigma^{1/2}\xi, \tag{A.12}$$

where $\xi \sim \phi(0, I_d)$, with I_d denoting the d-dimensional identity matrix, d is the dimension of the posterior distribution, and $U(\boldsymbol{\theta}) = -\nabla \log \pi(\mathbf{y} \mid \boldsymbol{\theta})$.

A.3 Implementation of adaptive sampling strategies

We seek to generate N_s independent samples that accurately characterise our posterior distribution, but samples are dependent on the starting point of the Markov chain. As increasingly many samples are generated, this dependence diminishes. The correlation between samples can be reduced if an appropriate covariance matrix is chosen for the proposal distribution. One suitable covariance matrix is the covariance of the target posterior distribution, which may be estimated from samples of the MCMC itself. To do so, we *warm up* the Markov chains in two steps by drawing N_{w_1} samples and discarding the first N_{b_1} samples as *burn-in*. With the remaining samples we construct a proposal covariance matrix, and repeat the process with N_{w_2} samples and a burn-in of size N_{b_2}. The adaptive MCMC is then started using the covariance matrix estimated in this way as the proposal covariance matrix. We adaptively update our step size at iteration i, ϵ_i, using a modified version of Algorithm 4 from Andrieu and Thoms (2008).

At each iteration i, the step size is updated using

$$\begin{aligned} \gamma_1 &= \frac{1}{i^{c_1}}, \\ \gamma_2 &= c_0\gamma_1, \\ \log(\epsilon_{i+1}^2) &= \log(\epsilon_i^2) + \gamma_2(\alpha_i - 0.574), \end{aligned} \tag{A.13}$$

where $c_0 = 1$, $c_1 = \log(10)/\log(N_s/5)$, and α_i is the current acceptance probability. We use 0.574 as the target acceptance rate for MALA, as shown above, and 0.234 for MH (Roberts & Rosenthal, 2001).

In our experience, this approach results in far better performance than without warm up or adaptively updating the step size. In practice, we found that consistent posteriors are obtained by running 8 Markov chains in parallel, where each chain has $N_s = 80,000$, $N_{w_1} = 0.3 \times N_s$, $N_{w_2} = 0.7 \times N_s$, and $N_{b_i} = 0.5 \times N_{w_i}$, $i = 1, 2$. Warm up samples from all eight chains are pooled to calculate the covariance matrix for the final run of MCMC. This approach to warm up is used in the same way when either sampler, MH or MALA, is chosen.

A.4 Computing the gradient of the log-posterior

The Kalman filter has previously been used within gradient based sampling schemes (Mbalawata et al., 2013), which illustrated that it is possible to calculate the gradient with an iterative scheme similar to the scheme for the direct likelihood calculation. In the following, we show that these previous approaches can be extended to derive the gradient of the delay-mediated Kalman filter used here.

For MALA, we compute the derivative of the negative log-posterior $U = -\log(\pi(\boldsymbol{\theta}|\mathbf{y})$ with respect to the parameters, $\boldsymbol{\theta}$,

$$\frac{\partial U(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{\partial(-\log(\pi(\mathbf{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})))}{\partial \boldsymbol{\theta}}$$

$$
\begin{aligned}
&= \frac{\partial\Big(-\log\Big(\prod_{t\in\mathcal{T}}\phi(y_t;F\rho_t,FP_tF^T+\Sigma_\epsilon)\pi(\boldsymbol{\theta})\Big)\Big)}{\partial\boldsymbol{\theta}}\\
&= -\frac{\partial}{\partial\boldsymbol{\theta}}\log\left(\prod_{t\in\mathcal{T}}\frac{\exp\left(-\frac{1}{2}(y_t-F\rho_t)^T(FP_tF^T+\Sigma_\epsilon)^{-1}(y_t-F\rho_t)\right)\pi(\boldsymbol{\theta})}{\sqrt{\det(2\pi(FP_tF^T+\Sigma_\epsilon))}}\right)\\
&= \frac{\partial}{\partial\boldsymbol{\theta}}\left[\sum_{t\in\mathcal{T}}\frac{1}{2}\log\left(\det(2\pi S_t(\boldsymbol{\theta}))\right)+\right.\\
&\qquad\left.\frac{1}{2}\sum_{t\in\mathcal{T}}(y_t-F\rho_t(\boldsymbol{\theta}))^T(S_t(\boldsymbol{\theta}))^{-1}(y_t-F\rho_t(\boldsymbol{\theta}))-\log(\pi(\boldsymbol{\theta}))\right],
\end{aligned}
$$

where $S = FP_tF^T + \Sigma_\epsilon$ has been used in the last line, and the dependence of ρ and S on $\boldsymbol{\theta}$ has been emphasized. $\mathcal{T}$ denotes the set of time points for each observation, $\{i\cdot l \mid i\in\{0,1,\ldots,n-1\}\}$. For a single parameter, this simplifies to

$$
\begin{aligned}
\frac{\partial U(\boldsymbol{\theta})}{\partial\theta_k} &= \frac{1}{2}\sum_{t\in\mathcal{T}}\left(\mathrm{Tr}\left(S_t^{-1}\frac{\partial S_t}{\partial\theta_k}\right)\right)-\frac{1}{2}\sum_{t\in\mathcal{T}}\left(F\frac{\partial\rho_t}{\partial\theta_k}\right)^T S_t^{-1}(y_t-F\rho_t)\\
&\quad-\frac{1}{2}\sum_{t\in\mathcal{T}}(y_t-F\rho_t)^TS_t^{-1}\frac{\partial S_t}{\partial\theta_k}S_t^{-1}(y_t-F\rho_t)\\
&\quad-\frac{1}{2}\sum_{t\in\mathcal{T}}(y_t-F\rho_t)^TS_t^{-1}F\frac{\partial\rho_t}{\partial\theta_k}-\frac{\partial(\log(\pi(\boldsymbol{\theta})))}{\partial\theta_k}. \qquad (A.14)
\end{aligned}
$$

This expression can be explicitly calculated if $\mathrm{d}\rho_t/\mathrm{d}\theta_k$ and $\mathrm{d}P_t/\mathrm{d}\theta_k$ are known at each observation time point t. These quantities are the predicted state space mean and variance in the prediction step of the Kalman filter. This implies that eq. (A.14) can be calculated if the prediction step is adjusted to predict the derivative of ρ and P in addition to their absolute values. To enable this adjustment of the prediction step, we derive differential equations for $\frac{\mathrm{d}}{\mathrm{d}t}(\mathrm{d}\rho_t/\mathrm{d}\theta_k)$ and $\frac{\mathrm{d}}{\mathrm{d}t}(\mathrm{d}P_t/\mathrm{d}\theta_k)$:

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial\rho_t}{\partial\theta_k}\right) &= \frac{\partial}{\partial\theta_k}\left(\frac{\mathrm{d}\rho_t}{\mathrm{d}t}\right)\\
&= \frac{\partial}{\partial\theta_k}\left(g(\rho_t)+f(\rho_{t-\tau})\right)\\
&= \frac{\partial g(\rho_t)}{\partial\rho_t}\frac{\partial\rho_t}{\partial\theta_k}+\frac{\partial g(\rho_t)}{\partial\theta_k}+\frac{\partial f(\rho_{t-\tau})}{\partial\rho_{t-\tau}}\frac{\partial\rho_{t-\tau}}{\partial\theta_k}+\frac{\partial f(\rho_{t-\tau})}{\partial\theta_k}\\
&= J_g(\rho_t)\frac{\partial\rho_t}{\partial\theta_k}+\frac{\partial g(\rho_t)}{\partial\theta_k}+J_f(\rho_{t-\tau})\frac{\partial\rho_{t-\tau}}{\partial\theta_k}+\frac{\partial f(\rho_{t-\tau})}{\partial\theta_k}, \qquad (A.15)
\end{aligned}
$$

and

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial P_t}{\partial \theta_k}\right) &= \frac{\partial}{\partial \theta_k}\left(\frac{\mathrm{d}P_t}{\mathrm{d}t}\right) \\
&= \frac{\partial}{\partial \theta_k}\Big(J_g(\rho_t)P_t + P_t^T J_g(\rho_t)^T \\
&\qquad + J_f(\rho_{t-\tau})P_{t-\tau,t} + P_{t,t-\tau}J_f(\rho_{t-\tau})^T \\
&\qquad + l(\rho_t) + q(\rho_{t-\tau})\Big) \\
&= J_g(\rho_t)\frac{\partial P_t}{\partial \theta_k} + \frac{\partial P_t^T}{\partial \theta_k}J_g(\rho_t)^T + J_f(\rho_{t-\tau})\frac{\partial P_{t-\tau,t}}{\partial \theta_k} + \frac{\partial P_{t,t-\tau}}{\partial \theta_k}J_f(\rho_{t-\tau})^T \\
&+ \left(\sum_{j=1}^{2}\left(\frac{\partial J_g(\rho_{t_j})}{\partial \rho_{t_j}}\frac{\partial \rho_{t_j}}{\partial \theta_k}\right) + \frac{\partial J_g(\rho_t)}{\partial \theta_k}\right)P_t + P_t^T\left(\sum_{j=1}^{2}\left(\frac{\partial J_g(\rho_{t_j})}{\partial \rho_{t_j}}\frac{\partial \rho_{t_j}}{\partial \theta_k}\right) + \frac{\partial J_g(\rho_t)}{\partial \theta_k}\right)^T \\
&+ \left(\sum_{j=1}^{2}\left(\frac{\partial J_f(\rho_{t-\tau_j})}{\partial \rho_{t-\tau_j}}\frac{\partial \rho_{t-\tau_j}}{\partial \theta_k}\right) + \frac{\partial J_f(\rho_{t-\tau})}{\partial \theta_k}\right)P_{t-\tau,t} \\
&+ P_{t,t-\tau}\left(\sum_{j=1}^{2}\left(\frac{\partial J_f(\rho_{t-\tau_j})}{\partial \rho_{t-\tau_j}}\frac{\partial \rho_{t-\tau_j}}{\partial \theta_k}\right) + \frac{\partial J_f(\rho_{t-\tau})}{\partial \theta_k}\right)^T \\
&+ \left(\sum_{j=1}^{2}\left(\frac{\partial l(\rho_{t_j})}{\partial \rho_{t_j}}\frac{\partial \rho_{t_j}}{\partial \theta_k}\right) + \frac{\partial l(\rho_t)}{\partial \theta_k}\right) + \left(\sum_{j=1}^{2}\left(\frac{\partial q(\rho_{t_j})}{\partial \rho_{t_j}}\frac{\partial \rho_{t_j}}{\partial \theta_k}\right) + \frac{\partial q(\rho_t)}{\partial \theta_k}\right) \quad \text{(A.16)}
\end{aligned}
$$

where ρ_{t_j} means the jth element of ρ_t. Similarly,

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial P_{s,t}}{\partial \theta_k}\right) &= \frac{\partial}{\partial \theta_k}\left(\frac{\mathrm{d}P_{s,t}}{\mathrm{d}t}\right) \\
&= \frac{\partial}{\partial \theta_k}\left(P_{s,t}J_g^T(\rho_t) + P_{s,t-\tau}J_f^T(\rho_{t-\tau})\right) \\
&= \frac{\partial P_{s,t}}{\partial \theta_k}J_g^T(\rho_t) + \frac{\partial P_{s,t-\tau}}{\partial \theta_k}J_f^T(\rho_{t-\tau}) \\
&+ P_{s,t}\left(\sum_{j=1}^{2}\left(\frac{\partial J_g(\rho_{t_j})}{\partial \rho_{t_j}}\frac{\partial \rho_{t_j}}{\partial \theta_k}\right) + \frac{\partial J_g(\rho_t)}{\partial \theta_k}\right)^T \\
&+ P_{s,t-\tau}\left(\sum_{j=1}^{2}\left(\frac{\partial J_f(\rho_{t-\tau_j})}{\partial \rho_{t-\tau_j}}\frac{\partial \rho_{t-\tau_j}}{\partial \theta_k}\right) + \frac{\partial J_f(\rho_{t-\tau})}{\partial \theta_k}\right)^T. \qquad \text{(A.17)}
\end{aligned}
$$

Similar to the delay differential equations for the state space mean and variance, it is necessary to define values for non-positive times for the delay differential equations provided here before the first prediction step can be executed. To do so, we need to provide derivatives of the initialisation of the state space mean and variance provided in Section A.1. We calculate the derivatives of p^* and m^* with respect to each parameter, θ_k, as follows:

$$\frac{\partial p^*}{\partial \theta_k} = \frac{-\frac{\partial}{\partial \theta_k}\left(\frac{\alpha_m \alpha_p}{\mu_m \mu_p}\right) G(p^*) - \left(\frac{\alpha_m \alpha_p}{\mu_m \mu_p}\right) \frac{\partial}{\partial \theta_k}\left(G(p^*)\right)}{\left(\frac{\alpha_m \alpha_p}{\mu_m \mu_p}\right) \frac{\partial}{\partial p^*} G(p^*) - 1}, \tag{A.18}$$

$$\frac{\partial m^*}{\partial \theta_k} = \frac{\partial}{\partial \theta_k}\left(\frac{\mu_p}{\alpha_p} p^*\right). \tag{A.19}$$

A.5 Update step derivatives

The delay differential equations given by eqs. (A.15) to (A.17) can be used to predict the derivatives of the state space mean and variance with respect to each model parameter, using $\mathrm{d}\rho_t^*/\mathrm{d}\theta_k$ and $\mathrm{d}P_t^*/\mathrm{d}\theta$ as initial (and past) conditions. We incorporate the integration of these ODEs into the prediction step of our Kalman filter. The derivatives of $\mathrm{d}\rho_t^*/\mathrm{d}\theta_k$ and $\mathrm{d}P_t^*/\mathrm{d}\theta$ at each observation can be calculated by extending the update step. Using eqs. (A.8) to (A.10), we have

$$\begin{aligned}\frac{\partial \rho^*_{t-\tau:t}}{\partial \theta_k} &= \frac{\partial}{\partial \theta_k}\left(\rho_{t-\tau:t} + K_{t-\tau:t}\left(y_t - F\rho_t\right)\right) \\ &= \frac{\partial \rho_{t-\tau:t}}{\partial \theta_k} + \frac{\partial K_{t-\tau:t}}{\partial \theta_k}\left(y_t - F\rho_t\right) - K_{t-\tau:t} F \frac{\partial \rho_t}{\partial \theta_k},\end{aligned} \tag{A.20}$$

$$\begin{aligned}\frac{\partial P^*_{t-\tau:t}}{\partial \theta_k} &= \frac{\partial}{\partial \theta_k}\left(P_{t-\tau:t} - K_{t-\tau:t} S_t K^T_{t-\tau:t}\right) \\ &= \frac{\partial P_{t-\tau:t}}{\partial \theta_k} - \frac{\partial K_{t-\tau:t}}{\partial \theta_k} S_t K^T_{t-\tau:t} - K_{t-\tau:t} \frac{\partial S_t}{\partial \theta_k} K^T_{t-\tau:t} - K_{t-\tau:t} S_t \left(\frac{\partial K_{t-\tau:t}}{\partial \theta_k}\right)^T,\end{aligned} \tag{A.21}$$

$$\begin{aligned}\frac{\partial K_{t-\tau:t}}{\partial \theta_k} &= \frac{\partial}{\partial \theta_k}\left(P_{t-\tau:t,t} F^T S_t^{-1}\right) \\ &= \frac{\partial P_{t-\tau:t,t}}{\partial \theta_k} F^T S_t^{-1} - P_{t-\tau:t,t} F^T S_t^{-1} \frac{\partial S_t}{\partial \theta_k} S_t^{-1} \\ &= \frac{\partial P_{t-\tau:t,t}}{\partial \theta_k} F^T S_t^{-1} - P_{t-\tau:t,t} F^T S_t^{-1} F \frac{\partial P_t}{\partial \theta_k} F^T S_t^{-1}.\end{aligned} \tag{A.22}$$

Here, $\frac{\partial \rho_{t-\tau:t}}{\partial \theta_k}$ and $\frac{\partial P_{t-\tau:t}}{\partial \theta_k}$ are the derivatives for the state space mean and variance that have been predicted in the previous prediction step. Hence, we can evaluate eq. (A.14) for each of the parameters iteratively by evaluating eqs. (A.15) to (A.17), (A.20) and (A.21) at each update step, and integrating eqs. (A.3) to (A.5) within each prediction step of the Kalman filter.

A.6 Choice of priors and reparameterisation of variables

Bayesian methods require the choice of prior parameter distributions. For the mRNA and protein degradation rates, μ_m and μ_p, we use fixed values throughout, which are based on previously identified experimental measurements (Manning et al., 2019). This facilitates our analysis of experimental time course data from the same paper in Figure 3.4. For the transcription and translation rates, α_m and α_p, we use prior distributions that are log-uniform. For all other parameters, we choose uniform prior distributions. Our chosen parameter ranges are informed by existing literature values (M. Suter et al., 2011; Manning et al., 2019; Monk, 2003; Phillips et al., 2016; Schwanhüusser et al., 2011; J. Singh & Padgett, 2009), and summarised in Table A.1.

Parameter	Range of Values
Repression threshold, P_0	0 - 120,000 (Monk, 2003)
Hill Coefficient, h	2 - 6 (Monk, 2003; Phillips et al., 2016)
Protein degradation rate, μ_p	log(2)/(90 min) (Manning et al., 2019)
mRNA degradation rate, μ_m	log(2)/(30 min) (Manning et al., 2019)
Basal transcription rate, α_m	0.01/min - 60/min (M. Suter et al., 2011; J. Singh & Padgett, 2009)
Translation rate, α_p	1/min - 40/min (Schwanhüusser et al., 2011)
Transcriptional delay, τ	5 min - 40 min (Monk, 2003; Phillips et al., 2016)

Table A.1: The values each of the parameters of the model can take, in relation to the HES5 system, informed by experimental work and biophysical limitations. These values define the prior distributions which we use in our parameter inference algorithms.

The choice of log-uniform priors for the transcription and translation rates, α_m and α_p, is motivated by the fact that possible values for these parameters span multiple orders of magnitude, from 0.01 – 120/min and 0.01 – 40/min respectively. To enable sampling from these log-uniform priors, we convert the parameters into logarithmic space using

$$\tilde{\theta} = \ln(\theta),$$

where θ represents the parameter that is being transformed into logarithmic space, i.e. α_m or α_p.

Let $P_1(\theta \mid \mathbf{y})$ and $P_2(\tilde{\theta} \mid \mathbf{y})$ define the posterior distributions for the original and transformed parameter respectively. A transformation of variables changes the shape of the posterior distribution according to

$$P_2(\tilde{\theta} \mid \mathbf{y})\mathrm{d}\tilde{\theta} = P_1(\theta \mid \mathbf{y})\mathrm{d}\theta$$

which in turn gives

$$\begin{aligned} P_2(\tilde{\theta} \mid \mathbf{y}) &= \theta P_1(\theta \mid \mathbf{y}) \\ &= e^{\tilde{\theta}} P_1(e^{\tilde{\theta}} \mid \mathbf{y}), \end{aligned} \tag{A.23}$$

since $\dfrac{\mathrm{d}\tilde{\theta}}{\mathrm{d}\theta} = \dfrac{1}{\theta}$.

Using $P_1(\theta \mid \mathbf{y}) = L(\mathbf{y} \mid \theta)\pi_1(\theta)$ and considering a prior that is uniform in logarithmic space, i.e. $\pi_1(\theta) = C/\theta$ with an appropriately chosen constant C, eq. (A.23) leads to

$$\begin{aligned} \ln(P_2(\tilde{\theta} \mid \mathbf{y})) &= \tilde{\theta} + \ln(L(\mathbf{y} \mid e^{\tilde{\theta}})) + \ln\left(\frac{C}{\theta}\right) \\ &= \ln(L(\mathbf{y} \mid e^{\tilde{\theta}})) + \ln(C) \\ &= \ln(L(\mathbf{y} \mid \theta)) + \ln(C). \end{aligned} \tag{A.24}$$

Lastly, taking the derivative gives

$$\frac{\partial}{\partial\tilde{\theta}} \ln(P_2(\tilde{\theta} \mid \mathbf{y})) = \theta \frac{\partial}{\partial\theta} \ln(L(\mathbf{y} \mid \theta)). \tag{A.25}$$

A.7 MCMC success is qualified by multiple convergence diagnostics

When using MCMC samplers such as MH and MALA, it is necessary to check for convergence of the sampled distribution and to identify whether a sufficient number of samples has been generated. A combination of convergence diagnostics gives confidence that the obtained samples are useful and represent the posterior accurately. Specifically, we use the split-$\hat{R}$ and effective sample size (ESS) methods (Gelman et al., 1992; Vehtari et al., 2020). We follow the guidance of Vehtari et. al. (Vehtari et al., 2020), running multiple chains ($m > 4$), and conclude convergence when split-$\hat{R} < 1.01$ and the total ESS is at least $50 \times 2 \times m$.

When testing our algorithm on *in silico* data, we use test diagnostics to determine inference uncertainty and accuracy. Specifically, for a given set of *in silico* data, let μ_{θ_i} define the true value of each parameter that generated the data set, and let $\hat{\mu}_{\theta_i}, \hat{\sigma}_{\theta_i}$ define the posterior mean and standard deviation, respectively, for each parameter. Throughout, we discuss the relative uncertainty (RU)

$$\mathrm{RU}_{\boldsymbol{\theta}} = \sum_{\theta_i} \frac{\hat{\sigma}_i}{|\mathrm{supp}\{\pi(\theta_i)\}|}, \tag{A.26}$$

where $|\text{supp}\{\pi(\theta_i)\}|$ is the size of the support of the prior for parameter θ_i. The relative uncertainty hence measures how the marginal standard deviations of the posterior distribution compare to their respective prior widths. Similarly, we introduce the mean error (ME) as

$$\text{ME}_{\boldsymbol{\theta}} = \sum_{\theta_i} \frac{|\mu_{\theta_i} - \hat{\mu}_{\theta_i}|}{|\text{supp}\{\pi(\theta_i)\}|}, \tag{A.27}$$

which measures how strongly the inferred posterior mean differs from the ground truth. For any parameters that are sampled in logarithmic space (supplementary Section A.6 above), we consider the samples and prior distributions in the original, *non*-transformed, space.

A.8 Oscillation quality is quantified by the coherence measure

In the presence of noise, time series can exhibit oscillations of varying quality. Here we introduce a measure to quantify the oscillatory properties of a time series, the coherence. We define coherence as the area under the power spectrum, $f(\omega)$, within a 20% band around the peak frequency, divided by the total area under the curve, following previous approaches (Alonso et al., 2007; Phillips et al., 2016). The power spectrum is defined by

$$f(\omega) = \langle \hat{p}(w)\hat{p}^*(w) \rangle,$$

where $\hat{p}(\omega)$ is the Fourier transform of the individual protein expression time course data, and * denotes complex conjugation. Values of the coherence around zero suggest a lack of any periodic behaviour, and values around one suggest sine-wave-like behaviour. For *in silico* data we estimate coherence by averaging the power spectra from 200 traces, which individually were simulated for 8500 minutes. The initial 1000 minutes of the simulation were discarded, and not used in the averaging, to minimise any influence from the initial conditions. The use of the power spectrum to define coherence ensures that coherence is robust to changes in the sampling interval, provided that this interval is shorter than the dominant period of oscillation.

The parameter combinations that correspond to different coherence values in Figure 3.5 are listed in supplementary Table A.2.

Coherence value	P_0	h	α_m	α_p	τ
0.00599 (low)	88288.602	5.589	0.644	17.321	34.0
0.200 (high)	30108.800	4.950	20.006	20.540	13.0

Table A.2: **Parameter combinations for specific coherence values.** The first row corresponds to 'low' coherence, and the second to 'high' coherence in Figure 3.5. To simulate noisy measurements, each synthetic data set is generated using a measurement variance of $\Sigma_\epsilon = 10^6$.

A.9 Extension of the likelihood function to account for data on mRNA expression levels

Our method generates posterior distributions from single-cell time series data of gene expression dynamics, using the likelihood function introduced in Section 3.4.2. It may be desirable to extend this likelihood function to account for further types of data, if these are available, in order to further constrain the model parameters. Here, we briefly describe how our likelihood function can be extended if, in addition to measurements of protein expression over time, data on the population-level distribution of mRNA copy numbers is available. Such data may for example be collected via single-molecule fluorescence *in situ* hybridisation (smFISH) experiments (Femino et al., 1998; Raj et al., 2008).

Let us assume a population-level distribution of mRNA copy numbers has been observed, and that this distribution can be approximated by a Gaussian with mean $\hat{\mu}_m$ and variance $\hat{\sigma}_m^2$. We can use these data to define an extended likelihood function as

$$\pi(\mathbf{y}, \hat{\mu}_m, \hat{\sigma}_m^2 \mid \boldsymbol{\theta}) = \pi(\mathbf{y} \mid \boldsymbol{\theta}) \times \phi(\hat{m}; \hat{\mu}_m, \hat{\sigma}_m^2), \tag{A.28}$$

where $\hat{m}$ is the time-average of the mean inferred mRNA copy number from the Kalman filter (i.e. the first entry of ρ_t in Section 3.4.2 averaged over all discretisation time points), $\pi(\mathbf{y} \mid \boldsymbol{\theta})$ is defined in eq. (3.6), and $\phi(\cdot)$ is a Normal density function, defined in eq. (3.7).

To generate data from *in silico* smFISH experiments, we generate the distribution of observed mRNA copy numbers numerically by simulating data from the same parameter combination that is used to generate a given *in silico* protein expression time series. Specifically, we simulate 5000 traces of time course data with a duration of 6000 minutes each, using eqs. (3.2) and (3.3). The first 1000 minutes of each time series are discarded to avoid any influence of the initial and past conditions. The mean and standard deviation are then calculated from all remaining simulated mRNA values, considering all simulated time points in all time series.

Our extended likelihood function designed in this way penalises parameter combinations for which the mean inferred mRNA copy numbers are outside the experimentally observed range. Hence, we expect our method to be applicable even if the mean mRNA copy number varies between cells. If the experimental setup is able to precisely measure the mean mRNA copy number for a specific time series of protein expression, our population-level variance $\hat{\sigma}_m^2$ may instead be replaced with the variance of the measured mean, thus enabling a more restrictive inference.

A.10 Data collection and analysis

A.10.1 Imaging of Primary NS cells

To generate data in Figure 3.1A, primary NS cells were isolated from the dissected cortex of E12.5 Venus::HES5 embryos (Imayoshi et al., 2013) and cultured as previously described (Pollard, 2013). 200,000 cells up to passage 15 were plated on laminin (Sigma, UK) coated 35mm glass-bottom dishes (Greiner-Bio One) and imaged in NS proliferation media at 37°C in 5% CO2 using a Plan Fluor 40x 1.3NA oil objective on a Nikon A1-R inverted confocal microscope. A 25μm z-range was used to ensure cells were maintained in focus. Maximum projections in z direction were generated before manual cell tracking using Imaris spot function (Bitplane).

A.10.2 Conversion of Venus::HES5 intensity to molecule number

Here, we briefly describe how we translate time series of fluorescent intensity from Manning et al. (2019) into time series of molecule numbers. Throughout our analysis, we use time series data of fluorescent intensity that were published by Manning et al. (2019) and which have been corrected for (i) photobleaching and (ii) for weaker intensity signals at deeper z-positions of the imaged nuclei. A quantile-quantile plot was generated between the distribution of tracked Venus::HES5 intensities and the distribution of nuclear Venus::Hes5 concentrations across the tissue. The distribution of tracked Venus::Hes5 intensities was generated from cells in tissues with heterozygous Venus::Hes5 reporter. All recorded single cells at all time points were used. The distribution of nuclear Venus::HES5 concentrations was generated using fluorescence correlation spectroscopy (FCS) on cells in tissue slices from the same domain of the E10.5 spinal cord and with a homozygous Venus::Hes5 reporter. Data from multiple tissue slices and multiple experiments was used. The required FCS data was also published by Manning et al. (2019). Linear regression on the quantile-quantile plot was used to generate a calibration curve between Venus::HES5 intensity and Venus::HES5 concentration over the middle 90% of the range. The gradient of the line was used as a scaling factor and applied to the intensity values in the Venus::HES5 expression time-series to transform intensity into concentrations. Finally, the average nuclear volume from Manning et al. (2019) was taken into account to convert concentrations into nuclear molecule numbers.

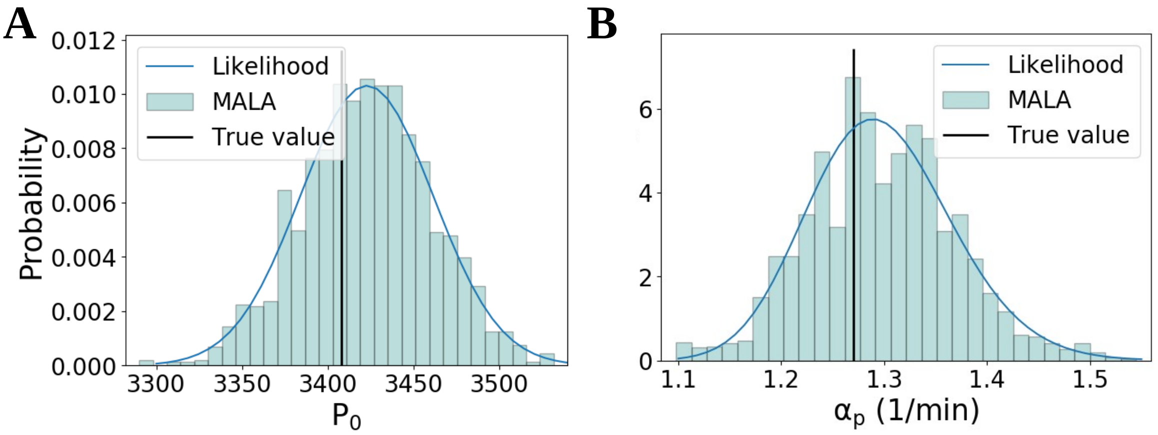


Figure A.1: **Our algorithm accurately samples posterior distributions. A, B.** Posterior distributions for one-dimensional inference. For individual model parameters, posterior distributions were inferred while keeping all other parameters fixed, respectively. Shown above are the inferred marginal posteriors for the repression threshold (A) and translation rate (B) respectively as histograms, using MALA as the underlying sampling algorithm for 2500 samples. The blue lines are the analytical likelihood calculations. The sampled and analytical distributions coincide. Other parameters are shown in Figure 3.2.

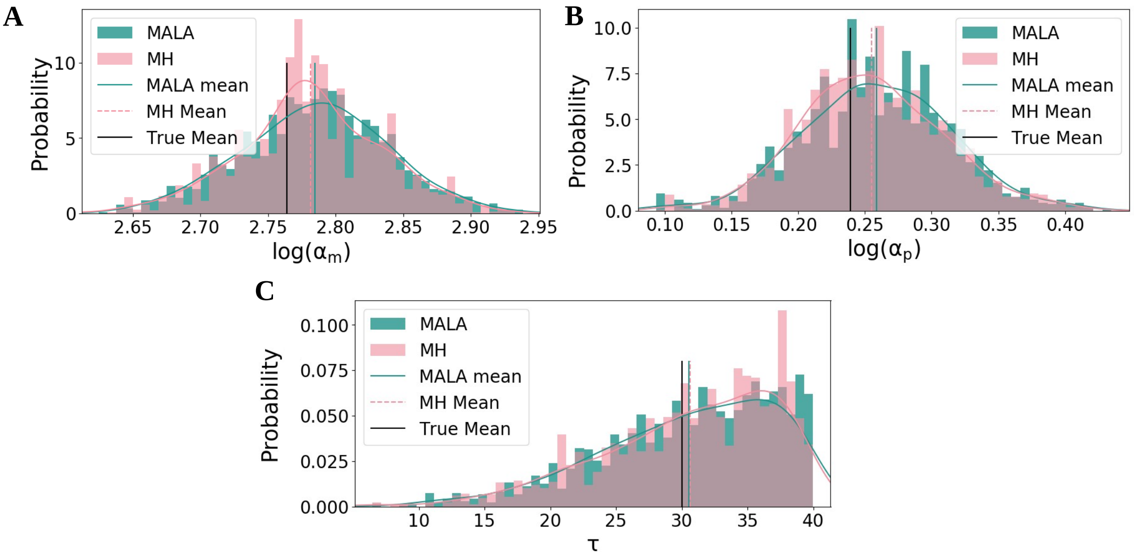


Figure A.2: **Our algorithm accurately samples posterior distributions. A, B, C.** Histograms for both MALA and MH on the 1-dimensional problem for the transcription rate (A), translation rate (B) and transcriptional delay (C). Histograms and kernel density estimates are plotted, alongside the mean from each chain and the ground truth value. Other parameters are shown in Figure 3.2.

Parameter	**True Value**	μ **(MALA)**	μ **(MH)**	σ **(MALA)**	σ **(MH)**
log Transcription rate, $\log(\alpha_m)$	2.764	2.784	2.782	0.054	0.052
log Translation rate, $\log(\alpha_p)$	0.239	0.259	0.255	0.055	0.053
Transcriptional delay, τ	30.0	30.546	30.652	6.653	6.578

Table A.3: The true values for the parameters which were used to generate the data in Figure 3.3A, alongside the means, μ, and standard deviations, σ of the corresponding one-dimensional posterior distributions, from both the MALA and MH algorithms (Figure A.2).

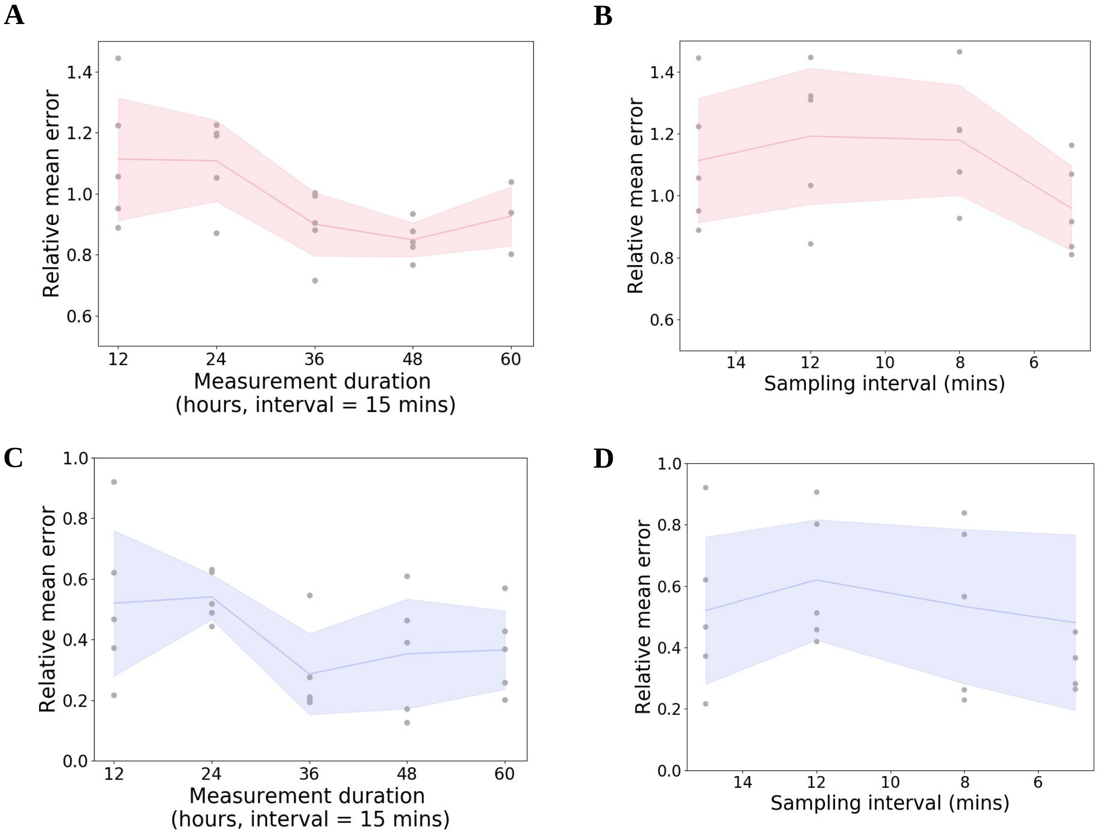


Figure A.3: **Increasing the length of time course data improves inference more than increased sampling frequency. A.** $\mathrm{ME}_{\boldsymbol{\theta}}$ for low coherence data sets sampled with different lengths, from 12 hours to 60 hours. **B.** $\mathrm{ME}_{\boldsymbol{\theta}}$ for low coherence data sets sampled with different intervals. **C** Same as A with the high coherence data sets. **D.** Same as B with the high coherence data sets.

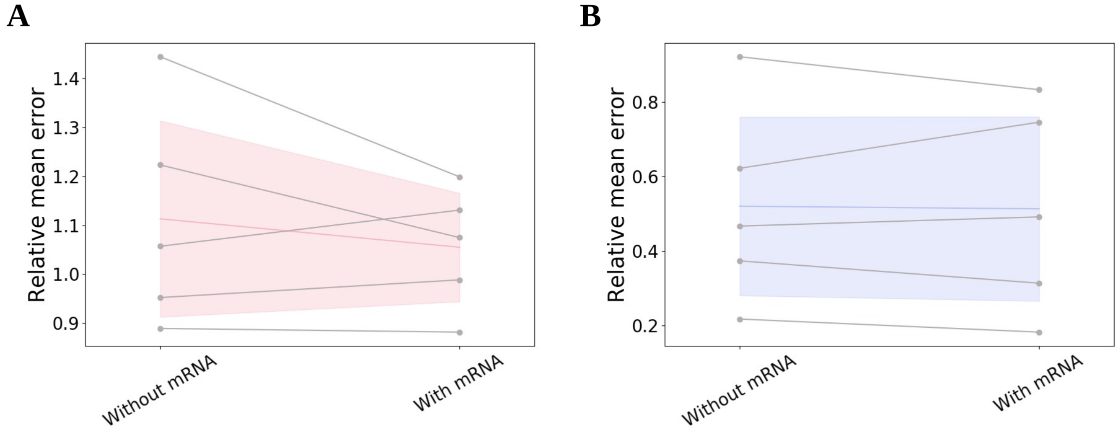


Figure A.4: **Additional measurements of mRNA copy numbers do not improve overall inference accuracy.** **A.** Values of $\mathrm{ME}_{\boldsymbol{\theta}}$ (see equation (A.27)) for low coherence data sets from Figure 3.5 with and without additional data on mRNA copy numbers (see supplementary Section A.9 for details). **B.** $\mathrm{ME}_{\boldsymbol{\theta}}$ for high coherence data sets from Figure 3.5 with and without additional data on mRNA copy numbers. The coloured lines and shaded areas represent the mean and standard deviation across observed values of $\mathrm{ME}_{\boldsymbol{\theta}}$, respectively.

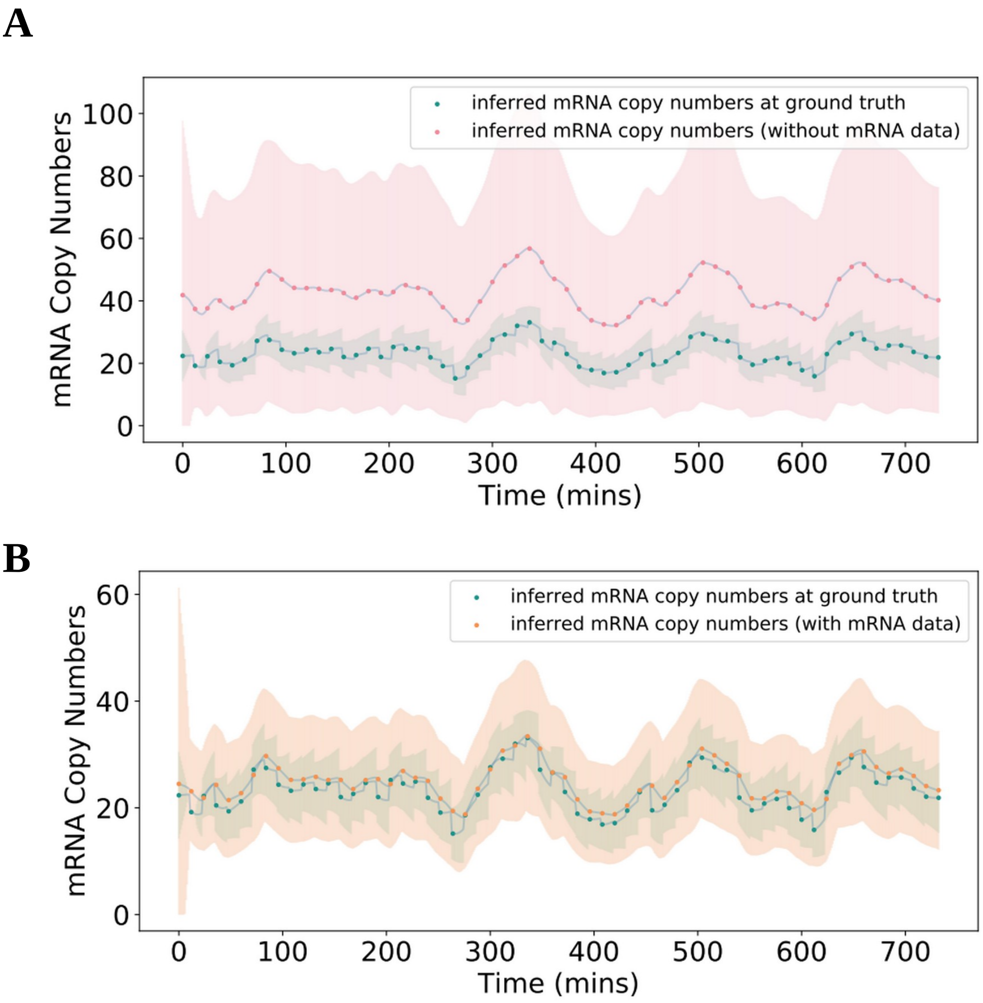


Figure A.5: **Inclusion of data on mRNA copy numbers reduces uncertainty on estimated mRNA time series.**

Figure A.5: **Inclusion of data on mRNA copy numbers reduces uncertainty on estimated mRNA time series. A.** The distribution of posterior inferred mRNA copy numbers from the Kalman filter is shown (pink line: mean, pink shaded area: 95% confidence interval). The mean was calculated using mean inferred mRNA copy number across 1000 randomly chosen posterior samples at each time point, and the variance was calculated from the same samples using the law of total variance, thus accounting for uncertainty on the model parameters and for uncertainty on mRNA copy numbers at each posterior parameter combination. Additionally, we show the inferred mRNA copy numbers if only the ground truth parameter combination is considered in the Kalman filter (green line: inferred mean, green shaded area: 95% confidence interval). The posterior mean estimates are far from the ground truth levels, with the confidence interval covering values up to three times larger than the ground truth mean. Ground truth parameters are as in the high coherence parameter set of Table A.2. **B.** If we include mRNA distribution information from an *in silico* smFISH experiment, the uncertainty on the inferred mRNA copy number is significantly reduced (orange dots: inferred mean, orange shading: 95% confidence interval), and also centered on the ground truth estimate (green dots and shading as in A).

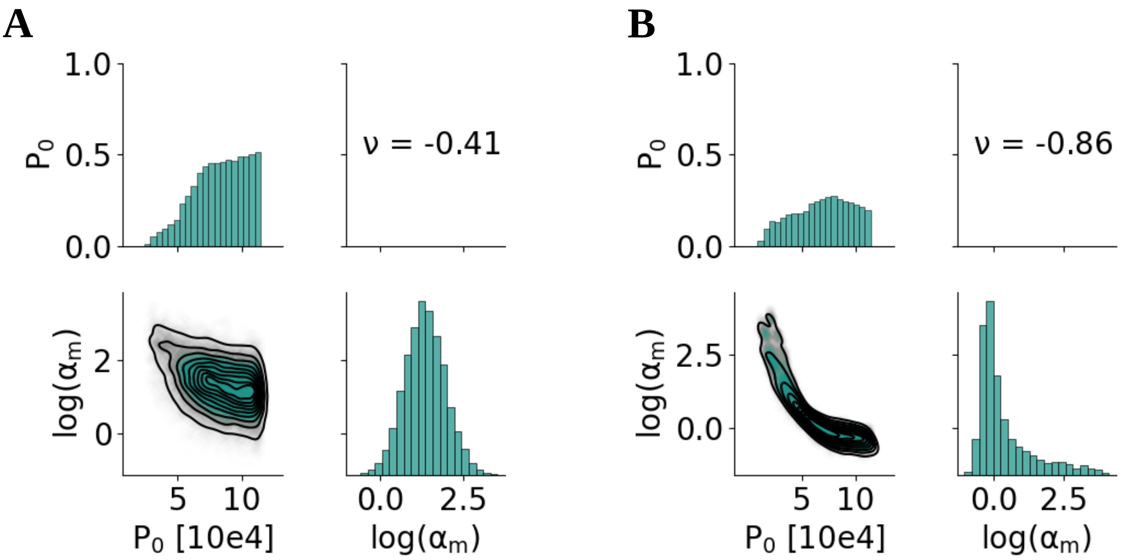


Figure A.6: **Additional data on mRNA copy numbers constrains relationships between parameters. A.** Joint posterior distributions for the P_0 and $\log(\alpha_m)$ parameters, corresponding to the data set analysed in Figure 3.6A. The parameters are not highly correlated. **B.** Posterior distribution on the same data as in A and with mRNA distribution information included in the inference method. The relationship between the parameters is now more strongly defined. However, the support of the individual marginal posteriors remains large. This illustrates that the relative uncertainty $\mathrm{RU}_{\boldsymbol{\theta}}$ may not be reduced despite improvements in the estimation of α_T (Figure 3.6D).

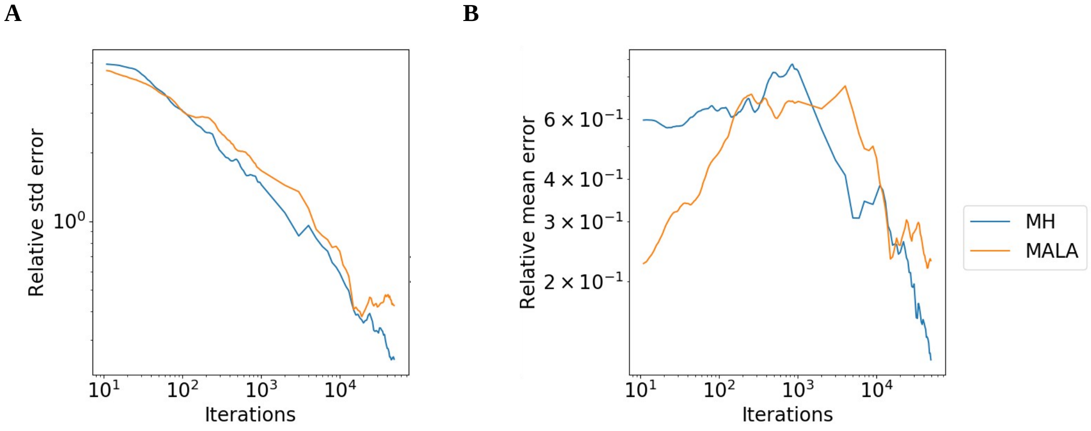


Figure A.7: **MH and MALA converge to the true posterior distribution at a similar rate.** Relative error in the **A** standard deviation and **B** mean of the posterior distribution as an increasing number of samples are drawn using MALA or MH, for an *in silico* data set of protein expression with a duration of 12 hours and with a sampling interval of 15 minutes, as well as assuming five unknown parameters. Note, that the data in Figures 3, 4 and 5 have the same structure. The *in silico* data set was generated using the parameter combination $P_0 = 47515, h = 4.77, \mu_m = \log(2)/30, \mu_p = \log(2)/90, \alpha_m = 2.65, \alpha_p = 17.61, \tau = 38.0, \Sigma_\epsilon = 10^6$. To calculate the error, a ground truth posterior distribution is generated by drawing 1,600,000 samples using MH. Errors are averaged over 8 chains for each sampler. For each chain, relative differences from the ground truth are calculated on each unknown parameter and then summed, to obtain an error estimate for the standard deviation and the mean. Before errors are calculated, chains are warmed up as described in supplementary Section A.3.

Appendix B

Continuous-time filtering and variational inference of combined single cell time-series data supplementary

B.1 MCMC

To perform MCMC sampling we used the Turing.jl Julia package (Ge et al., 2018). Specifically, we use the NUTS sampler (M. D. Hoffman & Gelman, 2014) with a target acceptance ratio of 0.8, and adaptation steps equal to half the desired number of samples per chain. For example, if we ran 4 chains to get 2000 samples from each, the number of adaptation steps would be 1000 per chain. The adaptation steps, often referred to as the burn-in phase, allow the sampler to learn the geometry of the posterior distribution, and find optimal tuning parameters for the HMC. These initial steps are discarded from the final sample. All other parameters were left as default.

For our prior distributions, we used a Normal$(\mu = 4, \sigma = 2)$ truncated to be between 2 and 6 for the Hill coefficient, h, a Normal$(\mu = 18, \sigma = 10)$ truncated to be between 5 and 40 for the transcriptional delay τ, and a Normal$(\mu = \bar{P}, \sigma = 500^2)$ truncated to be between 1000 and $2 \times \bar{P}$, where $\bar{P}$ is the mean protein expression calculated either for a single time-series or across multiple time-series. These are uninformative priors which slightly favour parameter values that are not close to the boundary of the accepted values.

B.2 Reporting on the accuracy of variational inference

We use the root-mean-square error statistic to report on the accuracy of posterior samples for each parameter using Pathfinder, defined by

$$\text{RMSE}_\theta = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\theta_i - \theta)^2}, \tag{B.1}$$

where θ is the true value of the parameter and $\{\theta_i\}_{i=1}^n$ are the posterior samples.

Number of cells	1	10	20	30	40
RMSE ($\mathbf{h}$)	0.69	0.58	0.55	0.62	0.53
SD ($\mathbf{h}$)	0.63	0.40	0.24	0.26	0.15
RMSE ($\boldsymbol{\tau}$)	5.0	4.8	5.6	2.8	4.9
SD ($\boldsymbol{\tau}$)	5.0	1.7	1.2	0.98	0.83

Table B.1: Root-mean-sqaure error and standard deviation values for the parameters h, and τ, as the amount of cells increases. These are calculated for the posterior samples shown in Figure 4.5. Values for P_0 are given in Table 4.6.

B.3 Continuous variance derivation

Here we derive the general form of the updated covariance for some arbitrary time point (t, s). In other words, $\mathbf{P}^*(s,t)$. Recall the notation

$$\mathbf{P}_{1:T} = \begin{bmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,T} \\ P_{2,1} & P_{2,2} & \cdots & P_{2,T} \\ \vdots & \vdots & \ddots & \vdots \\ P_{T,1} & P_{T,2} & \cdots & P_{T,T} \end{bmatrix}.$$

Writing out the *discretised* version we derived above over a range $[1, T]$, we have

$$\begin{aligned} \mathbf{P}^*_{1:T} &= \mathbf{P}_{1:T} - P_{1:T,T}F^T(FP_TF^T + \Sigma_\epsilon)^{-1}FP_{T,1:T} && \text{(B.2)} \\ &= \mathbf{P}_{1:T} - S_T^{-1}\left(P_{1:T,T}F^TFP_{T,1:T}\right) && \text{(B.3)} \end{aligned}$$

$$= \mathbf{P}_{1:T} - S_{1:T}^{-1} \begin{bmatrix} P_{1,T}F^T \\ P_{2,T}F^T \\ \vdots \\ P_{T,T}F^T \end{bmatrix} \begin{bmatrix} FP_{T,1} & FP_{T,2} & \cdots & FP_{T,T} \end{bmatrix} \tag{B.4}$$

$$= \mathbf{P}_{1:T} - S_T^{-1} \begin{bmatrix} P_{1,T}F^TFP_{T,1} & \cdots & P_{1,T}F^TFP_{T,T} \\ \vdots & \ddots & \vdots \\ P_{T,T}F^TFP_{T,1} & \cdots & P_{T,T}F^TFP_{T,T} \end{bmatrix}. \tag{B.5}$$

where $S_T = FP_TF^T + \Sigma_\epsilon \in \mathbb{R}$. The (i,j)-th entry of the matrix in eq. (B.5) is given by

$$P_{i,j} - S_T^{-1} P_{i,T} F^T F P_{T,j}.$$

The continuous version can therefore be written as

$$\mathbf{P}^*(s,t) = \mathbf{P}(s,t) - S_T^{-1}\mathbf{P}(s,T)F^TF\mathbf{P}(T,t) \tag{B.6}$$

$$= \mathbf{P} - C(s)F\mathbf{P}(T,t) \tag{B.7}$$

where

$$C(s) = \mathbf{P}(s,T)F^T(F\mathbf{P}(T,T)F^T + \Sigma_\epsilon)^{-1}.$$

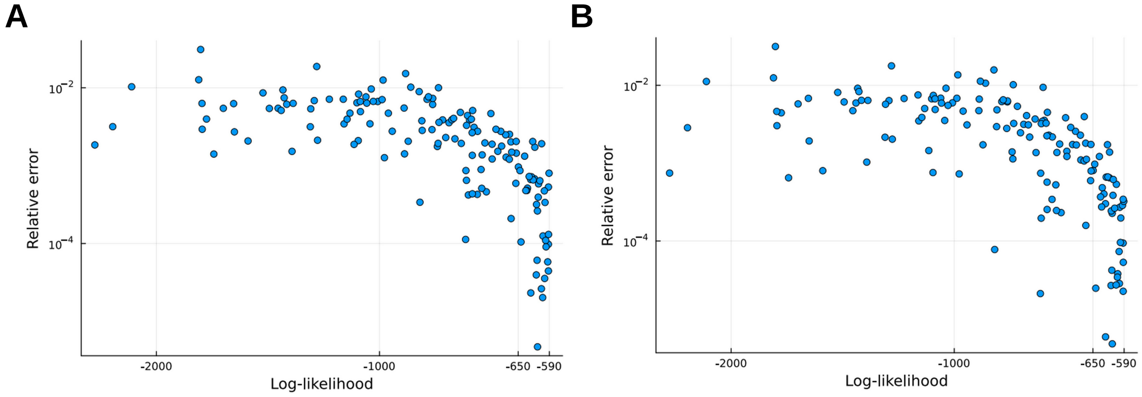


Figure B.1: **Increasing the solver step size does not significantly improve accuracy across parameter space.** **A.** Relative log-likelihood error using the RK4 solver with a solver step size and off-diagonal step size of 10.0. **B.** Relative log-likelihood error using the RK4 solver with a solver step size and off-diagonal step size of 5.0. In both cases, the log-likelihood is compared to the value calculated when the solver and off-diagonal step sizes are 1.0.

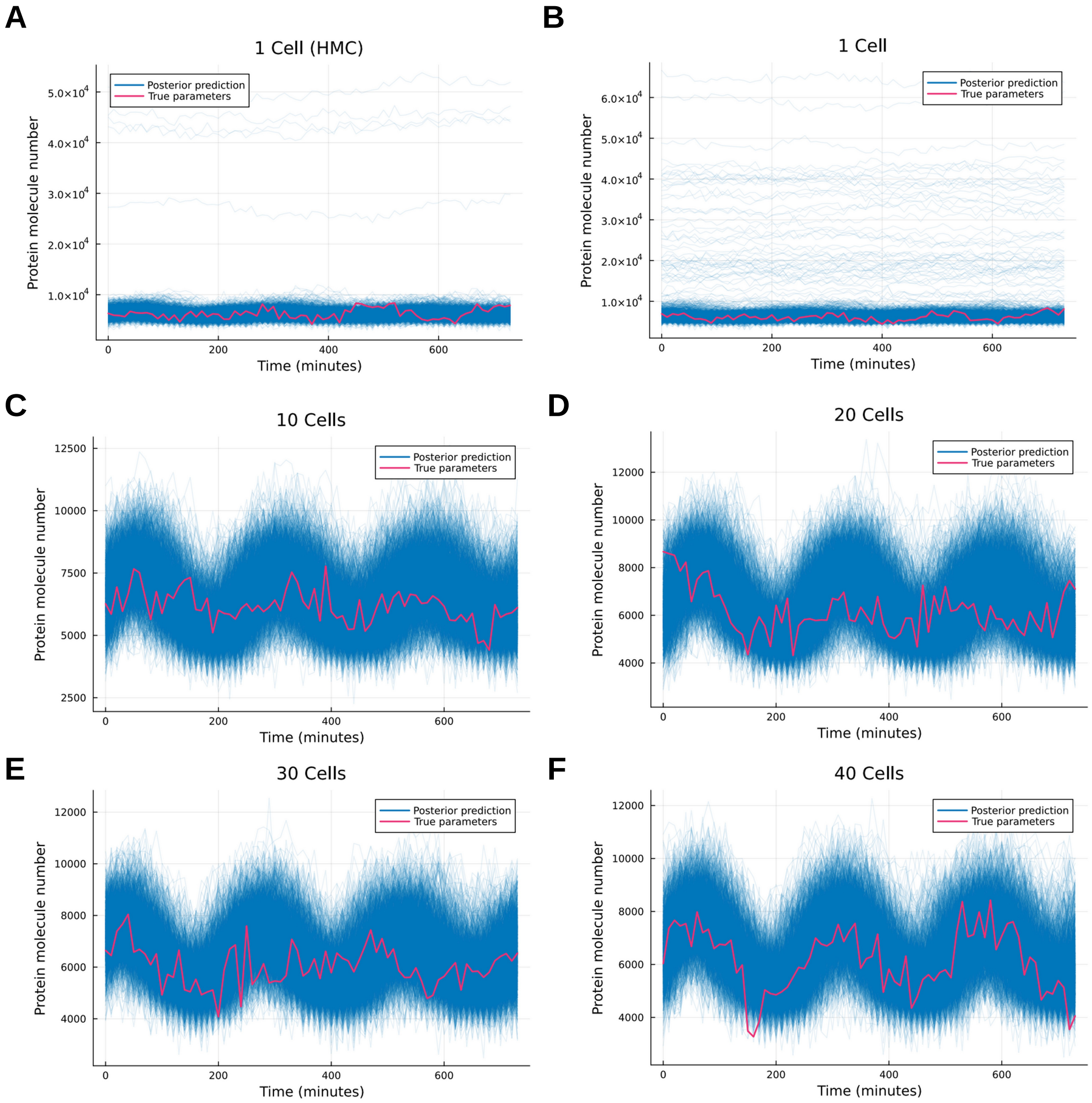


Figure B.2: **Posterior predictive checks using posterior samples from Figure 4.5.** We use the posterior samples obtained from both HMC and Pathfinder to simulate experimental data (blue lines) which are plotted alongside experimental data simulated using the true parameter values (pink line). **A.** Simulated experimental data from samples obtained using HMC. For the most part, the mean expression levels are captured, but there are some outliers indicating the posterior is inconsistent with the data. **B.** Same as A but with samples from Pathfinder. **C–F.** Posterior predictions improve when more data is used in the inference, and both the mean expression and periodicity of oscillations in the data are captured.

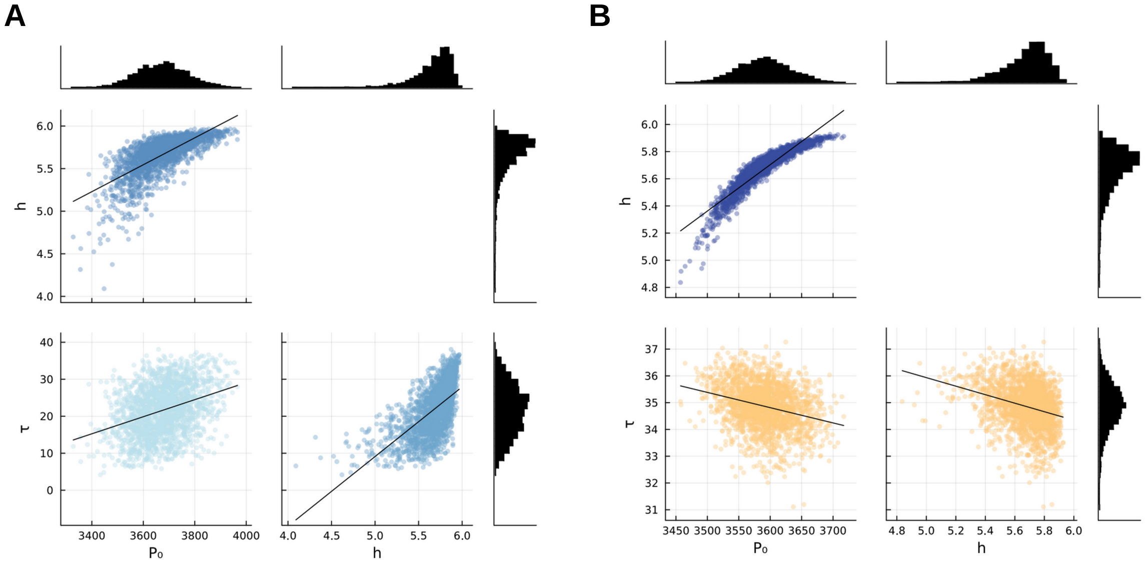


Figure B.3: **Joint posterior plots show the correlation between parameters. A.** Joint posterior plots for the posterior samples from Pathfinder when using data from 1 cell. **B.** Joint posterior plots for the posterior samples from Pathfinder when using data from 40 cells. The relationship between P_0 and h becomes more constrained when more data is taken into account. This agrees with previous results (see Figure A.6, (Burton et al., 2021)).

Appendix C

Sequential and additive expression of miR-9 precursors control timing of neurogenesis supplementary

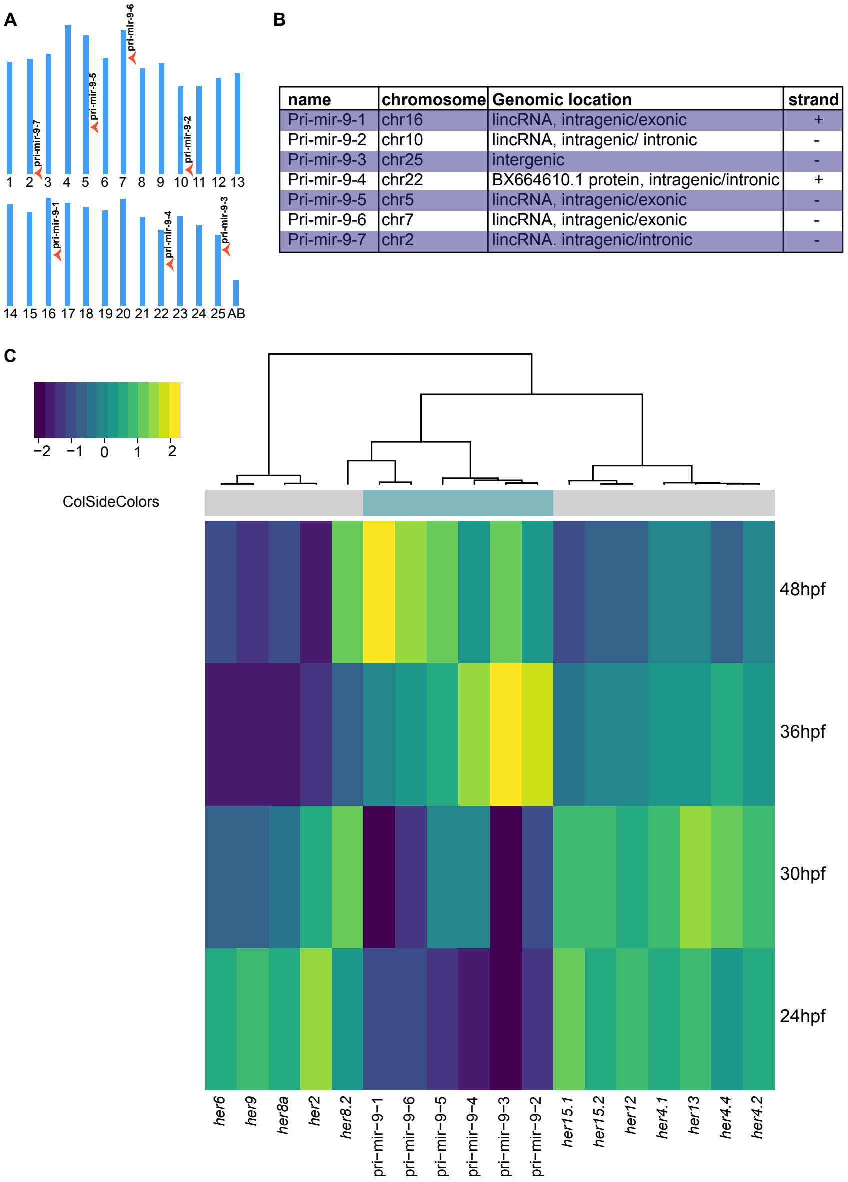
Gene	Oligonucleotides sequence 5 > 3
Pri-mir-9-1 f	cagattgacagagttgtgag
Pri-mir-9-1 r	cagtgctgactactctaatg
Pri-mir-9-2 f	gttcaatcctcttccgtttg
*Pri-mir-9-2 r	cagcatcccgttacacattc
*Pri-mir-9-3 f	aatgctgagttttgccacct
Pri-mir-9-3 r	tgctgcggaaaataacacaa
Pri-mir-9-4 f	ttccacaagggtatcgatag
Pri-mir-9-4 r	tattatatgagaaccacgtg
Pri-mir-9-5 f	gatgttatttccgcgtgcac
Pri-mir-9-5 r	tcagctctctctatatgtcc
*Pri-mir-9-6 f	gagagacaacatcgcatcca
*Pri-mir-9-6 r	tcaaacatagcaggatagggtct
Pri-mir-9-7 f	tactaattaacctataacgcttgc
Pri-mir-9-7 r	tctactttcggttctctagc

Table C.1: **Sequences of primers used to generate probes for *in situ* hybridisation.** * Primers for Pri-mir-9-3 and Pri-mir-9-6 were used to clone the probes that were kindly provided by the Bally-Cuif lab.

Number	Gene name	Primer sequence 5 > 3
1	Pri-mir-9-1	aggggttggctgttatctttgatctagcagtatcagtgttattc
2	Pri-mir-9-4	tgggttagtttttctcttaggtaatcctgtatgagtttatgtgatatcataaa
3	Pri-mir-9-5	aaatactcatacagcaagataaaacataacaactcgcttccaattcc
4	Pri-mir-9-7	acgggttagtttttctctttcgttttcttgtatgagttatgaaatatcataaag

Table C.2: **Primers for site-directed mutagenesis.**

Figure S1



name	chromosome	Genomic location	strand
Pri-mir-9-1	chr16	lincRNA, intragenic/exonic	+
Pri-mir-9-2	chr10	lincRNA, intragenic/ intronic	-
Pri-mir-9-3	chr25	intergenic	-
Pri-mir-9-4	chr22	BX664610.1 protein, intragenic/intronic	+
Pri-mir-9-5	chr5	lincRNA, intragenic/exonic	-
Pri-mir-9-6	chr7	lincRNA, intragenic/exonic	-
Pri-mir-9-7	chr2	lincRNA. intragenic/intronic	-

Figure C.1: **miR-9 host gene expression negatively correlates with Hes/Her gene expression over developmental time. A.** Map of miR-9 paralogues in *Danio rerio* produced with information from Vega genome browser, data was collated from ensembl genome browser GRCz10, showing chromosomal loci. Red arrowhead indicates location of respective pri-mir-9. **B.** Table showing the genomic loci of the seven Danio rerio miR-9 paralogues, their host gene, exonic or intronic location within the host gene and strand orientation. **C.** Heatmap showing z-scored expression data over developmental time for the miR-9 host / pri-mirs and Hes/Her genes over developmental time in the zebrafish embryo. Data is from White et al. (2017)

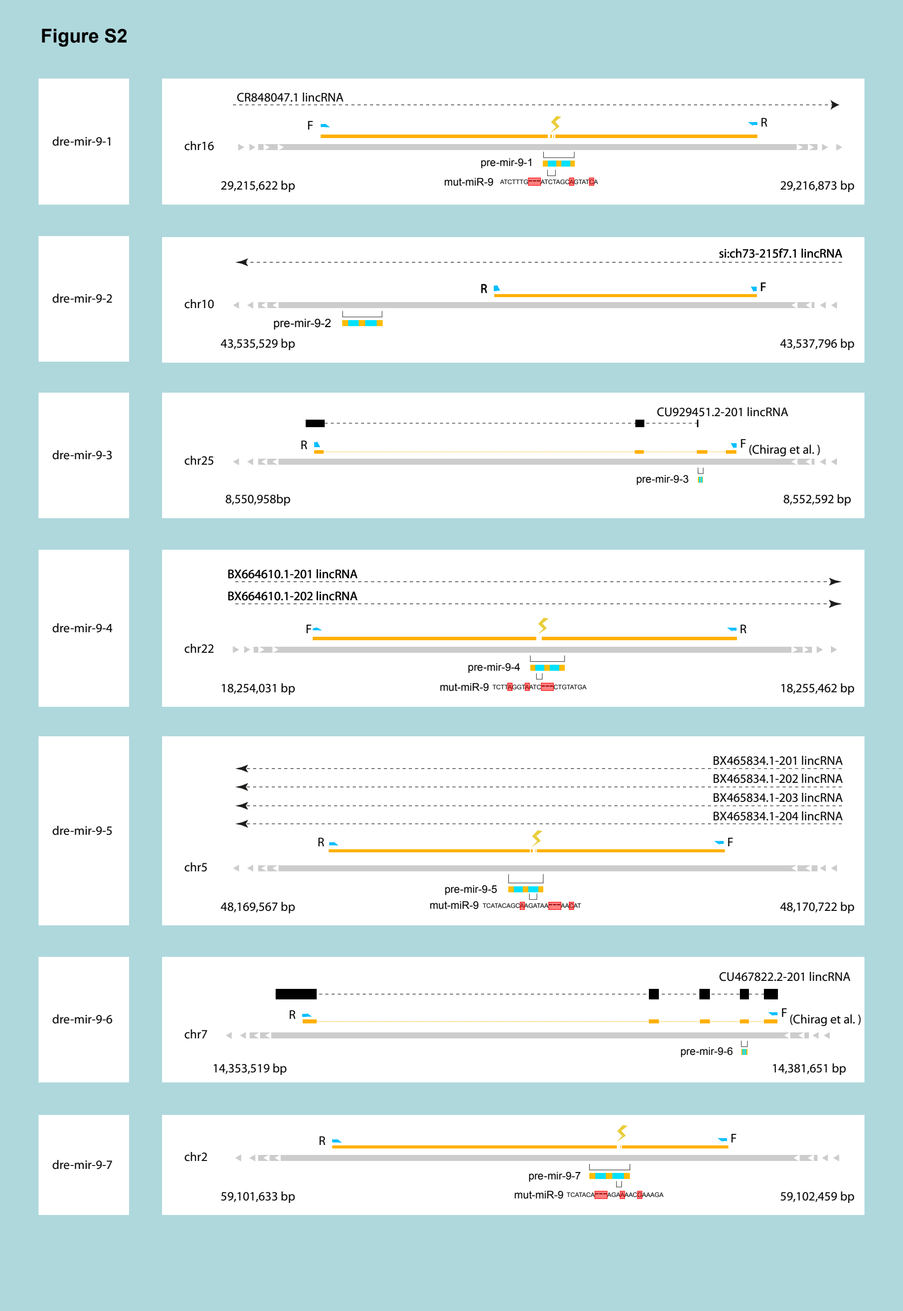


Figure C.2: **Probe design for in situ hybridisations of pri-mir-9s.** The probes are indicated by the orange blocks. Where probes are on exons, introns are indicated by dashed lines. Forward (F) and reverse (R) are visualised by blue arrows. Overlapping transcripts are indicated by the black boxes and dashed lines. The microRNA hairpin is indicated by a small orange box broken by two blue boxes indicating the mature sequences. Probes for pri-mir-9-3 and pri-mir-9-6 were based on those from Nepal et al. (2016). Mutations in the mature sequence are indicated by lightning bolts and also annotated underneath.

Figure S3

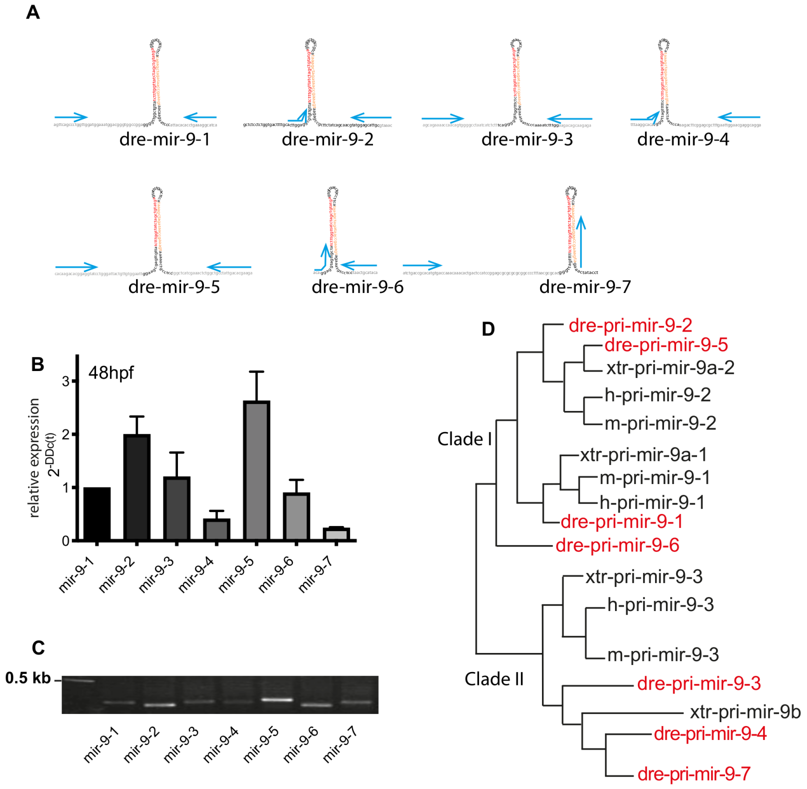
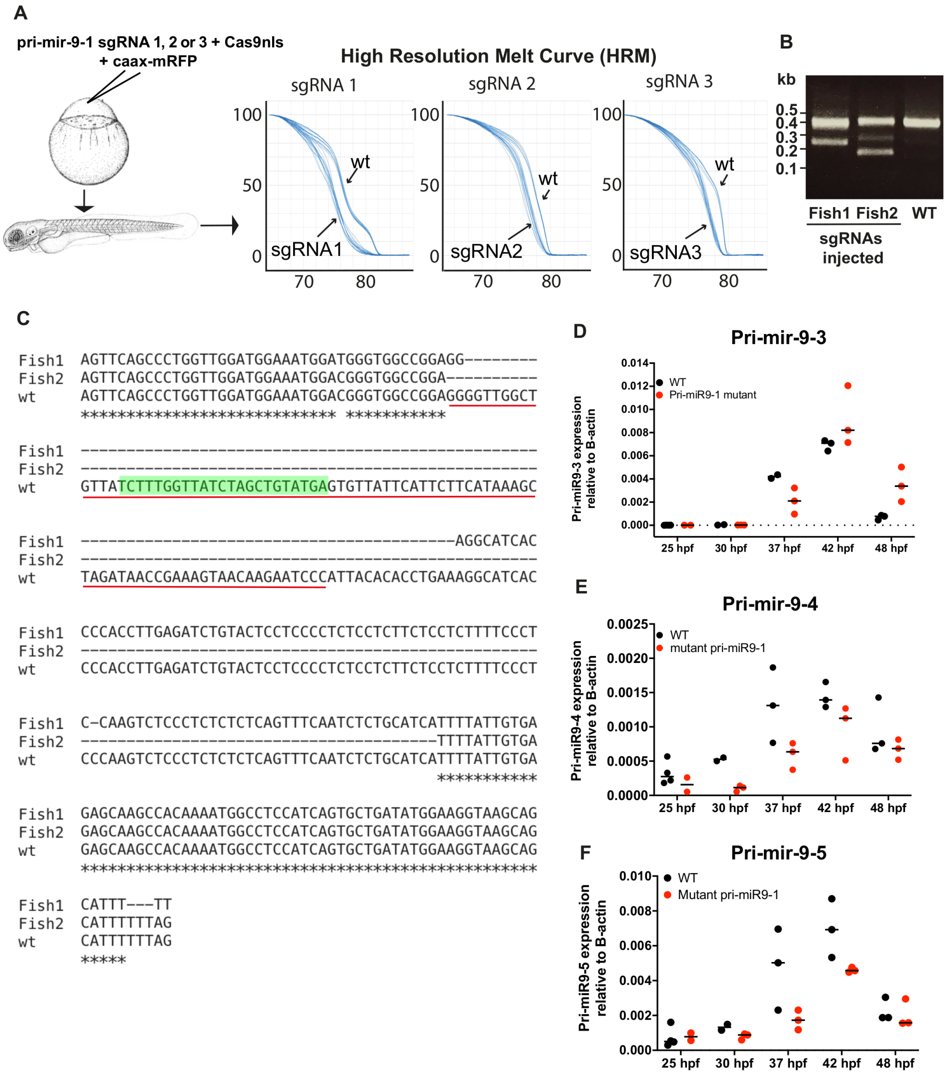


Figure C.3: **All seven miR-9 paralogues are expressed at 48hpf. A.** Schematic of the seven miR-9 paralogues hairpin loops with the respective primers used for qRTPCR annotated as blue arrows (Section 5.6.2; Table C.3). Red sequence: miR-9-5' arm, orange sequence: miR-9-3' arm, black letters: pre-mir-9, grey sequence: partial sequence of pri-mir-9. **B.** Bar plot showing relative expression of the seven miR-9 paralogues at 48hpf, $N = 3$. Error bar represent mean and SEM. **C.** Amplicons of the seven miR-9 paralogues generated by qRT-PCR using the respective primers (Table C.3). **D.** Evolutionary relationship of pri-mir-9 family modified from Alwin Prem Anand et al. (2018) phylogenetic analysis. Highlighted in red letter are the seven miR-9 paralogues. The zebrafish miR-9 paralogues cluster into two Clades: Clade I that is divided into 2 subgroups: Subgroup I corresponding to the group in which human (h) and mouse (m) pri-mir-9-2 precursor cluster with Danio rerio (dre) pri-mir-9-2 and pri-mir-9-5 precursors and subgroup II corresponding to the group where h/m-pri-mir-9-1 precursor clusters with dre-pri-mir9-1. Interestingly, dre-pri-mir-9-6 is closely related to clade I but doesn't group with any subgroup. Clade II correspond to the branch where m/h-pri-miR-9-3 group with dre-pri-mir-9-3, dre-pri-mir-9-4 and dre-pri-mir-9-7.

Figure S4



```
Fish1   AGTTCAGCCCTGGTTGGATGGAAATGGATGGGTGGCCGGAGG--------
Fish2   AGTTCAGCCCTGGTTGGATGGAAATGGACGGGTGGCCGGA----------
wt      AGTTCAGCCCTGGTTGGATGGAAATGGACGGGTGGCCGGAGGGGTTGGCT
        **************************** ***********

Fish1   --------------------------------------------------
Fish2   --------------------------------------------------
wt      GTTATCTTTGGTTATCTAGCTGTATGAGTGTTATTCATTCTTCATAAAGC

Fish1   -----------------------------------------AGGCATCAC
Fish2   --------------------------------------------------
wt      TAGATAACCGAAAGTAACAAGAATCCCATTACACACCTGAAAGGCATCAC

Fish1   CCCACCTTGAGATCTGTACTCCTCCCCTCTCCTCTTCTCCTCTTTTCCCT
Fish2   --------------------------------------------------
wt      CCCACCTTGAGATCTGTACTCCTCCCCTCTCCTCTTCTCCTCTTTTCCCT

Fish1   C-CAAGTCTCCCTCTCTCTCAGTTTCAATCTCTGCATCATTTTATTGTGA
Fish2   ---------------------------------------TTTTATTGTGA
wt      CCCAAGTCTCCCTCTCTCTCAGTTTCAATCTCTGCATCATTTTATTGTGA
                                               ***********

Fish1   GAGCAAGCCACAAAATGGCCTCCATCAGTGCTGATATGGAAGGTAAGCAG
Fish2   GAGCAAGCCACAAAATGGCCTCCATCAGTGCTGATATGGAAGGTAAGCAG
wt      GAGCAAGCCACAAAATGGCCTCCATCAGTGCTGATATGGAAGGTAAGCAG
        **************************************************

Fish1   CATTT---TT
Fish2   CATTTTTTAG
wt      CATTTTTTAG
        *****
```

Figure C.4: **Deletion of pre-mir-9-1. A.** (A-left) Schematic representation of experimental procedure used to delete premir-9-1, (A-right) high resolution melt graphs obtained from: wt (wildtype) versus sgRNA 1, 2 or 3 injected fish, respectively. **B.** Agarose gel showing the size of the amplicon in wt fish (395bp) and in injected with three sgRNA fish, ∼250bp and 395bp in fish1 and ∼190bp and 395bp in fish2. **C.** Representative examples of sequences obtained from F0 embryos showing deletion of pre-mir-9-1. Red underline indicates premir-9-1 sequence. Green highlighted indicates mature miR-9 sequence. **D–F.** SYBR green relative quantification of pri-mir-9-3 (D), pri-mir-9-4 (E) and pri-mir-9-5 (F) from dissected hindbrain at different stages of development, in wild-type conditions (black dots) and deletion of pre-mir-9-1 (red dots), quantification was normalised using β-actin. $N = 3$.

Figure S5

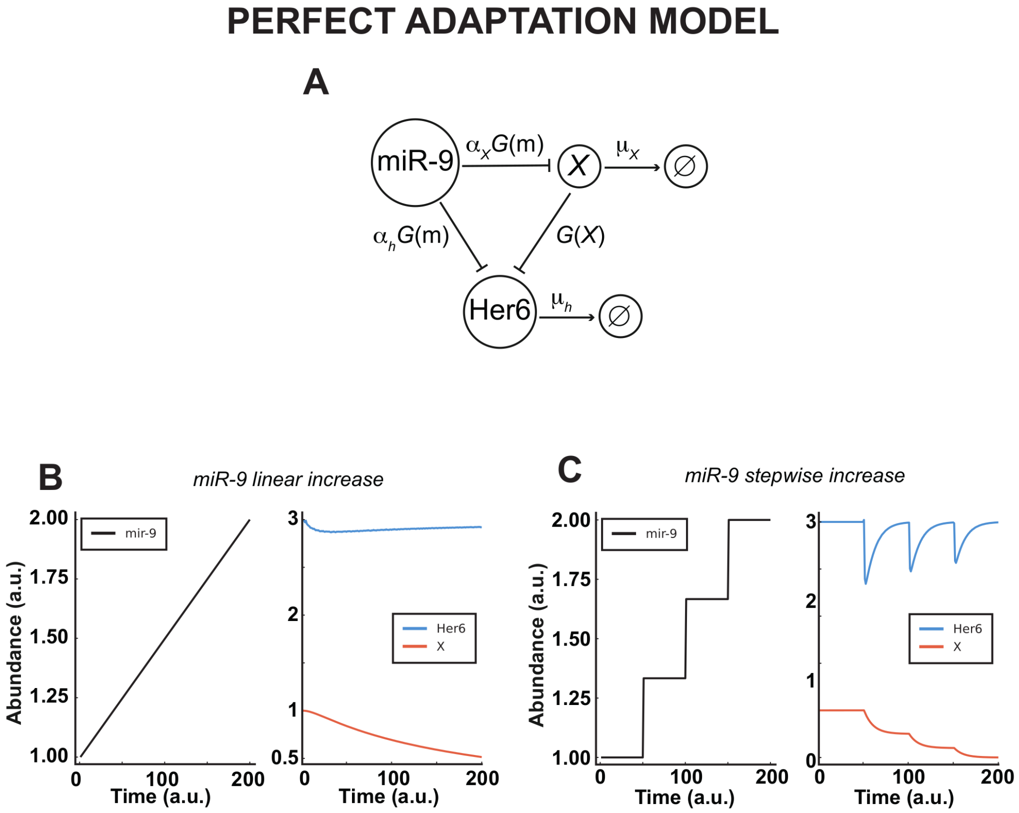


Figure C.5: **Her6 is not down-regulated by miR-9 in the perfect adaptation model. A–C.** Perfect adaptation model. (A) A schematic of the perfect adaptation model based on an incoherent feed forward loop. The parameters μ_h, α_h and α_X are positive real constants. μ_h represents the degradation rates of Her6 and α_h and α_X represent the basal production rate of Her6 and X respectively. See Section 5.6.13, parameter values are given in Table C.14. **B.** A linear mir-9 expression profile leads to a small initial response in Her6 expression levels, which returns to steady state levels due to the perfect adaptation. **C.** Large instantaneous changes in miR-9 cause a large drop in Her6 levels, which then returns to steady state. The model detects fold changes in miR-9 and repetitive steps result in a diminished response from Her6.

Figure S6

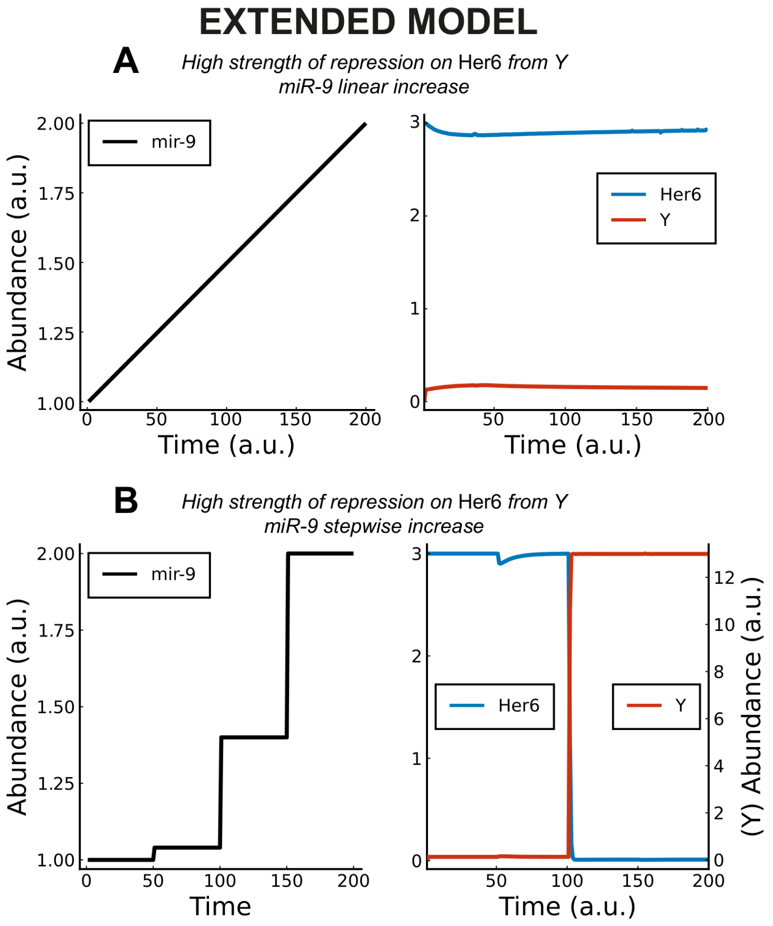
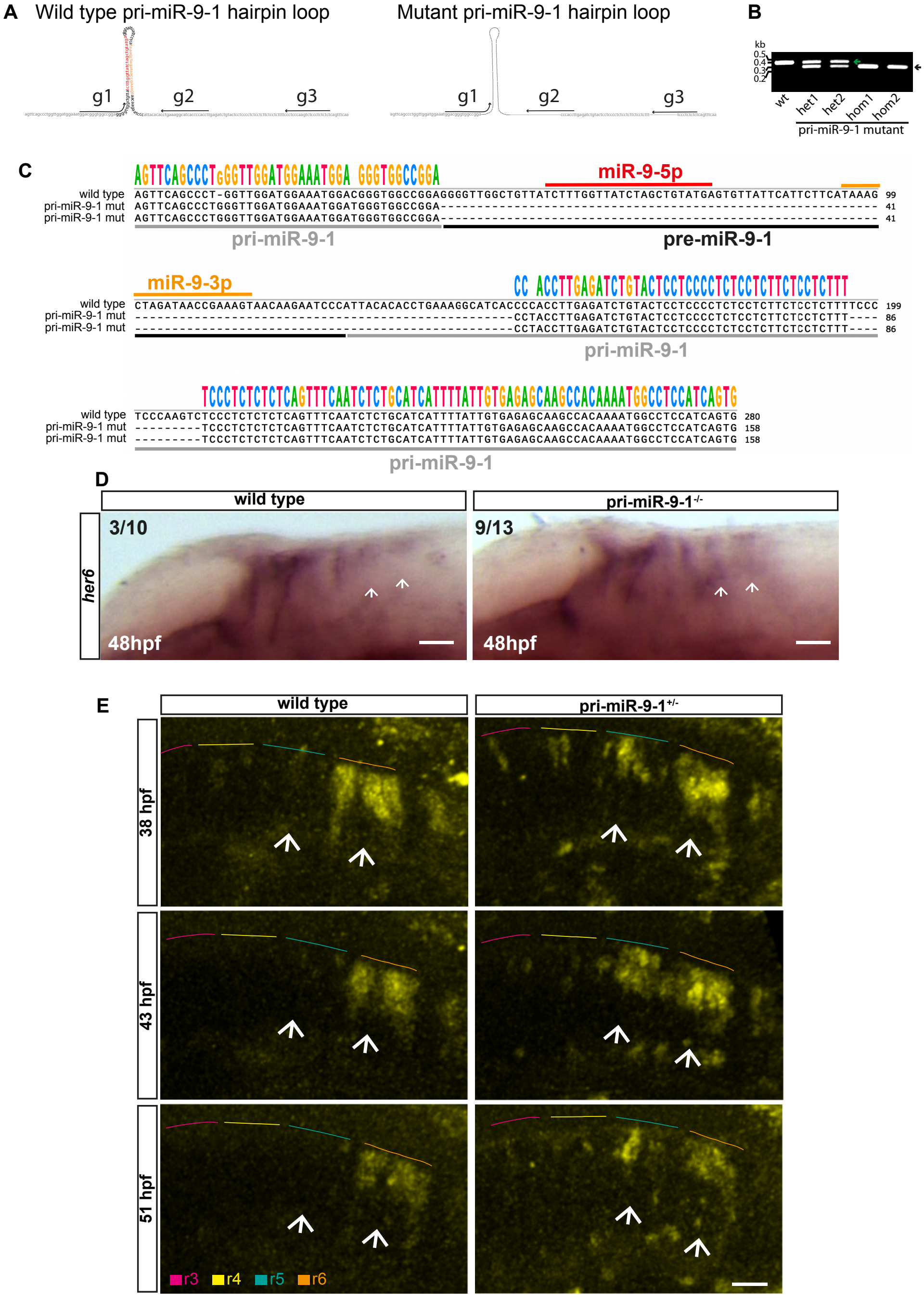


Figure C.6: **Her6 is down-regulated by miR-9 stepwise increase in extended adaptation mathematical model. A–B.** Dynamics of Her6 in response to linear increase in miR-9 (A) or stepwise increase in miR-9 (B) with increased repression strength from Y. The stepwise increase in miR-9 under high repression strength from Y causes Her6 to switch off given a large enough increase in miR-9.

Figure S7



```
AGTTCAGCCCTgGGTTGGATGGAAATGGA GGGTGGCCGGA
wild type       AGTTCAGCCCT-GGTTGGATGGAAATGGACGGGTGGCCGGAGGGGTTGGCTGTTATCTTTGGTTATCTAGCTGTATGAGTGTTATTCATTCTTCATAAAG 99
pri-miR-9-1 mut AGTTCAGCCCTGGGTTGGATGGAAATGGATGGGTGGCCGGA----------------------------------------------------------- 41
pri-miR-9-1 mut AGTTCAGCCCTGGGTTGGATGGAAATGGATGGGTGGCCGGA----------------------------------------------------------- 41

CC ACCTTGAGATCTGTACTCCTCCCCTCTCCTCTTCTCCTCTTT
wild type       CTAGATAACCGAAAGTAACAAGAATCCCATTACACACCTGAAAGGCATCACCCCACCTTGAGATCTGTACTCCTCCCCTCTCCTCTTCTCCTCTTTTCCC 199
pri-miR-9-1 mut ---------------------------------------------------CCTACCTTGAGATCTGTACTCCTCCCCTCTCCTCTTCTCCTCTTT---- 86
pri-miR-9-1 mut ---------------------------------------------------CCTACCTTGAGATCTGTACTCCTCCCCTCTCCTCTTCTCCTCTTT---- 86

TCCCTCTCTCTCAGTTTCAATCTCTGCATCATTTTATTGTGAGAGCAAGCCACAAAATGGCCTCCATCAGTG
wild type       TCCCAAGTCTCCCTCTCTCTCAGTTTCAATCTCTGCATCATTTTATTGTGAGAGCAAGCCACAAAATGGCCTCCATCAGTG 280
pri-miR-9-1 mut ---------TCCCTCTCTCTCAGTTTCAATCTCTGCATCATTTTATTGTGAGAGCAAGCCACAAAATGGCCTCCATCAGTG 158
pri-miR-9-1 mut ---------TCCCTCTCTCTCAGTTTCAATCTCTGCATCATTTTATTGTGAGAGCAAGCCACAAAATGGCCTCCATCAGTG 158
```

Figure C.7: **Generation of pri-miR-9-1 mutant fish line.**

Figure C.7: **A.** Schematic of pri-miR-9-1 hairpin loops with the customized guide RNAs used to specifically delete pre-mir-9-1, annotated as g1, g2 and g3. Left panel, sequence of wild type pri-miR-9-1 hairpin loop. Right panel, sequence of mutant pri-miR-9-1 after CRISPR deletion with dashed line indicating the deleted region and loss of hairpin loop. **B.** Amplicons obtained from F1 generation of wildtype (green arrow) and pri-miR-9-1 mutant (black arrow) embryos, 395bp and 273bp, respectively. wt: wildtype; het: heterozygote; home: homozygote. **C.** Sequence showing the region deleted in the pri-miR-9-1 mutant. (A and C) Red: miR-9 5' arm, orange: miR-9 3' arm, black: pre-mir-9, grey: partial sequence of pri-mir-9-1. **D.** Representative image of Chromogenic WM-ISH performed at 48hpf showing *her6* expression in wild type embryos and in pri-miR-9-1$^{-/-}$ mutant embryos. Note ventral extension of her6 expression in pri-miR-9-1$^{-/-}$ embryos (9/13) compared to wildtype embryos (3/10) as indicated by white arrowheads; longitudinal view, scale bar 30μm. **E.** Representative confocal images of wild type and pri-miR-9-1$^{+/-}$ embryos showing Her6::Venus$^{+/+}$ expression in the hindbrain, rhombomeres 3 to 6 (r3-r6) over the course of development; longitudinal view, scale bar 30μm. Images are representing 2D maximum projection; $N = 3$ independent experiments with 1 embryo, 1 embryo, 2 embryos per experimental condition, Her6::Venus$^{+/+}$;pri-mir-9-1$^{+/+}$ or Her6::Venus$^{+/+}$;pri-mir-9-1$^{-/+}$or$^{-/-}$. Note: The in-cross of Her6::Venus$^{+/-}$;pri-mir-9-1$^{+/-}$ adults had low rate of Her6::Venus$^{+/+}$;pri-mir-9-1$^{-/+}$or$^{-/-}$ embryos explaining the low number of embryos analysed per independent experiment. (D-E) White arrowheads indicate increased Her6::Venus expression in pri-miR-9-1 mutants.

Number	**Name**	**Sequence 5' > 3'**	**Amplicon size (bp)**
1	pri-mir-9-1 f	agttcagccctggttgg	149
2	pri-mir-9-1 r	tgatgcctttcaggtgtg	149
3	pri-mir-9-2 f	acttggaggcgtgttg	120
4	pri-mir-9-2 r	gtttacgcaatgctccatac	120
5	pri-mir-9-3 f	agcagaaaaccaacagtgg	150
6	pri-mir-9-3 r	tctcttgctgtcttccaaag	150
7	pri-mir-9-4 f	tttaaggcacatgggttag	135
8	pri-mir-9-4 r	tcctgcctcgttccaattc	135
9	pri-mir-9-5 f	cacaagacacggaggtatcc	166
10	pri-mir-9-5 r	tcttcgtgtcaataggcagcc	166
11	pri-mir-9-6 f	acaggaggtagttgctatc	104
12	pri-mir-9-6 r	tgtatgcagttaaggaggac	104
13	pri-mir-9-7 f	atctgaccgcacatgtgac	146
14	pri-mir-9-7 r	tctactttcggttctctagc	146
15	β-actin f	cgtgctgtcttcccatcca	176
16	β-actin r	tcaccaacgtagctgtctttctg	176
17	*elavl4* f	ccgtcaacaatgtcaaggtg	147
18	*elavl4* r	acacttgcaagacacggtca	147
19	*barhl2* f	ttcggtcacgatggagcatc	120
20	*barhl2* r	agcagatgaggctgaggtga	120
21	*dmbx1a* f	gcgaccccatcatctataccat	105
22	*dmbx1a* r	cgcgtggtgagatagagactga	105
23	*pax2a* f	gtgacaggtcgagagatggc	198
24	*pax2a* r	acttaataacgcggggttgct	198
25	*otpb* f	tgttggagcgatttttcagg	107
26	*otpb* r	ttcctgttgatcccgtttgaa	107
27	*tal1/scl* f	gcacgacggcggagaca	226
28	*tal1/scl* r	ggaggcagtggcaagggaa	226
29	*isl1* f	gcagcagcaacccaacgacaa	224
30	*isl1* r	ttgagcctggaccaccttcagaa	224

Table C.3: **Sequences of primers used for qRT-PCR.**

MicroRNA	ensembl gene code	Host name
dre-mir-9-1	ENSDARG00000095906	CR848047.1
dre-mir-9-2	ENSDARG00000098512	si:ch73-215f7.1
dre-mir-9-3	ENSDARG00000098395	CU929451.2
dre-mir-9-4	ENSDARG00000095557	BX664610.1
dre-mir-9-5	ENSDARG00000095715	BX465834.1
dre-mir-9-6	ENSDARG00000101584	CU467822.2
dre-mir-9-7	NA	NA

Table C.4: **microRNA host genes.**

Probe sequence name	Probe sequence (5' to 3')
pri-mir-9-1 smiFISH_1	CCTCCTAAGTTTCGAGCTGGACTCAGTGcactcactttctttctcaca
pri-mir-9-1 smiFISH_2	CCTCCTAAGTTTCGAGCTGGACTCAGTGgaggtctttctctctctcag
pri-mir-9-1 smiFISH_3	CCTCCTAAGTTTCGAGCTGGACTCAGTGcacagcaaagtgctagaatc
pri-mir-9-1 smiFISH_4	CCTCCTAAGTTTCGAGCTGGACTCAGTGgatttaccctcaagttgttg
pri-mir-9-1 smiFISH_5	CCTCCTAAGTTTCGAGCTGGACTCAGTGctagttgctttagttttccg
pri-mir-9-1 smiFISH_6	CCTCCTAAGTTTCGAGCTGGACTCAGTGaataggggggaaaagaccct
pri-mir-9-1 smiFISH_7	CCTCCTAAGTTTCGAGCTGGACTCAGTGcgcaaaggctgggatttatt
pri-mir-9-1 smiFISH_8	CCTCCTAAGTTTCGAGCTGGACTCAGTGgagagatggcttcaaacgca
pri-mir-9-1 smiFISH_9	CCTCCTAAGTTTCGAGCTGGACTCAGTGattgcaggctgatcattctc
pri-mir-9-1 smiFISH_10	CCTCCTAAGTTTCGAGCTGGACTCAGTGttcttgttgtctctgtctta
pri-mir-9-1 smiFISH_11	CCTCCTAAGTTTCGAGCTGGACTCAGTGagatgcaaagagtccattca
pri-mir-9-1 smiFISH_12	CCTCCTAAGTTTCGAGCTGGACTCAGTGaaccagcctgaactggttta
pri-mir-9-1 smiFISH_13	CCTCCTAAGTTTCGAGCTGGACTCAGTGctatgacttcaccagtctgt
pri-mir-9-1 smiFISH_14	CCTCCTAAGTTTCGAGCTGGACTCAGTGctcttgattgggtagcttaa
pri-mir-9-1 smiFISH_15	CCTCCTAAGTTTCGAGCTGGACTCAGTGtgatcacagtgctgagactg
pri-mir-9-1 smiFISH_16	CCTCCTAAGTTTCGAGCTGGACTCAGTGactgacgtaggttatctgta
pri-mir-9-1 smiFISH_17	CCTCCTAAGTTTCGAGCTGGACTCAGTGggctgaactgcaacctgaaa
pri-mir-9-1 smiFISH_18	CCTCCTAAGTTTCGAGCTGGACTCAGTGctttcaggtgtgtaatggga
pri-mir-9-1 smiFISH_19	CCTCCTAAGTTTCGAGCTGGACTCAGTGagtacagatctcaaggtggg
pri-mir-9-1 smiFISH_20	CCTCCTAAGTTTCGAGCTGGACTCAGTGagaggagaagaggagagggg
pri-mir-9-1 smiFISH_21	CCTCCTAAGTTTCGAGCTGGACTCAGTGgagagagggagacttgggag
pri-mir-9-1 smiFISH_22	CCTCCTAAGTTTCGAGCTGGACTCAGTGtggcttgctctcacaataaa
pri-mir-9-1 smiFISH_23	CCTCCTAAGTTTCGAGCTGGACTCAGTGagcactgatggaggccattt
pri-mir-9-1 smiFISH_24	CCTCCTAAGTTTCGAGCTGGACTCAGTGaaaatgctgcttaccttcca
pri-mir-9-1 smiFISH_25	CCTCCTAAGTTTCGAGCTGGACTCAGTGggaggagataatcacagctt
pri-mir-9-1 smiFISH_26	CCTCCTAAGTTTCGAGCTGGACTCAGTGagcaatgagagagactgggt
pri-mir-9-1 smiFISH_27	CCTCCTAAGTTTCGAGCTGGACTCAGTGacgcaccttttacaatagca
pri-mir-9-1 smiFISH_28	CCTCCTAAGTTTCGAGCTGGACTCAGTGtagatttgtgaggaggggtg
pri-mir-9-1 smiFISH_29	CCTCCTAAGTTTCGAGCTGGACTCAGTGagtgctgactactctaatgt
pri-mir-9-1 smiFISH_30	CCTCCTAAGTTTCGAGCTGGACTCAGTGattccgcgtattaaacgctt
pri-mir-9-1 smiFISH_31	CCTCCTAAGTTTCGAGCTGGACTCAGTGccctacggagagtgaaatga
pri-mir-9-1 smiFISH_32	CCTCCTAAGTTTCGAGCTGGACTCAGTGcattcagcttctccgtatcg
pri-mir-9-1 smiFISH_33	CCTCCTAAGTTTCGAGCTGGACTCAGTGcactccagaatgttgtcttg
pri-mir-9-1 smiFISH_34	CCTCCTAAGTTTCGAGCTGGACTCAGTGaaatatgcgcgtgcgagctg
pri-mir-9-1 smiFISH_35	CCTCCTAAGTTTCGAGCTGGACTCAGTGattcaggaccgatcgtaatg
pri-mir-9-1 smiFISH_36	CCTCCTAAGTTTCGAGCTGGACTCAGTGcctatcctcgagcatcgatg

Table C.5: **smiFISH Probe sequences for pri-mir-9-1.**

Probe sequence name	**Probe sequence (5' to 3')**
pri-mir-9-4 smiFISH_1	CCTCCTAAGTTTCGAGCTGGACTCAGTGctatcgatacccttgtggaa
pri-mir-9-4 smiFISH_2	CCTCCTAAGTTTCGAGCTGGACTCAGTGtggaaacgttgcgtttggtt
pri-mir-9-4 smiFISH_3	CCTCCTAAGTTTCGAGCTGGACTCAGTGacactgtcaagacactgcta
pri-mir-9-4 smiFISH_4	CCTCCTAAGTTTCGAGCTGGACTCAGTGatggttagcatttgttcgtg
pri-mir-9-4 smiFISH_5	CCTCCTAAGTTTCGAGCTGGACTCAGTGatgcacctcaaaaaactgct
pri-mir-9-4 smiFISH_6	CCTCCTAAGTTTCGAGCTGGACTCAGTGcagtatgtaggcgcttaagg
pri-mir-9-4 smiFISH_7	CCTCCTAAGTTTCGAGCTGGACTCAGTGtcttttatgtttggagctgg
pri-mir-9-4 smiFISH_8	CCTCCTAAGTTTCGAGCTGGACTCAGTGgcatcgcagttgtttacaat
pri-mir-9-4 smiFISH_9	CCTCCTAAGTTTCGAGCTGGACTCAGTGctcttgcgtttgcactttta
pri-mir-9-4 smiFISH_10	CCTCCTAAGTTTCGAGCTGGACTCAGTGtagtctaggtgttattgcgc
pri-mir-9-4 smiFISH_11	CCTCCTAAGTTTCGAGCTGGACTCAGTGtacgctcatattatgtgcgt
pri-mir-9-4 smiFISH_12	CCTCCTAAGTTTCGAGCTGGACTCAGTGcacacacacatacacgtggg
pri-mir-9-4 smiFISH_13	CCTCCTAAGTTTCGAGCTGGACTCAGTGatacagtctcactcactctt
pri-mir-9-4 smiFISH_14	CCTCCTAAGTTTCGAGCTGGACTCAGTGatggagctattgcattcact
pri-mir-9-4 smiFISH_15	CCTCCTAAGTTTCGAGCTGGACTCAGTGcagatgcagcttctgttgtg
pri-mir-9-4 smiFISH_16	CCTCCTAAGTTTCGAGCTGGACTCAGTGaaaactaacccatgtgcctt
pri-mir-9-4 smiFISH_17	CCTCCTAAGTTTCGAGCTGGACTCAGTGtaatctcaaggtctacgggg
pri-mir-9-4 smiFISH_18	CCTCCTAAGTTTCGAGCTGGACTCAGTGtcaaacgacagcactcacaa
pri-mir-9-4 smiFISH_19	CCTCCTAAGTTTCGAGCTGGACTCAGTGacatccagaacatccatttc
pri-mir-9-4 smiFISH_20	CCTCCTAAGTTTCGAGCTGGACTCAGTGccatccagcgttttacaaaa
pri-mir-9-4 smiFISH_21	CCTCCTAAGTTTCGAGCTGGACTCAGTGaattcaaagcgctccgaagt
pri-mir-9-4 smiFISH_22	CCTCCTAAGTTTCGAGCTGGACTCAGTGggattcgatgctctttgtag
pri-mir-9-4 smiFISH_23	CCTCCTAAGTTTCGAGCTGGACTCAGTGggaaacgcattaacgcagcg
pri-mir-9-4 smiFISH_24	CCTCCTAAGTTTCGAGCTGGACTCAGTGacagcttttggtattccagt
pri-mir-9-4 smiFISH_25	CCTCCTAAGTTTCGAGCTGGACTCAGTGacgtaaaatccacgtcttcc
pri-mir-9-4 smiFISH_26	CCTCCTAAGTTTCGAGCTGGACTCAGTGtatgtagccggaggcagatt
pri-mir-9-4 smiFISH_27	CCTCCTAAGTTTCGAGCTGGACTCAGTGcgtactgccagcgtactaac
pri-mir-9-4 smiFISH_28	CCTCCTAAGTTTCGAGCTGGACTCAGTGgtatccgcctgaataccaag
pri-mir-9-4 smiFISH_29	CCTCCTAAGTTTCGAGCTGGACTCAGTGcttgagaatgatcgcgaagt
pri-mir-9-4 smiFISH_30	CCTCCTAAGTTTCGAGCTGGACTCAGTGggaatgttgcccagcacaag
pri-mir-9-4 smiFISH_31	CCTCCTAAGTTTCGAGCTGGACTCAGTGacatcatcatcacgtctatg
pri-mir-9-4 smiFISH_32	CCTCCTAAGTTTCGAGCTGGACTCAGTGgtcatcccgaattcatacat
pri-mir-9-4 smiFISH_33	CCTCCTAAGTTTCGAGCTGGACTCAGTGaacacctcgatgaaccgatt
pri-mir-9-4 smiFISH_34	CCTCCTAAGTTTCGAGCTGGACTCAGTGagagatccgaccaatcggtt
pri-mir-9-4 smiFISH_35	CCTCCTAAGTTTCGAGCTGGACTCAGTGtatatgagaaccacgtgacc

Table C.6: **smiFISH Probe sequences for pri-mir-9-4.**

Probe sequence name	Probe sequence (5’ to 3’)
pri-mir-9-5 smiFISH_1	CCTCCTAAGTTTCGAGCTGGACTCAGTGacggaggagatcggtaatgc
pri-mir-9-5 smiFISH_2	CCTCCTAAGTTTCGAGCTGGACTCAGTGttagttttggccgagtggtg
pri-mir-9-5 smiFISH_3	CCTCCTAAGTTTCGAGCTGGACTCAGTGactgaatggagggatcgctc
pri-mir-9-5 smiFISH_4	CCTCCTAAGTTTCGAGCTGGACTCAGTGataacatcgcgtttgtttcg
pri-mir-9-5 smiFISH_5	CCTCCTAAGTTTCGAGCTGGACTCAGTGaggaaaaaaagtgcacgcgg
pri-mir-9-5 smiFISH_6	CCTCCTAAGTTTCGAGCTGGACTCAGTGgaggggaaaaatccgattcc
pri-mir-9-5 smiFISH_7	CCTCCTAAGTTTCGAGCTGGACTCAGTGgaaccagcagcgagattata
pri-mir-9-5 smiFISH_8	CCTCCTAAGTTTCGAGCTGGACTCAGTGtggcagcagcgatagatatc
pri-mir-9-5 smiFISH_9	CCTCCTAAGTTTCGAGCTGGACTCAGTGgctcctttggctctattaaa
pri-mir-9-5 smiFISH_10	CCTCCTAAGTTTCGAGCTGGACTCAGTGttacctcccaatctcacaaa
pri-mir-9-5 smiFISH_11	CCTCCTAAGTTTCGAGCTGGACTCAGTGtacaggccatcatttgcatt
pri-mir-9-5 smiFISH_12	CCTCCTAAGTTTCGAGCTGGACTCAGTGcctgtcgaataagaagcagt
pri-mir-9-5 smiFISH_13	CCTCCTAAGTTTCGAGCTGGACTCAGTGtatgagagttggagtgaggg
pri-mir-9-5 smiFISH_14	CCTCCTAAGTTTCGAGCTGGACTCAGTGaccggagcagttaacgttaa
pri-mir-9-5 smiFISH_15	CCTCCTAAGTTTCGAGCTGGACTCAGTGtggatatgctcgcaaccaaa
pri-mir-9-5 smiFISH_16	CCTCCTAAGTTTCGAGCTGGACTCAGTGggcaatggacttcacaattg
pri-mir-9-5 smiFISH_17	CCTCCTAAGTTTCGAGCTGGACTCAGTGtccaagtggagaagttgcta
pri-mir-9-5 smiFISH_18	CCTCCTAAGTTTCGAGCTGGACTCAGTGctccgtgtcttgtgtaagaa
pri-mir-9-5 smiFISH_19	CCTCCTAAGTTTCGAGCTGGACTCAGTGcacaacagtaatcccaggat
pri-mir-9-5 smiFISH_20	CCTCCTAAGTTTCGAGCTGGACTCAGTGgataacaactcgcttccaat
pri-mir-9-5 smiFISH_21	CCTCCTAAGTTTCGAGCTGGACTCAGTGctcatacagctagataacca
pri-mir-9-5 smiFISH_22	CCTCCTAAGTTTCGAGCTGGACTCAGTGgaggcagtttttactttcgg
pri-mir-9-5 smiFISH_23	CCTCCTAAGTTTCGAGCTGGACTCAGTGcagccagagtttcgatgagc
pri-mir-9-5 smiFISH_24	CCTCCTAAGTTTCGAGCTGGACTCAGTGagctactgacacaggctata
pri-mir-9-5 smiFISH_25	CCTCCTAAGTTTCGAGCTGGACTCAGTGtctaacatgttgtgtagcca
pri-mir-9-5 smiFISH_26	CCTCCTAAGTTTCGAGCTGGACTCAGTGttatagcaccaagtgaagcc
pri-mir-9-5 smiFISH_27	CCTCCTAAGTTTCGAGCTGGACTCAGTGcccgcgagctatgagaaata
pri-mir-9-5 smiFISH_28	CCTCCTAAGTTTCGAGCTGGACTCAGTGtttgtttgcttgagttaccg
pri-mir-9-5 smiFISH_29	CCTCCTAAGTTTCGAGCTGGACTCAGTGttcctccacaaggaaaagga
pri-mir-9-5 smiFISH_30	CCTCCTAAGTTTCGAGCTGGACTCAGTGccagtctattcgtctatttg
pri-mir-9-5 smiFISH_31	CCTCCTAAGTTTCGAGCTGGACTCAGTGttcacgtgtcagtttcagta
pri-mir-9-5 smiFISH_32	CCTCCTAAGTTTCGAGCTGGACTCAGTGaatagtgagcacgtgggttc
pri-mir-9-5 smiFISH_33	CCTCCTAAGTTTCGAGCTGGACTCAGTGcaaaggcgaccgagctgaaa
pri-mir-9-5 smiFISH_34	CCTCCTAAGTTTCGAGCTGGACTCAGTGatttctgtttgggatgccag
pri-mir-9-5 smiFISH_35	CCTCCTAAGTTTCGAGCTGGACTCAGTGgctctctctatatgtcctaa

Table C.7: **smiFISH Probe sequences for pri-mir-9-5.**

Probe sequence name	**Probe sequence (5’ to 3’)**
her6 smiFISH_1	CCTCCTAAGTTTCGAGCTGGACTCAGTGtccatgatatcggcaggcat
her6 smiFISH_2	CCTCCTAAGTTTCGAGCTGGACTCAGTGgaccggagaagaggagtttt
her6 smiFISH_3	CCTCCTAAGTTTCGAGCTGGACTCAGTGtgggtttatcaggtgtagtg
her6 smiFISH_4	CCTCCTAAGTTTCGAGCTGGACTCAGTGagactttctgtgttccgaag
her6 smiFISH_5	CCTCCTAAGTTTCGAGCTGGACTCAGTGtcttttctccataatgggtt
her6 smiFISH_6	CCTCCTAAGTTTCGAGCTGGACTCAGTGtttcgttgattctcgctctt
her6 smiFISH_7	CCTCCTAAGTTTCGAGCTGGACTCAGTGattaacgttttcagctgacc
her6 smiFISH_8	CCTCCTAAGTTTCGAGCTGGACTCAGTGatcttttttcagagcatcca
her6 smiFISH_9	CCTCCTAAGTTTCGAGCTGGACTCAGTGggctttctcaagtttagagt
her6 smiFISH_10	CCTCCTAAGTTTCGAGCTGGACTCAGTGtcactgtcatctccaggatg
her6 smiFISH_11	CCTCCTAAGTTTCGAGCTGGACTCAGTGcgctgcatgtttctgagatg
her6 smiFISH_12	CCTCCTAAGTTTCGAGCTGGACTCAGTGtttagggcagcggtcatttg
her6 smiFISH_13	CCTCCTAAGTTTCGAGCTGGACTCAGTGttcccaagaacggtgggatc
her6 smiFISH_14	CCTCCTAAGTTTCGAGCTGGACTCAGTGattcactgaatccagctcgg
her6 smiFISH_15	CCTCCTAAGTTTCGAGCTGGACTCAGTGaaccgggtaacctcgttcat
her6 smiFISH_16	CCTCCTAAGTTTCGAGCTGGACTCAGTGtgttaacccccttcacatgtg
her6 smiFISH_17	CCTCCTAAGTTTCGAGCTGGACTCAGTGcgttgatctgtgtcatgcag
her6 smiFISH_18	CCTCCTAAGTTTCGAGCTGGACTCAGTGtgctgtgttggatagttcat
her6 smiFISH_19	CCTCCTAAGTTTCGAGCTGGACTCAGTGtgaaggatggatgaggaggc
her6 smiFISH_20	CCTCCTAAGTTTCGAGCTGGACTCAGTGgggatctgaaccatgggttg
her6 smiFISH_21	CCTCCTAAGTTTCGAGCTGGACTCAGTGcgctaagaggcacaacgttg
her6 smiFISH_22	CCTCCTAAGTTTCGAGCTGGACTCAGTGgtcaaattggaggatgagcc
her6 smiFISH_23	CCTCCTAAGTTTCGAGCTGGACTCAGTGccatatactttagttgcgtc
her6 smiFISH_24	CCTCCTAAGTTTCGAGCTGGACTCAGTGttgccggcacaagctggaaa
her6 smiFISH_25	CCTCCTAAGTTTCGAGCTGGACTCAGTGcaaaaaggcgaactgtccgt
her6 smiFISH_26	CCTCCTAAGTTTCGAGCTGGACTCAGTGttggagcaaaggcagcgttg
her6 smiFISH_27	CCTCCTAAGTTTCGAGCTGGACTCAGTGtagactggaataacagggcc
her6 smiFISH_28	CCTCCTAAGTTTCGAGCTGGACTCAGTGgaaccggtgtgttggaattg
her6 smiFISH_29	CCTCCTAAGTTTCGAGCTGGACTCAGTGaaacggagtctgacgtgacg

Table C.8: **smiFISH Probe sequences for *her6*.**

Probe sequence name	**Probe sequence (5’ to 3’)**
her9 smiFISH_1	CCTCCTAAGTTTCGAGCTGGACTCAGTGcttctccatattatcggctg
her9 smiFISH_2	CCTCCTAAGTTTCGAGCTGGACTCAGTGcagcaataggtgatgctgtc
her9 smiFISH_3	CCTCCTAAGTTTCGAGCTGGACTCAGTGgcttgtcaggagtatgagat
her9 smiFISH_4	CCTCCTAAGTTTCGAGCTGGACTCAGTGagactttctatgctcgctgg
her9 smiFISH_5	CCTCCTAAGTTTCGAGCTGGACTCAGTGgcttttccatgattggcttt
her9 smiFISH_6	CCTCCTAAGTTTCGAGCTGGACTCAGTGaaggctctcgttgattctcg
her9 smiFISH_7	CCTCCTAAGTTTCGAGCTGGACTCAGTGgaatgagagtcttcagctgc
her9 smiFISH_8	CCTCCTAAGTTTCGAGCTGGACTCAGTGgctatcttttttaagagcat
her9 smiFISH_9	CCTCCTAAGTTTCGAGCTGGACTCAGTGtctccaatttagagtgtctg
her9 smiFISH_10	CCTCCTAAGTTTCGAGCTGGACTCAGTGgtcatctccagaatatcagc
her9 smiFISH_11	CCTCCTAAGTTTCGAGCTGGACTCAGTGtaaattgcgcaggtgcttga
her9 smiFISH_12	CCTCCTAAGTTTCGAGCTGGACTCAGTGaaggctgcgctcatctgaac
her9 smiFISH_13	CCTCCTAAGTTTCGAGCTGGACTCAGTGtacttgctgaggacgtttgt
her9 smiFISH_14	CCTCCTAAGTTTCGAGCTGGACTCAGTGcatgcactcgttgaatcctg
her9 smiFISH_15	CCTCCTAAGTTTCGAGCTGGACTCAGTGagagaaatcgagtcacctcg
her9 smiFISH_16	CCTCCTAAGTTTCGAGCTGGACTCAGTGtctgacctctgtattcactc
her9 smiFISH_17	CCTCCTAAGTTTCGAGCTGGACTCAGTGacaggtggttaagaagtcgc
her9 smiFISH_18	CCTCCTAAGTTTCGAGCTGGACTCAGTGcatcatctgtcccatacaac
her9 smiFISH_19	CCTCCTAAGTTTCGAGCTGGACTCAGTGcaggctgagggtagttcatg
her9 smiFISH_20	CCTCCTAAGTTTCGAGCTGGACTCAGTGagccaaatgagcctgttgag
her9 smiFISH_21	CCTCCTAAGTTTCGAGCTGGACTCAGTGgaagctgcacgtgaagaggc
her9 smiFISH_22	CCTCCTAAGTTTCGAGCTGGACTCAGTGccgttgatgggtaacgttga
her9 smiFISH_23	CCTCCTAAGTTTCGAGCTGGACTCAGTGtgagtttggaacccattgag
her9 smiFISH_24	CCTCCTAAGTTTCGAGCTGGACTCAGTGtggtgagaccgcttctgaag
her9 smiFISH_25	CCTCCTAAGTTTCGAGCTGGACTCAGTGagctggaatcctccaaagac
her9 smiFISH_26	CCTCCTAAGTTTCGAGCTGGACTCAGTGaaaagcaaactgtccgtccg
her9 smiFISH_27	CCTCCTAAGTTTCGAGCTGGACTCAGTGcaaacgctgggttggggata
her9 smiFISH_28	CCTCCTAAGTTTCGAGCTGGACTCAGTGaatgaccggagttgtggcag
her9 smiFISH_29	CCTCCTAAGTTTCGAGCTGGACTCAGTGcgcttgcgtttgcgtacaag
her9 smiFISH_30	CCTCCTAAGTTTCGAGCTGGACTCAGTGactggcgttgacagtcactg
her9 smiFISH_31	CCTCCTAAGTTTCGAGCTGGACTCAGTGcattccctggacaggagatg
her9 smiFISH_32	CCTCCTAAGTTTCGAGCTGGACTCAGTGcttgtggaacgcccgagaag
her9 smiFISH_33	CCTCCTAAGTTTCGAGCTGGACTCAGTGctgacaccaacgggactgac

Table C.9: **smiFISH Probe sequences for *her9*.**

Probe sequence name	**Probe sequence (5' to 3')**
neurog1 smiFISH_1	CCTCCTAAGTTTCGAGCTGGACTCAGTGtgctcttaaccctcaatcag
neurog1 smiFISH_2	CCTCCTAAGTTTCGAGCTGGACTCAGTGaagataatggcacgcgtctg
neurog1 smiFISH_3	CCTCCTAAGTTTCGAGCTGGACTCAGTGatcctgcagatagtttgtgt
neurog1 smiFISH_4	CCTCCTAAGTTTCGAGCTGGACTCAGTGgagatgcttgaggttttgca
neurog1 smiFISH_5	CCTCCTAAGTTTCGAGCTGGACTCAGTGtgataaccttattggtgggc
neurog1 smiFISH_6	CCTCCTAAGTTTCGAGCTGGACTCAGTGcggagtatacgatctccatt
neurog1 smiFISH_7	CCTCCTAAGTTTCGAGCTGGACTCAGTGtagtcacagcttgaggtttc
neurog1 smiFISH_8	CCTCCTAAGTTTCGAGCTGGACTCAGTGatcatccgtgtgcgaaaagg
neurog1 smiFISH_9	CCTCCTAAGTTTCGAGCTGGACTCAGTGtggagacgcaggtggttttc
neurog1 smiFISH_10	CCTCCTAAGTTTCGAGCTGGACTCAGTGttcttcttcacgacgtgcac
neurog1 smiFISH_11	CCTCCTAAGTTTCGAGCTGGACTCAGTGttaaggttgtgcatcctgtt
neurog1 smiFISH_12	CCTCCTAAGTTTCGAGCTGGACTCAGTGgcttctcaaagcatccaatg
neurog1 smiFISH_13	CCTCCTAAGTTTCGAGCTGGACTCAGTGtgtgtcgtcaggaaacgcag
neurog1 smiFISH_14	CCTCCTAAGTTTCGAGCTGGACTCAGTGgagtctcaattttggtcagc
neurog1 smiFISH_15	CCTCCTAAGTTTCGAGCTGGACTCAGTGatgtagttgtgagcgaagcg
neurog1 smiFISH_16	CCTCCTAAGTTTCGAGCTGGACTCAGTGatccggatggtctccgaaag
neurog1 smiFISH_17	CCTCCTAAGTTTCGAGCTGGACTCAGTGatgaagacgacgaggatgcc
neurog1 smiFISH_18	CCTCCTAAGTTTCGAGCTGGACTCAGTGggtctgagttgcagtaagac
neurog1 smiFISH_19	CCTCCTAAGTTTCGAGCTGGACTCAGTGtatccaaaatcgtccatggc
neurog1 smiFISH_20	CCTCCTAAGTTTCGAGCTGGACTCAGTGctgtacactacgtcggtttg
neurog1 smiFISH_21	CCTCCTAAGTTTCGAGCTGGACTCAGTGagatgctaggcacgaagttg
neurog1 smiFISH_22	CCTCCTAAGTTTCGAGCTGGACTCAGTGaccgacatgagaacgcttaa
neurog1 smiFISH_23	CCTCCTAAGTTTCGAGCTGGACTCAGTGtacatacttctggagattct
neurog1 smiFISH_24	CCTCCTAAGTTTCGAGCTGGACTCAGTGagtaacagtggtcttacact
neurog1 smiFISH_25	CCTCCTAAGTTTCGAGCTGGACTCAGTGgttgtgctcttggttctaat
neurog1 smiFISH_26	CCTCCTAAGTTTCGAGCTGGACTCAGTGaggcaaacagtgaatacctа
neurog1 smiFISH_27	CCTCCTAAGTTTCGAGCTGGACTCAGTGtctcatttgcagactgtcat
neurog1 smiFISH_28	CCTCCTAAGTTTCGAGCTGGACTCAGTGcacgctccaaggaatgcaaa
neurog1 smiFISH_29	CCTCCTAAGTTTCGAGCTGGACTCAGTGattttctctaacggggttct
neurog1 smiFISH_30	CCTCCTAAGTTTCGAGCTGGACTCAGTGcaatctgccttgctttttaa
neurog1 smiFISH_31	CCTCCTAAGTTTCGAGCTGGACTCAGTGtgacttttcaccttggacag
neurog1 smiFISH_32	CCTCCTAAGTTTCGAGCTGGACTCAGTGtcagctttatcgctctacaa
neurog1 smiFISH_33	CCTCCTAAGTTTCGAGCTGGACTCAGTGaactgatttttcacgctcgt
neurog1 smiFISH_34	CCTCCTAAGTTTCGAGCTGGACTCAGTGttcagtctattgtcacagcg
neurog1 smiFISH_35	CCTCCTAAGTTTCGAGCTGGACTCAGTGcataaggccagatctttgtc
neurog1 smiFISH_36	CCTCCTAAGTTTCGAGCTGGACTCAGTGtttcttcgggtcaaaataca
neurog1 smiFISH_37	CCTCCTAAGTTTCGAGCTGGACTCAGTGggatcagttggacagatgag
neurog1 smiFISH_38	CCTCCTAAGTTTCGAGCTGGACTCAGTGcatgagagctggttaactgt
neurog1 smiFISH_39	CCTCCTAAGTTTCGAGCTGGACTCAGTGagaaaagtggtgggaaagcc
neurog1 smiFISH_40	CCTCCTAAGTTTCGAGCTGGACTCAGTGcgtacaaacatgtttgcacc

Table C.10: **smiFISH Probe sequences for *neurog1*.**

Probe sequence name	**Probe sequence (5’ to 3’)**
atoh1a smiFISH_1	CCTCCTAAGTTTCGAGCTGGACTCAGTGggtgaaggggatttctttac
atoh1a smiFISH_2	CCTCCTAAGTTTCGAGCTGGACTCAGTGagatttctcttcttacacct
atoh1a smiFISH_3	CCTCCTAAGTTTCGAGCTGGACTCAGTGtacagggacggatattttgc
atoh1a smiFISH_4	CCTCCTAAGTTTCGAGCTGGACTCAGTGttgggaggaaagtttgtggc
atoh1a smiFISH_5	CCTCCTAAGTTTCGAGCTGGACTCAGTGatccattctgttggtttgtg
atoh1a smiFISH_6	CCTCCTAAGTTTCGAGCTGGACTCAGTGttcaaccacctctcttgtat
atoh1a smiFISH_7	CCTCCTAAGTTTCGAGCTGGACTCAGTGaagctcgaatgctggacgtc
atoh1a smiFISH_8	CCTCCTAAGTTTCGAGCTGGACTCAGTGtgaagttggaggcagaggac
atoh1a smiFISH_9	CCTCCTAAGTTTCGAGCTGGACTCAGTGcgtactttgacaggactctt
atoh1a smiFISH_10	CCTCCTAAGTTTCGAGCTGGACTCAGTGgttggtggatttgctggatg
atoh1a smiFISH_11	CCTCCTAAGTTTCGAGCTGGACTCAGTGttcaatccgtgcattcttcg
atoh1a smiFISH_12	CCTCCTAAGTTTCGAGCTGGACTCAGTGcaaaggctgggatgacactg
atoh1a smiFISH_13	CCTCCTAAGTTTCGAGCTGGACTCAGTGttggagagtttcttgtcgtt
atoh1a smiFISH_14	CCTCCTAAGTTTCGAGCTGGACTCAGTGttgatgtagatctgggccat
atoh1a smiFISH_15	CCTCCTAAGTTTCGAGCTGGACTCAGTGctgtagtaagtcggacaggg
atoh1a smiFISH_16	CCTCCTAAGTTTCGAGCTGGACTCAGTGtttctaacacgttggcatgc
atoh1a smiFISH_17	CCTCCTAAGTTTCGAGCTGGACTCAGTGctcgtactggtacgggtaac
atoh1a smiFISH_18	CCTCCTAAGTTTCGAGCTGGACTCAGTGgtcttgctccatgaaagagt
atoh1a smiFISH_19	CCTCCTAAGTTTCGAGCTGGACTCAGTGcgaaccagacttgctcgttc
atoh1a smiFISH_20	CCTCCTAAGTTTCGAGCTGGACTCAGTGaccgaggcgagtctttactg
atoh1a smiFISH_21	CCTCCTAAGTTTCGAGCTGGACTCAGTGcgagtgaggcgagaactctc
atoh1a smiFISH_22	CCTCCTAAGTTTCGAGCTGGACTCAGTGgtttcgtctgagtcactgaa
atoh1a smiFISH_23	CCTCCTAAGTTTCGAGCTGGACTCAGTGgacagctcgtcttcactctg
atoh1a smiFISH_24	CCTCCTAAGTTTCGAGCTGGACTCAGTGtttcttaaaaagcgcggcgc
atoh1a smiFISH_25	CCTCCTAAGTTTCGAGCTGGACTCAGTGtgacacgattgagggacagt
atoh1a smiFISH_26	CCTCCTAAGTTTCGAGCTGGACTCAGTGaagcaacccattacaaagcc
atoh1a smiFISH_27	CCTCCTAAGTTTCGAGCTGGACTCAGTGtctattggcgtcagacacaa
atoh1a smiFISH_28	CCTCCTAAGTTTCGAGCTGGACTCAGTGgtcaatgtcacgagaaggca
atoh1a smiFISH_29	CCTCCTAAGTTTCGAGCTGGACTCAGTGttatgcatgacattcgagcc
atoh1a smiFISH_30	CCTCCTAAGTTTCGAGCTGGACTCAGTGgtctcaaaacagtttgcgga
atoh1a smiFISH_31	CCTCCTAAGTTTCGAGCTGGACTCAGTGaggtccatgacaatcatgtg
atoh1a smiFISH_32	CCTCCTAAGTTTCGAGCTGGACTCAGTGgtcgcaaattgttacagggt
atoh1a smiFISH_33	CCTCCTAAGTTTCGAGCTGGACTCAGTGccgtttttaaagtgcaagt
atoh1a smiFISH_34	CCTCCTAAGTTTCGAGCTGGACTCAGTGgaatatttctcaagcctacg
atoh1a smiFISH_35	CCTCCTAAGTTTCGAGCTGGACTCAGTGcacatcatttcttgttccat
atoh1a smiFISH_36	CCTCCTAAGTTTCGAGCTGGACTCAGTGattggcattacttctacata
atoh1a smiFISH_37	CCTCCTAAGTTTCGAGCTGGACTCAGTGgcagaacacaacttctttgc
atoh1a smiFISH_38	CCTCCTAAGTTTCGAGCTGGACTCAGTGatagcttacttcagctacag
atoh1a smiFISH_39	CCTCCTAAGTTTCGAGCTGGACTCAGTGgctcatttccaatgtaacac

Table C.11: **smiFISH Probe sequences for *atoh1a*.**

Probe sequence name	**Probe sequence (5' to 3')**
ascl1a smiFISH_1	CCTCCTAAGTTTCGAGCTGGACTCAGTGtctcgacctgttctgagttg
ascl1a smiFISH_2	CCTCCTAAGTTTCGAGCTGGACTCAGTGgttcacgtgtggcttcaatg
ascl1a smiFISH_3	CCTCCTAAGTTTCGAGCTGGACTCAGTGaaagttttctttggactgcc
ascl1a smiFISH_4	CCTCCTAAGTTTCGAGCTGGACTCAGTGtccatttcgcggagtcaaaa
ascl1a smiFISH_5	CCTCCTAAGTTTCGAGCTGGACTCAGTGtggtttacgcttatttccat
ascl1a smiFISH_6	CCTCCTAAGTTTCGAGCTGGACTCAGTGaagcaagcaggtggcatgaa
ascl1a smiFISH_7	CCTCCTAAGTTTCGAGCTGGACTCAGTGtgagttggatgctctgagag
ascl1a smiFISH_8	CCTCCTAAGTTTCGAGCTGGACTCAGTGttgacgcggacttgttgctg
ascl1a smiFISH_9	CCTCCTAAGTTTCGAGCTGGACTCAGTGattgagtcttcttttgcacc
ascl1a smiFISH_10	CCTCCTAAGTTTCGAGCTGGACTCAGTGgggtaagctgtagccgaaac
ascl1a smiFISH_11	CCTCCTAAGTTTCGAGCTGGACTCAGTGcaaagccgttgttcacaagc
ascl1a smiFISH_12	CCTCCTAAGTTTCGAGCTGGACTCAGTGctccattgggaacgtgttcg
ascl1a smiFISH_13	CCTCCTAAGTTTCGAGCTGGACTCAGTGctttgctcatcttcttgttg
ascl1a smiFISH_14	CCTCCTAAGTTTCGAGCTGGACTCAGTGgtagttttgggagatggtgg
ascl1a smiFISH_15	CCTCCTAAGTTTCGAGCTGGACTCAGTGgccatagagttcatgtcatt
ascl1a smiFISH_16	CCTCCTAAGTTTCGAGCTGGACTCAGTGctcatccgatgagtatgagg
ascl1a smiFISH_17	CCTCCTAAGTTTCGAGCTGGACTCAGTGttgttcttctggactcagag
ascl1a smiFISH_18	CCTCCTAAGTTTCGAGCTGGACTCAGTGaaaccagttggtgaagtcca
ascl1a smiFISH_19	CCTCCTAAGTTTCGAGCTGGACTCAGTGagtttccttttacgaacgct
ascl1a smiFISH_20	CCTCCTAAGTTTCGAGCTGGACTCAGTGccaagcgagtgctgatattt
ascl1a smiFISH_21	CCTCCTAAGTTTCGAGCTGGACTCAGTGttgtgttcttggaggacatc
ascl1a smiFISH_22	CCTCCTAAGTTTCGAGCTGGACTCAGTGcttggctctttgacactcgg
ascl1a smiFISH_23	CCTCCTAAGTTTCGAGCTGGACTCAGTGatagatttcttgggcgagtg
ascl1a smiFISH_24	CCTCCTAAGTTTCGAGCTGGACTCAGTGggtgtcgtggaaagtctttt
ascl1a smiFISH_25	CCTCCTAAGTTTCGAGCTGGACTCAGTGcaacgtttgcttgttgttgt
ascl1a smiFISH_26	CCTCCTAAGTTTCGAGCTGGACTCAGTGagggcaaaccttctttgatt
ascl1a smiFISH_27	CCTCCTAAGTTTCGAGCTGGACTCAGTGagagttttgagagaggggtc
ascl1a smiFISH_28	CCTCCTAAGTTTCGAGCTGGACTCAGTGaattcgccaagttggaagca
ascl1a smiFISH_29	CCTCCTAAGTTTCGAGCTGGACTCAGTGcggcaggctataggtcaaaa
ascl1a smiFISH_30	CCTCCTAAGTTTCGAGCTGGACTCAGTGctgcgatgcatttgagacta
ascl1a smiFISH_31	CCTCCTAAGTTTCGAGCTGGACTCAGTGctccacaaccgtaaagggaa
ascl1a smiFISH_32	CCTCCTAAGTTTCGAGCTGGACTCAGTGttgagcattacactcctcta
ascl1a smiFISH_33	CCTCCTAAGTTTCGAGCTGGACTCAGTGataagacacgttggtgctga
ascl1a smiFISH_34	CCTCCTAAGTTTCGAGCTGGACTCAGTGtcactctgagtcacattaca
ascl1a smiFISH_35	CCTCCTAAGTTTCGAGCTGGACTCAGTGggacaatagctgcataacct
ascl1a smiFISH_36	CCTCCTAAGTTTCGAGCTGGACTCAGTGgagacacaaaacacctccgt
ascl1a smiFISH_37	CCTCCTAAGTTTCGAGCTGGACTCAGTGgcaaagtggaacaggcagtg
ascl1a smiFISH_38	CCTCCTAAGTTTCGAGCTGGACTCAGTGtgactgcaacacgtaaagca
ascl1a smiFISH_39	CCTCCTAAGTTTCGAGCTGGACTCAGTGttagcaggatggcttatcac
ascl1a smiFISH_40	CCTCCTAAGTTTCGAGCTGGACTCAGTGttggcattcattaagagcgc

Table C.12: **smiFISH Probe sequences for *ascl1a*.**

sgRNA number	**Target sequence 5'-3' with PAM**	**CRISPRscan Primer 5'-3'**
1	GGACGGGTGGCCGGAGGGGT**TGG**	taatacgactcactataGGACGGGTG GCCGGAGGGGTgttttagagctagaa
2	AAGGTGGGGTGATGCCTTTC**AGG**	taatacgactcactataGGGGTGGGGTG ATGCCTTTCgttttagagctagaa
3	AGAGAGAGGGAGACTTGGGA**GGG**	taatacgactcactataGGAGAGAGGG AGACTTGGGAgttttagagctagaa

Table C.13: **CRISPR/Cas9 Target sequences for pre-mir-9-1 and primers used to generate respective sgRNA.** PAM region is highlighted in red.

Parameter	**Value**
h_1	3.0
h_2	3.0
h_3	3.0
p_1 (low, high)	2.0, 10.0
p_2	1.0
p_3	1.0
α_X	0.1
μ_X	0.1
α_h	6.0
μ_h	2.0
α_Y	30.0
β_Y	100.0
μ_Y	10.0

Table C.14: **The parameter values used for the mathematical model.** Highlighted in grey are the additional parameters used for the extended model. α_X, μ_X, α_h and μ_h parameters were used in both, adaptation and extended, models.