THESIS

USING FINITE STATE PROJECTION AND FISHER INFORMATION TO IMPROVE SINGLE-CELL EXPERIMENT DESIGN TO GAIN BETTER UNDERSTANDING OF DUSP1 TRANSCRIPTION DYNAMICS

Submitted by

Joshua A. Cook

School of Biomedical Engineering

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Summer 2023

Master's Committee:

Advisor: Brian Munsky

Edwin Chong
Soham Ghosh

ABSTRACT

USING FINITE STATE PROJECTION AND FISHER INFORMATION TO IMPROVE SINGLE-CELL EXPERIMENT DESIGN TO GAIN BETTER UNDERSTANDING OF DUSP1 TRANSCRIPTION DYNAMICS

Many recent studies have combined fluorescent biochemical labels, single-cell microscopy, and discrete stochastic modeling to understand and predict how organisms react to environmental changes to control gene expression. The experimental data used in these studies is often collected using intuitively-designed applications of techniques such as single-cell immunnocytochemistry (ICC) to measure protein expression and transport or single-molecule Fluorescence *in situ* Hybridization (smFISH) to measure the number and position of transcribed mRNA. Once collected, these single-cell data are then analyzed using discrete stochastic models, often based on the framework of the Chemical Master Equation (CME), which can be solved using the Finite State Projection (FSP) algorithm. Unfortunately, these experiments can be expensive and labor intensive to perform, primarily due to long imaging and image analysis times, and it is not clear how these experiments must be designed to obtain the most information when their results are later analyzed using the FSP techniques. The recently discovered Finite State Projection based Fisher information Matrix (FSP-FIM) provides a potential and practical solution to this experiment design challenge by providing direct estimates for how well any potential experiment should be expected to constrain parameters for a given model or set of models. In this report, we examine this challenge of experiment design in the situation where multiple different types of experiments (i.e., ICC and smFISH) are possible, for different time points, for different numbers of measurements per time point, for different environmental inputs, and for different assumed models and combinations of unknown parameters. We extend the previous FSP-FIM theory to address these multiple challenges, and we introduce new computational tools in the form of advances to the Stochastic System Identification

Toolkit (in Mathworks Matlab™) that allow users to easily and efficiently compute the FSP and FIM for each of these circumstances. Using experimental smFISH data, we demonstrate the effectiveness of the FSP tools to quantitatively reproduce the single-cell transcription dynamics of the Dual Specific Phosphatase 1 (DUSP1) gene under stimulation by Dexamethasone (Dex), and we show how the FSP-FIM can be used to design optimal combinations of ICC and smFISH to further improve quantification of this gene regulatory process, including predicting the optimal allocation of measurement times to obtain the most amount of information from each experiment. To probe the generality of our results, these FSP and FSP-FIM analyses are conducted for different models, under different assumptions on known and unknown parameters, and under different drug dosage regimens. The approach developed in this work is expected to have substantial impact on how computational models can be employed to improve the selection and design of future single-cell experiments.

ACKNOWLEDGEMENTS

ChatGPT-3.5 was used to assist in rewording the paragraphs in the paper. All paragraphs were originally written by the author (J.A.C.) before being rephrased using ChatGPT-3.5. The final selection of wording in the paper was determined by J.A.C.

TABLE OF CONTENTS

ABSTRACT ii
ACKNOWLEDGEMENTS iv

Chapter 1 Introduction 1

Chapter 2 Methods 3
2.1 DUSP1 Gene Regulation Models 3
2.2 Single-Cell Experiments 6
2.3 Stochastic Systems Identification Toolkit 7
2.4 Defining Discrete Stochastic Models 7
2.5 Defining the Chemical Master Equation 9
2.6 Generating Stochastic Simulations of Gene Regulation 11
2.7 Solving the CME using the Finite State Projection algorithm 11
2.8 Computing Likelihoods of Single-Cell Data 13
2.9 Metropolis Hastings to Samples Parameter Posteriors 14
2.10 Sensitivity Analysis of the CME Using the FSP 17
2.11 Computing the Fisher Information Matrix 18
2.12 Using the FIM to Optimize Experiment Designs 22
2.13 Triptolide Repression Model 23

Chapter 3 Results 27
3.1 Existing measurements of DUSP1 transcription regulation are quantitatively reproduced using discrete stochastic models. 27
3.2 Upon parameter uncertainty quantification using Metropolis Hastings sampling, the parameters ($k_{ON}, k_{OFF}, k_r, g_r$) are tightly constrained. 32
3.3 The Fisher Information Matrix quantitatively predicts the variation in Maximum Likelihood Estimates among different simulated data set replicates. . 35
3.4 Measuring GR and DUSP1 expression simultaneously is more informative than measuring these quantities in the same number cells divided into separate GR and DUSP1 experiments. 39
3.5 FIM analysis suggests the an optimal triptolide administration time for the EGRNT and SGRS models to identify a DUSP1 repression model. 50
3.6 Bayesian Information Criterion Suggest the SGRS is a Better Model of DUSP1 Regulation 53

Chapter 4 Discussion 55

Bibliography 59

Appendix A Cell Visualization and Experiment Verification 65

Appendix B Stochastic Systems Identification Toolkit tutorial 71

B.1 Model Set Up 71
B.2 Running SSA trajectories 72
B.3 Finite State Projection Calculation 73
B.4 Sensitivity calculation and Analysis 73
B.5 Fisher Information Computation 75
B.6 Data loading and fitting 75

Chapter 1

Introduction

Understanding the mechanisms of gene regulation is crucial for comprehending cellular responses to environmental changes and various stresses. To gain insight into these mechanisms, researchers often employ a discrete stochastic modeling approach [1–5]. Subsequently, these models can be fitted to experimental data to determine parameters and evaluate their ability to replicate the variability observed in the experimental data. Such data is frequently generated from single-cell experiments, such as Single-Molecule in situ Hybridization (smFISH) [6], immunostaining methods like immunocytochemistry (ICC) [7], or live cell imaging techniques [8–10]. Traditionally, the experimental approach to uncover gene regulation mechanisms involves collecting experimental data, constructing a model, fitting the model, assessing the reproducibility of simulated data, and then restarting the cycle with newly designed experiments [11]. Initially, experimenters often have limited information on how to design the experiments and only gain insights into better experiment design options after going through this cycle. Here, we present an alternative approach to enhancing experiment design, showcasing the utilization of the Fisher Information Matrix (FIM) as a metric for optimal experiment design [12–15]. The FIM finds practical utility in experiment design and in assessing the identifiability of deterministic ordinary differential equation (ODE) models in biological systems [16].

In previous studies, the Fisher Information Matrix (FIM) has been applied to stochastic modeling of gene expression, where a numerical method was developed to compute the FIM using the Linear Noise Approximation of the Chemical Master Equation (CME) [5, 17]. This approach allowed for investigating the impact of bulk measurements and time series measurements on parameter uncertainties. Furthermore, ongoing research has focused on approximating the FIM using moment closure techniques [13, 18], demonstrating its effectiveness in designing optimal optogenetic experiments [13]. Derivation of the Fisher Information Matrix (FIM) from the Finite State Projection (FSP) [19] demonstrated an effective approach for constructing the FIM in time-varying

and non-linear models. This approach is commonly referred to as FSP-FIM [12,19]. Moreover, the FSP-FIM framework enables the incorporation of distortion operators, providing the user with the ability to account for measurement distortions [15]. The FIM based experiment design approach aims to provide insights into the development of improved experiments and creating an optimal experiment design, thereby reducing the cost and time required for fitting stochastic processes like gene expression. In this study, we will apply this design approach to the gene expression of Dual Specific Phosphatase 1 (DUSP1).

The DUSP1 gene exhibits involvement in the inflammatory response at the cellular level and displays activity in immune cells, fibroblasts, endothelial cells, lung epithelial cells, HeLa cells, and other cell types. DUSP1 encodes the Mitogen Kinase Phosphatase 1 (MKP1) protein, which acts as a crucial regulator within the DUSP1/MKP1 pathway [20]. This pathway exerts control over the anti-inflammatory response of the cells by dephosphorylating P38 and JNK, which are two branches of the mitogen-activated protein kinase (MAPK) pathway [21]. Both the strength and duration of MAPK activation significantly influence cellular function [22], necessitating precise regulation. Dysregulation of MAPK activity has been implicated in the development of various pathologies, including neurodegenerative diseases, diabetes, cancer, and inflammation [20].

Chapter 2

Methods

2.1 DUSP1 Gene Regulation Models

The activation of the DUSP1 gene primarily occurs through the binding of glucocorticoids (GC) to glucocorticoid receptors (GR) [23], resulting in their translocation into the nucleus. Once in the nucleus, GR induces or represses the transcription of thousands of genes by directly and indirectly binding to GR DNA response elements [21]. One of the genes activated by GR is DUSP1 by triggering the transcription of the gene [20], and the up-regulation of DUSP1 by GCs is believed to play a crucial role in the suppression of inflammation [23]. In-order to understand the temporal dynamics of the DUSP1 gene activation, a discrete stochastic model is created to fit data generated by smFISH and ICC experiments. The smFISH experiment captures the transcription dynamics, while ICC experiment captures the dynamics of GR translocation. Two models are suggested for fitting the data. The first model is an Explicit GR Nuclear Translocation (EGRNT) Model. The mechanisms of the ERGNT model are shown in figure 2.1.

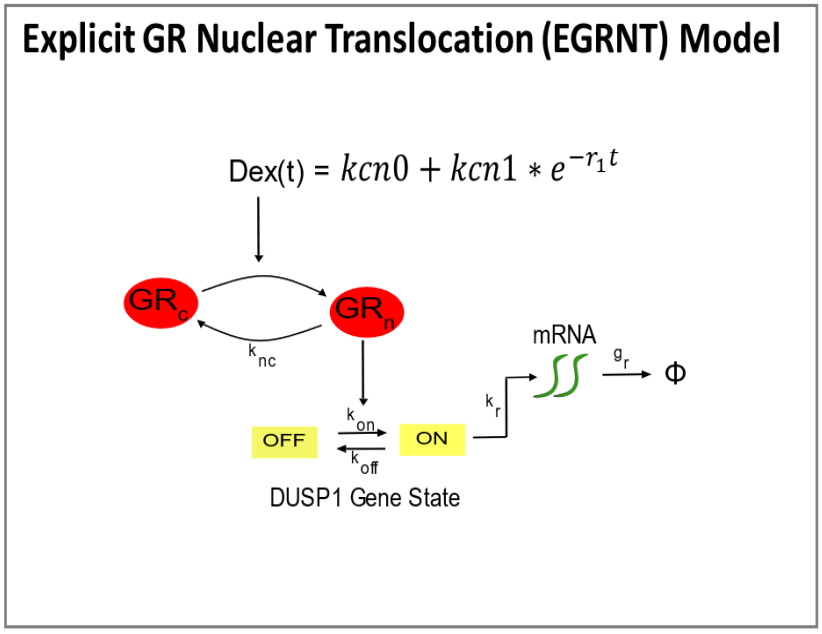

Figure 2.1: Explicit GR Nuclear Translocation Model. The Explicit GR Nuclear Translocation (EGRNT) model utilizes three species: nuclear GR, gene state, and mRNA, to depict the regulatory mechanisms of the DUSP1 gene following stimulation by Dexamethasone (Dex). In this model, the nuclear GR can either enter or exit the nucleus. The gene state can exist in two states: ON or OFF. When the gene state is in the on state, mRNA can be transcribed. The mRNA, in turn, can undergo degradation or exit the nucleus.

The EGRNT model incorporates the mechanisms of GR translocation to the nucleus and transcription dynamics of mRNA. The term GR_n represents the number of GR within the nucleus. The parameters k_{cn0} and k_{cn1} deal with the rates of nuclear GR translocation. The term $e^{-r_1 t}$ represents the time-varying input of Dexamethasone (Dex) stimulation. k_{nc} is the rate of GR leaving the nucleus or degrading. k_{ON} is the rate of gene activation. k_{OFF} is the rate of gene deactivation. k_r is the rate of mRNA production. The last parameter, g_r, is the rate of mRNA leaving the nucleus or degrading.

The second model is a Simplified GR Signal (SGRS) model. This model simplifies the EGRNT model by representing the nuclear translocation of GR as a time varying signal. This is represented

by Equation 2.1 with the starting nuclear GR being equal to $\frac{K_{cn0}}{k_{nc}}$.

$$\frac{d}{dt}GR_n = k_{cn0} + k_{cn1}e^{-r_1 t} - kncGR_n. \tag{2.1}$$

The number of GR_n at any time is then represented by another variable U to account for the starting nuclear GRs. This is represented by Equation 2.2.

$$U = GR_n - \frac{K_{cn0}}{k_{nc}}. \tag{2.2}$$

This equation is then written as a rate to get Equation 2.3.

$$\frac{d}{dt}U = K_{cn0} + k_{cn1}e^{-r_1 t} - k_{nc}(U + \frac{K_{cn0}}{k_{nc}}). \tag{2.3}$$

Equation 2.3 is further simplified to get Equation 2.4.

$$\frac{d}{dt}U = k_{cn1}e^{-r_1 t} - k_{nc}U. \tag{2.4}$$

Integrating equation 2.4 gives Equation 2.5.

$$U(t) = \frac{k_{cn1}}{k_{nc} - r_1}e^{-k_{nc}t}(e^{k_{nc}-r_1 t} - 1). \tag{2.5}$$

Plugging Equation 2.5 into Equation 2.2 give the time varying signal represented in the model as $I_{GR}(t)$. $I_{GR}(t)$ is represented by Equation 2.6.

$$I_{GR}(t) = \frac{k_{cn0}}{k_{nc}} + \frac{k_{cn1}}{r_1 - k_{nc}}e^{-k_{nc}t}(1 - e^{r_1 - k_{nc}t}). \tag{2.6}$$

SGRS model's mechanism is shown in figure 2.2.

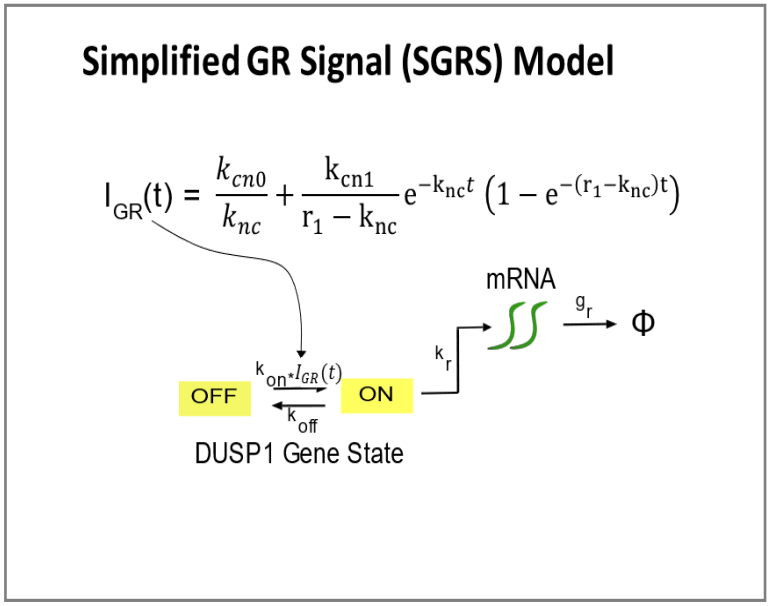

Figure 2.2: Simplified GR Signal Model. The Simplified GR Signal (SGRS) model describes the activation of the DUSP1 gene using a simplified version of the GR nuclear translocation mechanism, where I_{GR} is a deterministic time-varying input derived as the mean of the nuclear GR level from the EGRNT model. The two remaining species in the SGRS model are the gene state and the nuclear mRNA. In this model, when the gene state is in the ON state, mRNA can be transcribed. The mRNA can then either leave the nucleus or undergo decay.

The SGRS model reduces the number of species by one because the nuclear translocation of GR is represented by the time-varying signal. The remaining species in the model are the number of active allels and the DUSP1 mRNA.

2.2 Single-Cell Experiments

The experimental data utilized in this study was obtained through single-molecule fluorescence in situ hybridization (smFISH), which enables the measurement of mRNA expression at different time points within a cell population [6]. Differences in gene expression among individual cell populations were quantified and used for parameter fitting in the models [24, 25]. In smFISH, a

primary probe specific to the target species (DUSP1 mRNA in this case) is designed. Subsequently, a secondary probe carrying a fluorescent dye binds to the primary probe, resulting in fluorescence upon attachment. Microscopic imaging is then conducted on the cells, and the captured images undergo processing. Live cell imaging was not used in this study; instead, different cell populations were measured at each time point, enabling the measurement of mRNA expression over time with a large number of cells. The utilized data for the modeling procedure includes smFISH data capturing the expression of DUSP1 over a three-hour period after stimulation with Dex. These images were processed using the BIG-FISH image processing package [26] and the cell segmentation tool called Cellpose [27]. The processed data was subsequently saved in a CSV file format, which can be easily incorporated into MATLAB code.

2.3 Stochastic Systems Identification Toolkit

The Stochastic Systems Identification Toolkit (SSIT) served as the primary tool for data analysis in this study. SSIT is a Mathworks Matlab™)-based software developed by Munsky et al. that facilitates modeling and parameter fitting of discrete stochastic systems. It offers various functionalities, including setting up discrete models, executing Stochastic Simulation Algorithm (SSA) trajectories, calculating Finite State Projection (FSP), conducting sensitivity analysis, loading and fitting data, and calculating the Fisher Information Matrix (FIM). The SSIT played a crucial role throughout the analysis and parameter fitting of the DUSP1 models. Detailed instructions and guidance on the key functions of the SSIT can be found in the appendix, providing a helpful tutorial for users.

2.4 Defining Discrete Stochastic Models

Gene regulation is a complex process that is influenced by various factors. The interactions and dynamics involved in gene expression introduce inherent stochasticity, leading to the propagation of noise within genetic circuits [1]. To understand these intricate connections, stochastic models are employed to fit gene regulation data. The activation and inactivation of genes can be effectively

described as a discrete Markov process [2], allowing for the design and fitting of multiple models to experimental data. The stochastic nature of gene regulation, characterized by random activation and deactivation of gene states [28], provides the motivation for adopting a discrete Markov process for modeling purposes. In order to capture the stochastic behavior of DUSP1 mRNA expression, we utilize a discrete stochastic modeling approach. This approach involves monitoring every possible state and transition, keeping track of the count for each species of interest. Each state can be represented by a vector $x_i = [s_1, s_2, ..., s_N]_i^T$, where x_i corresponds to the i^{th} state and s_N represents the species of interest. For the EGRNT model, the vector x_i can be defined as $[c_i, g_i, m_i]$, where c_i, g_i, and m_i denote the counts of nuclear glucocorticoid receptors, gene states, and mRNA, respectively, for the i^{th} state. Similarly, for the SGRS model, x_i is defined as $[g_i, m_i]$, encompassing the gene state and mRNA count. The transition from the current state, x_i, to the next state, x_j, is denoted as $x_j = x_i + \psi_v$, where ψ_v represents the stoichiometry associated with the reaction occurring at the j^{th} step. This transition reflects the changes in species counts resulting from the reactions taking place within the system.

An example of how the models are generated and the analysis is performed using the SSIT codes will be demonstrated for the EGRNT model of DUSP1 gene regulation. In the codes, any line ending in "..." is treated as a single line by the SSIT codes. The initial model setup is generated using the following codes:

```
Model = SSIT;
Model.species = {'x1';'x2';'x3'};  % GRnuc, geneOn, dusp1
Model.initialCondition = [0;0;0];
```

The example code above initializes the SSIT program and defines the number of species and the initial conditions of the model. In this model, the parameters x1, x2, and x3 represent the numbers of nuclear glucocorticoid receptors, active DUSP1 alleles, and DUSP1 mRNA, respectively. The following line of code sets the propensity function of the model.

```
Model.propensityFunctions = {'(kcn0+kcn1*IDex)';'knc*x1';...
                  'kon*x1*(2-x2)';'koff*x2';'kr*x2';'gr*x3'};
```

In this description, the six propensity functions correspond to size model reactions: the translocation of GR to the nucleus ($w_1 = (kcn0 + kcn1 * IDex(t))$); the translocation of GR to the cytoplasm ($w_2 = (knc * x1)$; the activation one of the alleles of DUSP1 ($w_3 = kon * x1 * (2 - x2)$; deactivation of one DUSP1 alleles ($w_4 = koff * x2$); transcription of a mRNA molecule ($w_5 = kr * x2$); and degradation or mRNA leaving the nucleus ($w_6 = gr * x3$).

The following lines of code set the time varying input for Dex stimulation and the initial parameter value guesses.

```
Model.inputExpressions = {'IDex','(t>0)*exp(-r1*t)'};
Model.parameters = ({'koff',0.14;'kon',0.01;'kr',1;'gr',0.01;...
                     'kcn0',0.01;'kcn1',0.1;'knc',1;'r1',0.01});
```

The stoichiometry matrix is then set by the code:

```
Model.stoichiometry = [ 1,-1, 0, 0, 0, 0;...
                        0, 0, 1,-1, 0, 0;...
                        0, 0, 0, 0, 1,-1];
Model.fspOptions.initApproxSS = true;
```

Here, the stoichiometry matrix describes the specific effect that each of the six reactions has on the system: $\psi_1 = [1, 0, 0]^T$ describes the increase of one molecule of nuclear GR; $\psi_2 = [-1, 0, 0]^T$ describes the decrease of one molecule of nuclear GR; $\psi_3 = [0, 1, 0]^T$ describes the activation of one DUSP1 allele; $\psi_4 = [0, -1, 0]^T$ describes the deactivation of one DUSP1 allele; $\psi_5 = [0, 0, 1]^T$ describes the increase of one molecule of nuclear mRNA;$\psi_6 = [0, 0, -1]^T$ describes the decrease of one molecule of nuclear mRNA.

2.5 Defining the Chemical Master Equation

The Chemical Master Equation (CME) serves as the foundation for constructing stochastic models of the DUSP1 gene. By utilizing the defined variables x_i and ψ_i, the propensity function

$w_v(x_i)dt$ is established as the probability of the v^{th} reaction occurring in the i^{th} state within a time step dt. Consequently, the CME can be expressed by Equation 2.7. [29, 30].

$$\frac{d}{dt}p(x_i : t) = \sum_{v=1}^{N_i}[w_v(x_i - \psi_v, t)p(x_i - \psi_v, t) - w_v(x_i, t)p(x_i, t)]. \tag{2.7}$$

Looking into all possible variations of x_i at the time point t gives the probability mass vector $P = [p(x_1, t), p(x_2, t), ..., p(x_i, t)]^T$ and grants the ability to write the CME in the matrix shown in form Equation 2.8 [31].

$$\frac{d}{dt}p(t) = Ap(t). \tag{2.8}$$

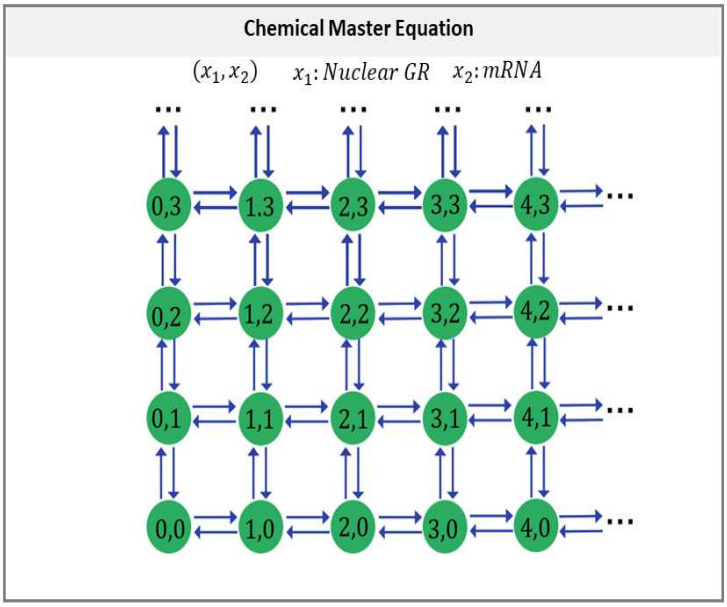

Figure 2.3: Visualization of the states and stoichiometries for the Chemical Master Equation description of the EGRNT model. The CME enumerates all possible combinations of the species mRNA and nuclear GR. Arrows in the diagram represent stoichiometries for reactions that can occur in the system (e.g., upward pointing arrows indicate mRNA transcription, while downward-pointing arrows represent loss of nuclear mRNA). Similarly, right- and left-pointing arrows in the horizontal direction indicate GR translocation into and out of the the nucleus, respectively.

The CME has an infinite number of states for many different models, including the EGRNT model. This is illustrated in figure 2.3 above. This makes the solution of the CME unsolvable so another method must be used.

2.6 Generating Stochastic Simulations of Gene Regulation

The stochastic simulation of gene regulation grants the ability to simulate trajectories and distributions for a given model. This aids in fitting the parameters of the gene regulation model so the species population distribution generated by the model matches experimental data.

In order to generate these simulations, the SSA is used. The SSA is a Monte Carlo simulation of the CME [1,32]. The SSA is run by using the discrete stochastic modeling method as mentioned in the methods above to create trajectories of how the state of the species evolves over time. The species are tracked in x_i up to the i^{th} state of the trajectory. The time until the next reaction, dt, is defined by $dt = \frac{1}{|w|_1} \log \frac{1}{r_1}$. Where r_1 is a random number chosen from a uniform distribution between 0 and 1. The reaction that occurs is chosen by first selecting another random number, r_2, from a uniform distribution between 0 and 1. A value of k is chosen such that: $\sum_v^{k-1} w_v(x) \leq r_2|w|_1 \leq \sum_v^k w_v(x)$. The state change is updated with the associated ϕ_v stochiometry change and the next reaction and time step is repeated until reaching the i^{th} state of the trajectory [1, 32].

2.7 Solving the CME using the Finite State Projection algorithm

The solution of the CME is crucial in successfully modeling gene regulation and using tools to optimize experimental design. Unfortunately, in many cases such as the DUSP1 regulation model, the CME is unsolvable due to having an infinite number of states. To address this issue, the Finite State Projection (FSP) is used as an approximation of the CME solution, which is solvable [31]. This representation of the FSP is shown in figure 2.4 below.

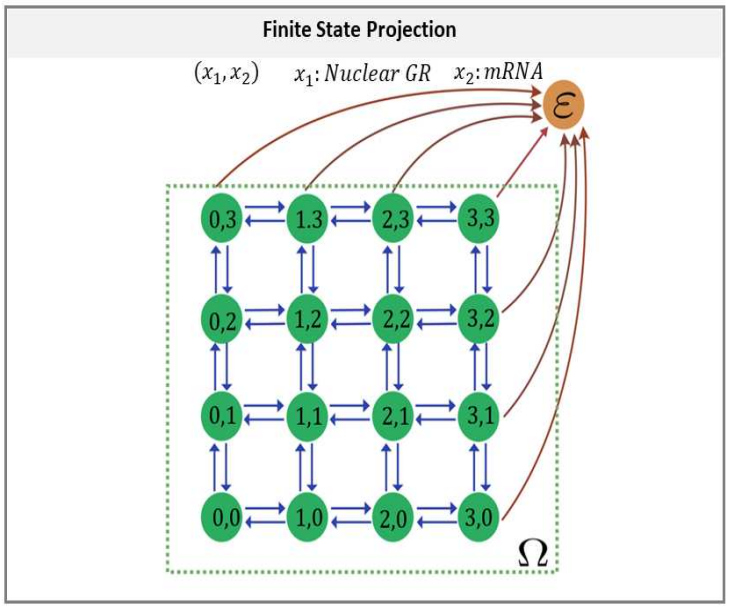

Figure 2.4: Visualization of the Finite State Projection (FSP) truncation of the CME for the EGRNT model. The original CME lists possible combinations of the mRNA and nuclear GR species as in Figure 2.3. Arrows shown in the diagram represent reactions that can occur in the system. Under the FSP truncation, after reaching the corresponding threshold for the number of mRNA or nuclear GR molecules, the remaining states are collected into an absorbing sink state denoted by ϵ, resulting in a finite dimensional approximation of the CME (Equation 2.10) that can be analyzed using standard ODE solvers.

The FSP is created by truncating the CME into a finite set of states, X_J and a set of low probability states, $X_{J'}$ [12, 19]. Using the matrix form of the CME above and the definition of $p_J(t) \equiv p(X_J; t)$, the matrix form of the CME split into two separate states shown in Equation 2.9.

$$\frac{d}{dt}\begin{pmatrix} P_J \\ P_{J'} \end{pmatrix} = \begin{pmatrix} A_{JJ} & A_{JJ'} \\ A_{J'J} & A_{J'J'} \end{pmatrix} \begin{pmatrix} P_J(t) \\ P_{J'}(t) \end{pmatrix} . \tag{2.9}$$

The low probability set $p_J(t)$ is then set to be the sink state $g(t)$ and the CME equation is redefined as:

$$\frac{d}{dt}\begin{pmatrix} P_{FSP} \\ g(t) \end{pmatrix} = \begin{pmatrix} A_{jj} & 0 \\ -1^T A_{jj} & 0 \end{pmatrix} \begin{pmatrix} P_{FSP}(t) \\ g(t) \end{pmatrix} . \tag{2.10}$$

This method allows for the calculation of the error which is given by Equation 2.11.

$$\left| \begin{pmatrix} P_J \\ P_{J'} \end{pmatrix} - \begin{pmatrix} P_{FSP} \\ 0 \end{pmatrix} \right|_1 = \epsilon. \tag{2.11}$$

The lower bound of the true solution at all time greater than zero is given by Equation 2.12.

$$\begin{pmatrix} P_{FSP} \\ 0 \end{pmatrix} \leq \begin{pmatrix} P \\ P \end{pmatrix} . \tag{2.12}$$

The DUSP1 models had their FSP matrix generated using the SSIT tool. The FSP is also fit to the experimental data by loading the data into the model and a fit being run to this data-set multiple times through the use of a for loop. The SSIT commands to run the FSP fit are shown below.

```
Model.solutionScheme = 'FSP';
Model.fspOptions.fspTol = 1e-4;
[fspSoln,Model.fspOptions.bounds] = Model.solve;
```

The solution scheme is set to the FSP option with a selected error tolerance. When it finds the parameter set that maximizes the likelihood function, it replaces the current parameter set in the DUSP1 model.

2.8 Computing Likelihoods of Single-Cell Data

The calculation of the log-likelihood of the data from the FSP will be necessary to calculate the FIM. The tracked species in are represented as distributions over time and put into the matrix form $D_t \equiv [d_1, d_2, ...d_{N_c}]_t \in Z_{\geq 0}^{N_t \times N_c}$. Each cell in the population is defined by N_c while the

measurement time points are defined by D_t. The log-likelihood for a given parameter set, $\Theta = [\theta_1, \theta_2, ...\theta_k]$, is given by Equation 2.13 [12].

$$logL(D;\Theta) = \sum_{k=1}^{N_t} \sum_{i=1}^{N_c(k)} log(p(x_i = d_i; t_k, \theta). \tag{2.13}$$

In order to calculate the likelihood of the parameter set given the data, the SSIT code uses the following command:

```
Model.computeLikelihood
```

One way to estimate parameters from single cell data is to find the parameters set that maximize the likelihood function. This method has been used various times to fit model parameters from single cell data [4, 19, 33–35]. The maximum likelihood estimate is needed several times for fitting the DUSP1 regulation models. The SSIT tool provides a function that returns the parameters that return the maximum likelihood estimate. This function is called using the line:

```
Model.maximizeLikelihood([],fitOptions);
```

2.9 Metropolis Hastings to Samples Parameter Posteriors

Further parameter fitting is done by using Metropolis Hastings (MH) algorithm. The MH algorithm is used in various systems such as jet milling models [36], infection models [37] and biological systems [38] to estimate the system's parameters from sampling from the posterior distribution of the experimental data. The initial parameters used in the MH algorithm for fitting the DUSP1 models are the parameters found from the FSP fits. We use the MH algorithm to further fit the parameters of the model and get a covariance between each par of parameters which helps identify the best parameter set. The parameter set that maximizes the likelihood function is chosen and saved to the model. This is done using the SSIT in the following codes below.

```
MHOptions = struct('numberOfSamples',15000,'burnin',100,'thin',1,...
            'useFIMforMetHast',true,'suppressFSPExpansion',true);
```

```
[bestParsFound,~,mhResults] = Model.maximizeLikelihood(...
            [Model.parameters{...
            Model.fittingOptions.modelVarsToFit,2}]',...
            MHOptions,'MetropolisHastings');


Model.parameters(Model.fittingOptions.modelVarsToFit,2)...
        = num2cell(bestParsFound);
```

In the options for the Metropolis Hastings algorithm, 'burnin' is the number of samples discarded from the beginning of the MCMC chain to allow for relaxation away from the initial parameter guess; $'thin'$ is the number of intermediate MCMC samples that are discarded to reduce memory requirements and ensure greater independence between successive MCMC steps; $'useFIMforMetHast'$ is a flag instructing the algorithm to use the FIM for a more efficient MCMC sample proposal distribution; and $'suppressFSPExpansion'$ instructs the algorithm not to re-expand the FSP projection at each step, which may reduce accuracy but typically leads to a faster implementation.

This section is all done in a for loop to ensure the best parameter fit. The first line within the for loop of sets the options to create 15000 data point samples and prevent the FSP for expanding any more. The MH is ran and the maximum likelihood estimate is found in the command "$Model.maximizeLikelihood$." The last two lines of code save the parameter set that maximizes the log-likelihood functions. A plot of the Metropolis Hastings sample is generated using the SSIT command: "$Model.plotMHResults(mhResults)$." An example of the samples generated for the SGRS model is shown below in figure 2.5.

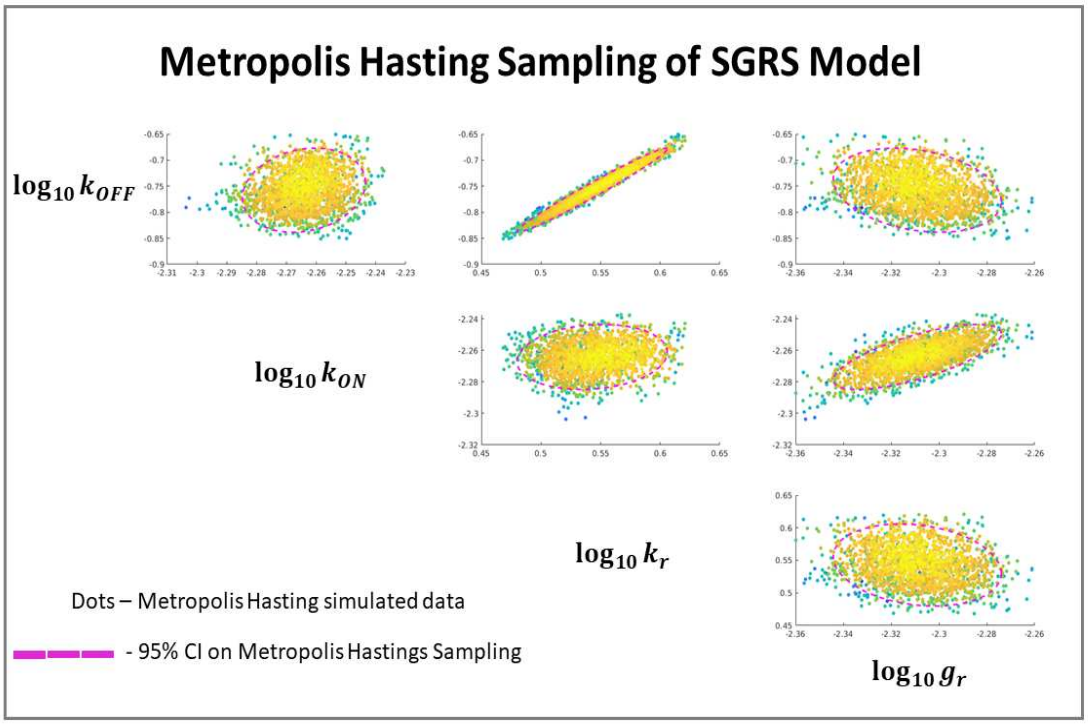

Figure 2.5: Metropolis Hastings (MH) analysis of parameter uncertainty given experimental data. The posterior parameter space of the SGRS model was sampled using MH to generate 15,000 parameter combinations depicted as individual dots. Colors denote the computed likelihood values from high (yellow) to low (blue). The dashed pink line denotes the 95% CI of the parameter estimates computed from the MH samples. Each panel shows the joint uncertainty of two parameters, and reveals that some combinations of parameters are highly correlated. For example, $\log_{10} k_{OFF}$ is linearly correlated to $\log_{10} k_r$, suggesting that the burst size k_r/k_{OFF} is tightly constrained by the data, but the actual values for each of these individual parameters is relatively uncertain.

2.10 Sensitivity Analysis of the CME Using the FSP

The sensitivity matrix is used in sensitivity analysis which identifies which parameters have the greatest influence on the resulting distribution a model creates [39–41]. The log-likelihood can be used to calculate the sensitivity matrix which is also able to be used for calculating the FIM. The Sensitivity matrix, S, is calculated by $S = \nabla_\theta p(X;\theta)$ [35]. By using the log-likelihood, the log of the sensitivity matrix can be calculated by Equation 2.14.

$$\nabla_\theta logp(X;\theta) = \begin{pmatrix} \frac{1}{p_0}\frac{\partial p_0}{\partial \theta_1} & \frac{1}{p_0}\frac{\partial p_0}{\partial \theta_2} & \cdots & \frac{1}{p_0}\frac{\partial p_0}{\partial \theta_{N_p}} \\ \frac{1}{p_1}\frac{\partial p_1}{\partial \theta_1} & \frac{1}{p_1}\frac{\partial p_1}{\partial \theta_2} & \cdots & \frac{1}{p_1}\frac{\partial p_1}{\partial \theta_{N_p}} \\ \vdots & \vdots & & \vdots \\ \frac{1}{p_N}\frac{\partial p_N}{\partial \theta_1} & \frac{1}{p_N}\frac{\partial p_N}{\partial \theta_2} & \cdots & \frac{1}{p_N}\frac{\partial p_N}{\partial \theta_{N_p}} \end{pmatrix}. \tag{2.14}$$

This is done through the SSIT tool with the following line of code:

```
Model.sensOptions.solutionMethod = 'finiteDifference';
Model.solutionScheme = 'fspSens';
Model.fspOptions.fspTol = 1e-6;
[sensSoln,Model.fspOptions.bounds] = Model.solve;
Model.makePlot(sensSoln,'marginals')
```

The solution method for calculating the sensitivity was chosen to be the finite difference method with the sensitivity being solved through the FSP solution scheme. The FSP tolerance of the solution was set to be $1e-6$. The second to last line is the command that make the SSIT tool run the sensitivity calculation. An example of the sensitivity plots for each parameter is shown for the EGRNT model for the distribution of mRNA (x3) is shown in figure 2.6 below.

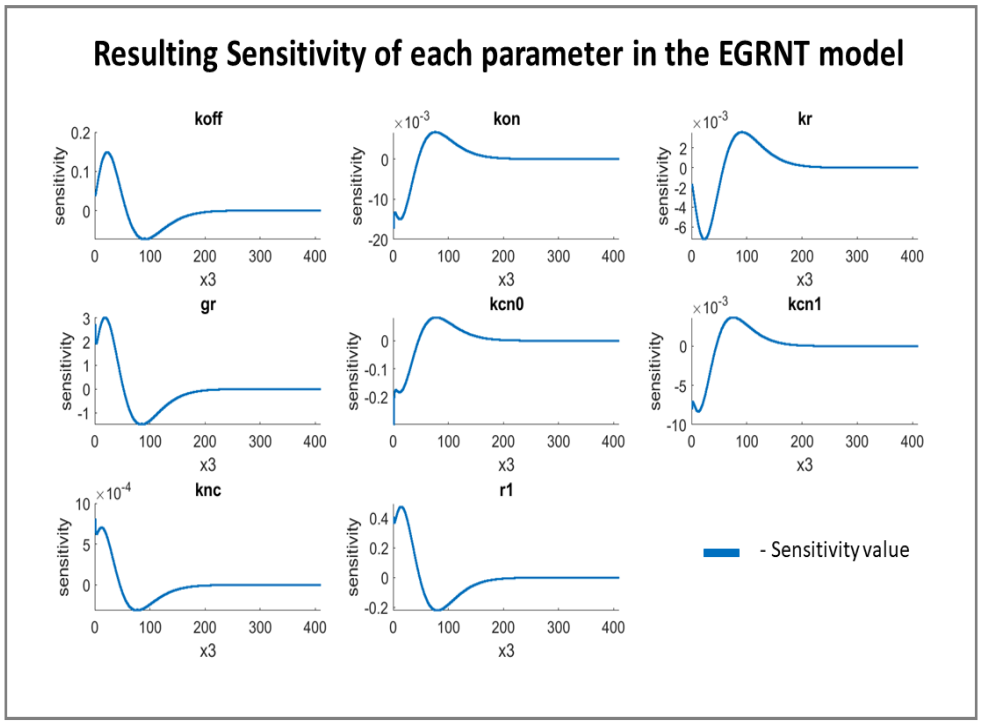

Figure 2.6: Sensitivity analysis for the Chemical Master Equation The sensitivity of each parameter's effect on the distribution of DUSP1 mRNA is shown as calculated using the FSP approach in the SSIT. The x axis shows the count of mRNA, (species 'x3'). The y-axis shows the sensitivity of the distribution at the specific time (t=180 min) to the parameter listed in the subplot title. For example, an increase in the mRNA degradation parameter 'gr' leads to more probability mass (positive sensitivity) at low mRNA levels and less probability mass (negative sensitivity) at higher values of mRNA.

2.11 Computing the Fisher Information Matrix

The main tool used to optimize the DUSP1 experiment design is the FIM. The determinant of the inverse of the FIM is equal to the determinant of the covariance matrix, $|COV|$ [12, 13]. Models with smaller $|COV|$ have less variance among the parameters which indicates a better model. The inverse of the determinant of the FIM will be the way the models are compared. With the sensitivity matrix defined in the previous section, the FIM can be defined by Equation 2.15.

$$I(\theta)_{ij} = N_c E\left\{(\tfrac{1}{p(x_l;\theta}})^2 S_{li} S_{lj}\right\}. \tag{2.15}$$

Taking the expectation of the previous equation gives Equation 2.16.

$$I(\theta)_{ij} = N_c \sum_{l=1}^{N} \frac{1}{p(x_l : \theta)} S_{li} S_{lj}. \tag{2.16}$$

This is the Fisher Information at a single time point. To get the Fisher Information of the entire experiment the $I(\theta)_{ij}$ at each time point, N_t, is summed together to get Equation 2.17.

$$I(\theta)_{ij} = \sum_{k=1}^{N_t} N_c(t_k) \sum_{l=1}^{N} \frac{1}{p(x_l : t_k : \theta)} S_{li}(t_k) S_{lj}(t_k). \tag{2.17}$$

The $|I(\theta)_{ij}^{-1}|$ of each experiment design could then be compared to each other to find the optimal experiment design. In the smFISH and ICC experiments, the variables in equation 2.17 are as follows. N_c is the number of cells measured for the experiment, t_k is the measurement time points, and S_{li}/S_{lj} are the sensitivities at the corresponding time points.

The FIM is calculated through the SSIT code by the following lines:

```
fims = Model.computeFIM(sensSoln.sens);
FIM = Model.evaluateExperiment(fims,Model.dataSet.nCells);
```

The sensitivity matrix that was previously calculated is loaded into the FIM calculation code. The FIM is then calculated again in the second line but, with the number of cells being measured in the experiment taken into account.

The FIM is verified by comparing it to the Maximum Likelihood Estimation (MLE). This is due to the Cramer-Rao bound (CRB) that states the inverse of the FIM provides a lower bound on the variance of any estimator of the model parameters [12, 42]. This lower bound is described by Equation 2.18.

$$\sqrt{N_c}(\hat{\theta} - \theta^*) \xrightarrow{dist} N(0, I(\theta^*)^{-1}). \tag{2.18}$$

This states as the number of measurements increases, the difference between the estimated parameters that maximize the likelihood of the data and the true parameters set approaches a normal

distribution with a mean of zero [42, 43]. An example of how the CRB property is used is shown in figure 2.7 below.

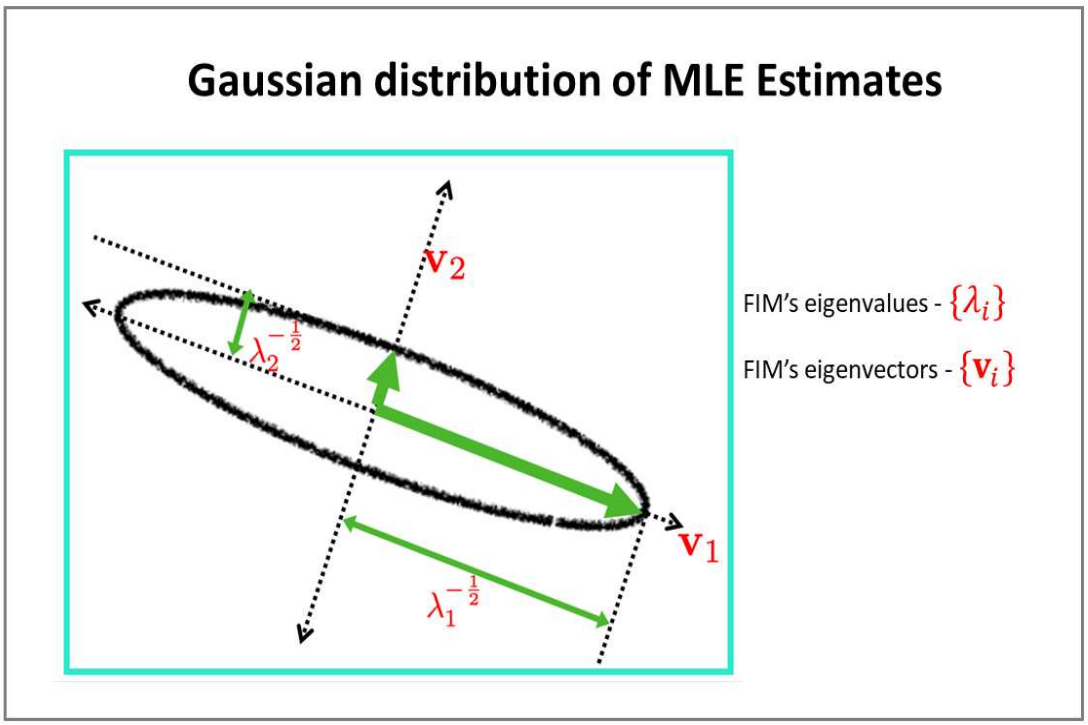

Figure 2.7: Schematic representation of the Cramer-Rao Lower Bound comparing the Eigenvectors and Eigenvalues of the FIM to the expected spread of maximum likelihood estimates. The CRB states that the inverse of the FIM provides a lower bound estimate for the uncertainty directions and magnitudes for the MLE. Here, λ_i are the eigenvalues of the FIM and v_i are the corresponding eigenvectors of the FIM; the largest and smallest eigenvalues of the FIM denote the directions in parameter space that are least and most uncertain, respectively.

Figure 2.7 shows the The FIM's eigenvalues, λ_i, and its eigenvectors, v_i, estimate the magnitudes and directions of uncertainty in MLE parameters (CRB).

To do a comparison of the FIM to the MLE, the MLE is generated by first generating 200 parameter estimates from SSA trajectories. The covariance of these parameters is then plotted and a 95% CI is calculated for the 200 data points. The ellipse generated from the MLE and the ellipse generated from the FIM is then compared.

This is done using the SSIT tool to run 200 SSA experiment simulations each having 100 individual SSA trajectories. The results of the simulations are saved into the model and the MLE is calculated and plotted with the parameter scatter plot. This is done using the codes below.

```
Model.ssaOptions.nSimsPerExpt = 100;
Model.ssaOptions.Nexp = 200;
Model.fspOptions.fspTol = 1e-8;  % Set FSP error tolerance.
Model.sampleDataFromFSP(FSPsoln,'EGRNT_model_sim_x2x3.csv');
```

The number of cell simulations per experiment is set to 100 and the number of simulated experiments is set to 200. The FSP tolerance for setting these experiments is set to $1e-8$. The generated data is then saved into the CSV file named $EGRNT\_model\_sim\_x2x3.csv$. The MLE is generated by the following codes:

```
par2Fit=length(find(Model.fittingOptions.modelVarsToFit));
fitOptions = optimset('Display','iter','MaxIter',200);

tmpModel = Model.loadData('EGRNT_model_sim_x2x3.csv',...
            {'x2',['exp',num2str(iExp),'_s2'];...
            'x3',['exp',num2str(iExp),'_s3']});

[MLE(1,:,iExp),fMLE(1,:,iExp)] = tmpModel.maximizeLikelihood(...
                                  x0,fitOptions);
```

First the parameters that will be fit to the simulated data is set in the first line. The simulated data is loaded in the third line and the parameters of the model that generated the data is fit for each experiment denoted by the variable $iExp$. In this case the data being generated and loaded are making the assumption that only species x2 and x3 are observed. Each experiment generates one parameter set that is plotted on a scatter plot with the MLE plotted with it.

2.12 Using the FIM to Optimize Experiment Designs

The FIM can be used as a metric to judge better experiment designs but the FIM can be used further to optimize each experiment. This is accomplished by using the FIM to determine the best way allocate the number of cells chosen to be measured at each time-point. This can be done because the experiment has a individual FIM for each time point. By looking through what time points gave the most information, these time-points can be prioritized over time-points with relatively low amounts of information obtained. In a typical smFISH experiment the experimenter chooses several time points ranging over a few hours and measures roughly the same number of cells at each time point. In the data set used for fitting the models there were twelve time points measured with 9639 cells measured over the entire experiment. By finding which time points give the most amount of information, the experimentalist can decide to measure more cells at those time points while ignoring time points that do not give much information. The cell allocation optimization was done in the following lines of code in the SSIT tool:

```
nTotal = sum(Model.dataSet.nCells);
nCellsOpt = Model.optimizeCellCounts(fims,nTotal,'TR[1:4]');
fimOpt = Model.evaluateExperiment(fims,nCellsOpt);
Model.plotMHResults(mhResults,{FIM,fimOpt});
```

The total number of cells in the experiment is loaded into the model with the FIM and a fitting option is given to optimize which times to allocate the cells to be measured. For the EGRNT model there are two cases for identifying the parameters. In Case 1, only the DUSP1 transcription parameters, $(k_{ON}, k_{OFF}, k_r, g_r)$ are being fit. This is due to an assumption that the GR nuclear translocation parameters are already known from literature or a previous experiment. Under Case 2, all of the parameters need to be identified. In the example code above, the experiment design is being optimized for Case 1 of the EGRNT model. For Case 2 where all of the parameters are being fit, $'[1:4]'$ would be used instead of $'TR[1:4]'$ on the second line of code. A plot is then generated with the MH samples, the 95% confidence Interval of the MH samples, the un-optimized FIM, and the optimized FIM.

2.13 Triptolide Repression Model

An additional mechanism involved in the regulation of the DUSP1 gene is its transcription repression in the presence of triptolide. Triptolide, an active component derived from the herb *Tripterygium wilfordii*, possesses anti-inflammatory and immunosuppressive properties [44]. Acting as a small-molecule inhibitor, triptolide hinders promoter DNA opening and transcription initiation, however it is not specific to DUSP1 [45]. Furthermore, triptolide has been employed for its anti-cancer effects and in the treatment of autoimmune diseases [45].

To understand the repression mechanism of triptolide, we will alter the existing model by incorporating a time input for triptolide. This mechanism is shown in figure 2.8.

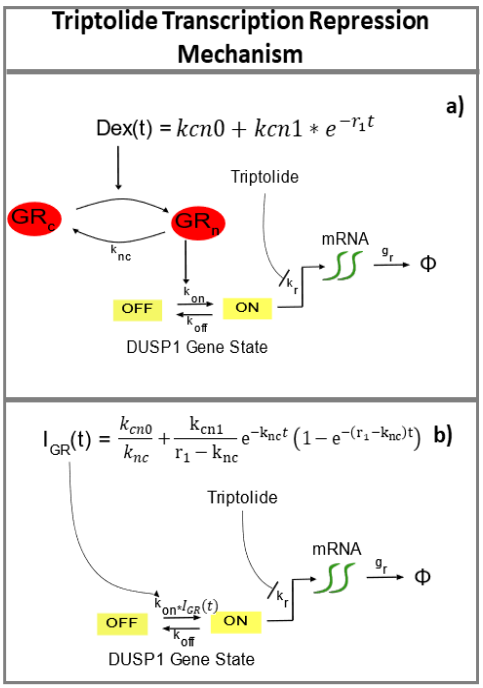

Figure 2.8: Extending the GR/DUSP1 models to include repression of transcription. The repression activation of triptolide acts by preventing the transcription of mRNA. This is modeled in a) for the EGRNT model and in b) for the SGRS model.

Incorporating triptolide into the DUSP1 models will create a difference in gene transcription overtime as seen in figure 2.9 for the EGRNT model and figure 2.10 for the SGRS model.

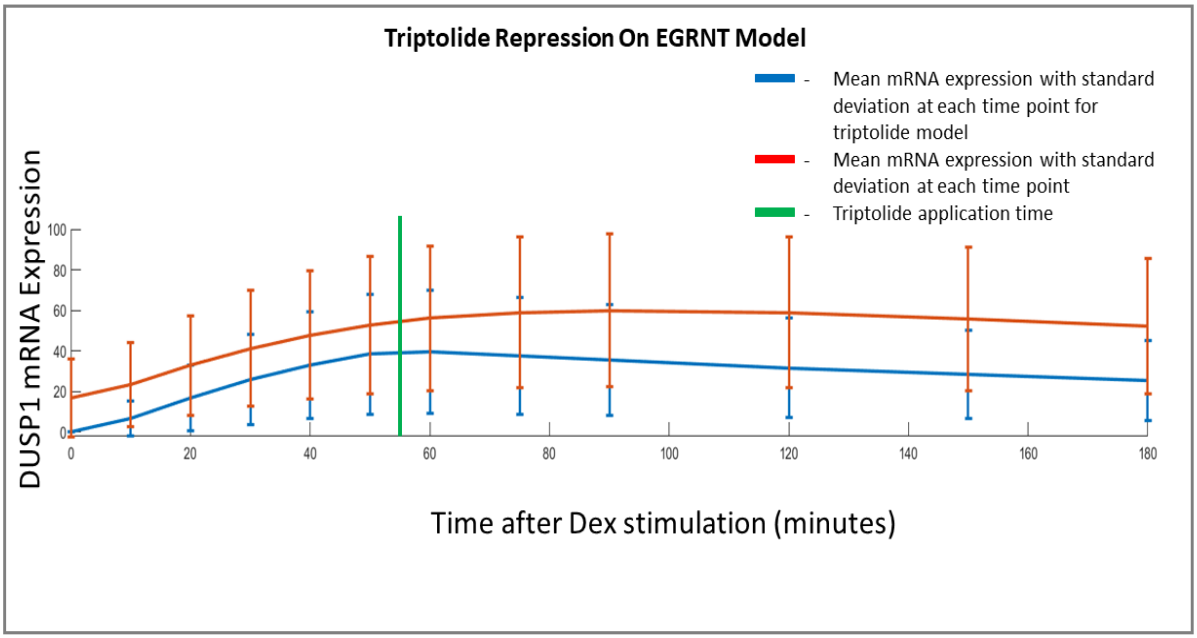

Figure 2.9: Average expression of DUSP1 mRNA over time predicted with the triptolide repression occurring at 55 minutes after Dex stimulation. The average DUSP1 expression in the EGRNT repression model is shown by the blue line. The base EGRNT model without triptolide is shown in red. The green line indicates when triptolide is administered in the model. The standard deviation is depicted by the error bars.

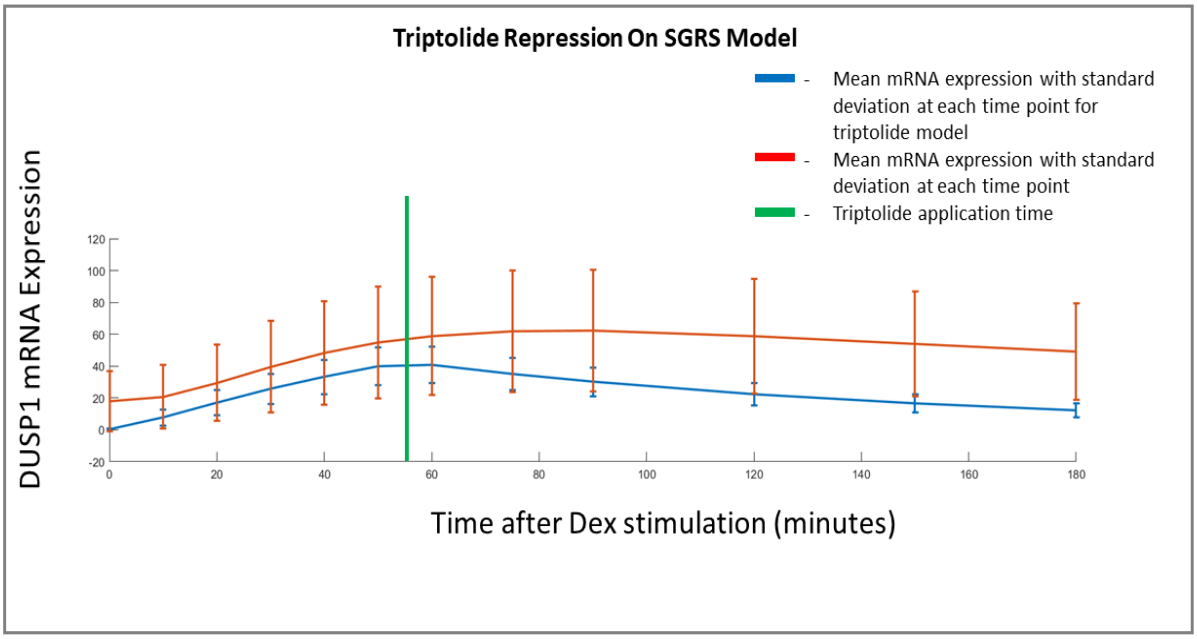

Figure 2.10: Average expression of DUSP1 mRNA over time predicted with the triptolide repression occurring at 55 minutes after Dex stimulation. The average DUSP1 expression in the SGRS repression model is shown by the blue line. The base SGRS model without triptolide is shown in red. The green line indicates when triptolide is administered in the model. The standard deviation is depicted by the error bars.

Before conducting the experiment, the experimental design will be optimized using the Fisher Information Matrix (FIM). The FIM will aid in determining the optimal timing for administering triptolide to the cells. By varying the timing of triptolide administration and calculating the FIM for each case, we will select the model that provides the most informative time point for the triptolide experiment. The code used to do this is shown below.

```
fims = ModelTrypt.computeFIM(sensSoln{iTpt}.sens);
FIM = ModelTrypt.evaluateExperiment(fims,
                ...ModelTrypt.dataSet.nCells);
expectedDetCov(iTpt) = det(FIM^(-1))
```

The sensitivity of the model with different triptolide application time points, listed as $iTpt$ in the code above, is calculated then used to get the FIM. The $|FIM^{-1}|$ is then plotted against the triptolide application time. The time that minimizes $|FIM|^{-1}$ is then chosen as the optimal triptolide experiment design.

Chapter 3

Results

3.1 Existing measurements of DUSP1 transcription regulation are quantitatively reproduced using discrete stochastic models.

The construction of the stochastic DUSP1 models relied on the SSIT code, as described in the methods section. While the EGRNT model's implementation was explicitly presented in the codes, both the SGRS and EGRNT models were devised and calibrated using the same method. Subsequently, the models were fitted to discern the optimal parameter values. Once well-defined parameters were obtained, the models successfully replicated the mRNA distribution observed in the experimental data. Figure 3.1 was generated by initially fitting both models using the FSP approach.

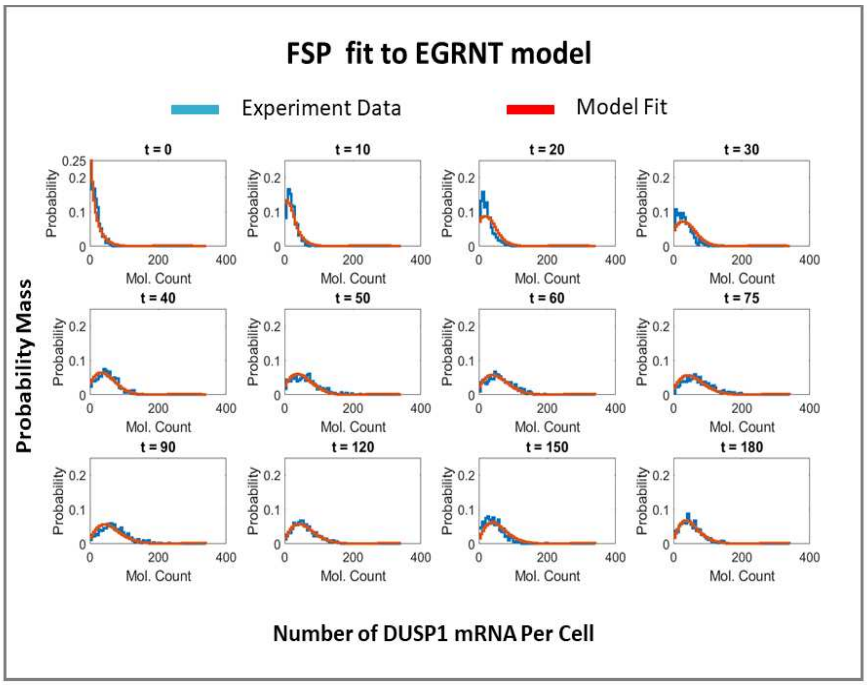

Figure 3.1: Fit of the EGRNT model to smFISH data at 12 time points. The EGRNT gene regulatory model is parameterized and calibrated to accurately capture the temporal distribution of the DUSP1 mRNA population in smFISH experimental data. Leveraging the FSP methodology, a comprehensive model fit is generated to depict the dynamics of DUSP1 mRNA across various time points, where the mRNA species is visually represented in blue. The fitted EGRNT model is visually showcased in red, demonstrating the model's aptitude in reproducing the observed experimental outcomes. the data shown here consist of 9,639 cells over 12 different time points.

The fitting procedure encompassed all temporally sampled instances within the data-set, ensuring a comprehensive alignment between the model and the observed data. Additionally, to validate the quality of the fits, the mean and variance of the fitted model were calculated and visualized along with the mean and variance of the original data-set. The graphical representation of these statistics is presented in Figure 3.2.

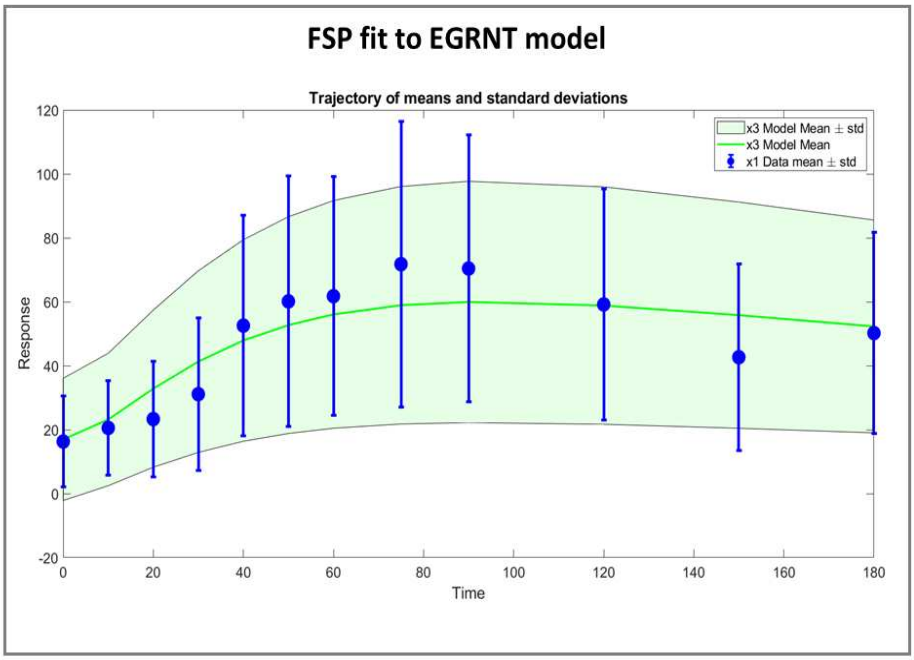

Figure 3.2: DUSP1 mean and variance over time after Dex-stimulation. The mRNA mean (circles) and standard deviation (bars) from the experiment data are shown in blue and compared to the mean (green line) and standard deviation (green shading) from the model.

The SGRS model was fit in a similar fashion to get model fits as shown in figure 3.3 and figure 3.4.

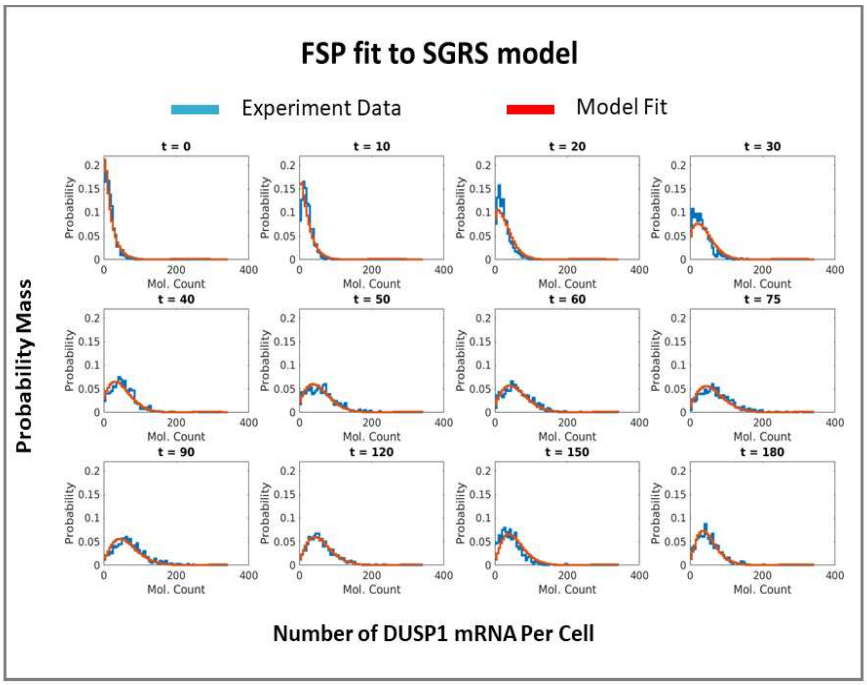

Figure 3.3: Fit of the SGRS model to smFISH data at 12 time points. The EGRNT gene regulatory model is parameterized and calibrated to accurately capture the temporal distribution of the DUSP1 mRNA population in smFISH experimental data. Leveraging the FSP methodology, a comprehensive model fit is generated to depict the dynamics of DUSP1 mRNA across various time points, where the mRNA species is visually represented in blue. The fitted EGRNT model is visually showcased in red, demonstrating the model's aptitude in reproducing the observed experimental outcomes. the data shown here consist of 9,639 cells over 12 different time points.

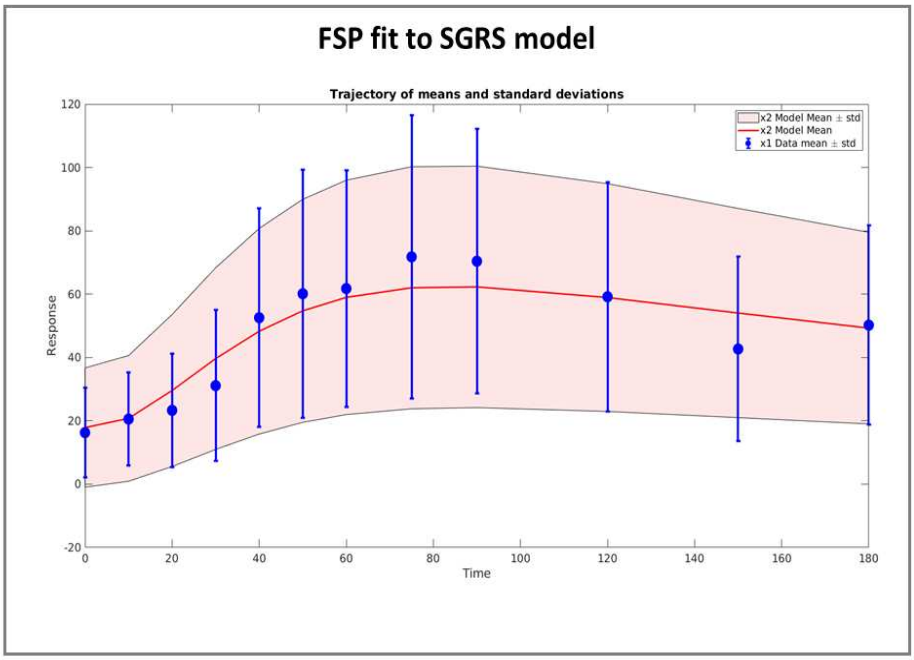

Figure 3.4: DUSP1 mean and variance over time after Dex-stimulation. The mRNA mean (circles) and standard deviation (bars) from the experiment data are shown in blue and compared to the mean (red line) and standard deviation (red shading) from the model.

Both models exhibit a remarkable adherence to the experimental data, supporting the identification of a robust parameter set achieved through the FSP fitting procedure. Notably, the standard deviation estimates generated by both models capture the inherent variation observed in the experimental data-set. Moreover, the concurrence between the two model fits further emphasizes their congruity. To fortify the validation of the parameter set selection, an additional fitting iteration is conducted utilizing the MH algorithm to ensure an optimal parameter set for both models.

3.2 Upon parameter uncertainty quantification using Metropolis Hastings sampling, the parameters ($k_{ON}, k_{OFF}, k_r, g_r$) are tightly constrained.

The parameter set is further fit to the data by performing a MH search on the posterior parameters given the distribution of the experimental data. The log-likelihood of each of the 15000 parameter sets generated by the MH search are calculated and the parameter set that maximizes the log-likelihood function is chosen as the new parameter set and saved in the model. The parameters ($k_{ON}, k_{OFF}, k_r, g_r$) are fit while the parameters ($k_{nc}, k_{cn0}, k_{cn1}, r_1$) remain the same. This is done because of the main interest in understanding the gene transcription dynamics. All of the parameters sets are plotted on a scatter plot and a 95% CI of the MH sample sets are shown. The plot of the EGRNT model can be seen in figure 3.5 and the SGRS model in figure 3.6.

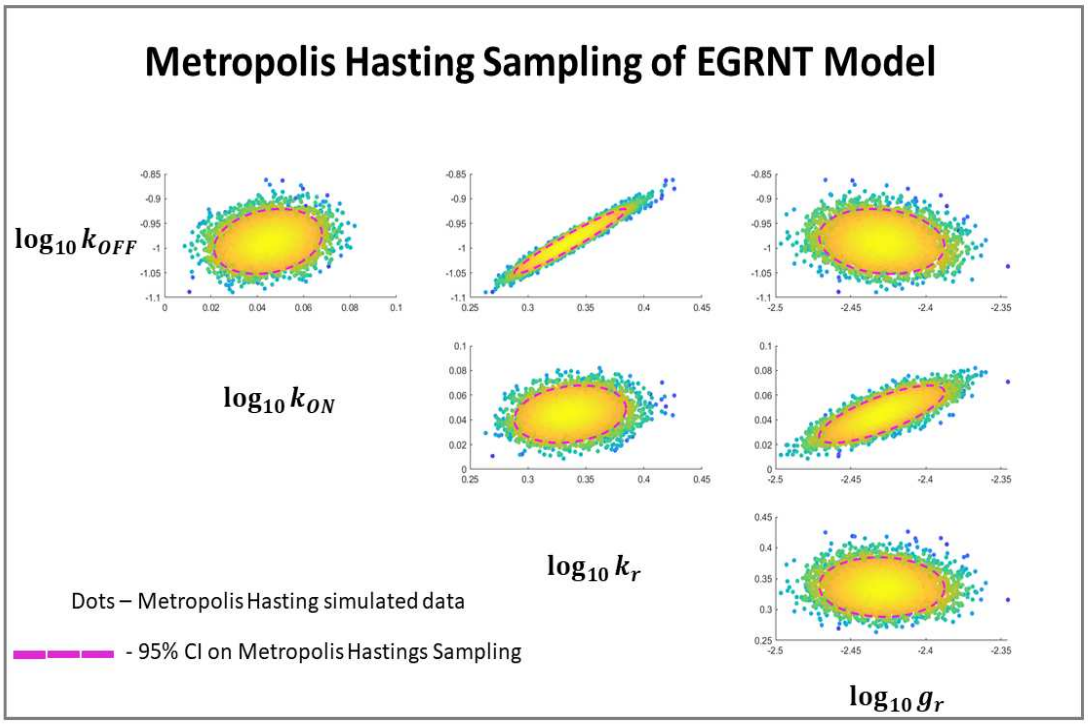

Figure 3.5: Metropolis Hastings (MH) analysis of parameter uncertainty given experimental data. The posterior parameter space of the EGRNT model was sampled using MH to generate 15,000 parameter combinations depicted as individual dots. Colors denote the computed likelihood values from high (yellow) to low (blue). The dashed pink line denotes the 95% CI of the parameter estimates computed from the MH samples. Each panel shows the joint uncertainty of two parameters, and reveals that some combinations of parameters are highly correlated. For example, $\log_{10} k_{OFF}$ is linearly correlated to $\log_{10} k_r$, suggesting that the burst size k_r/k_{OFF} is tightly constrained by the data, but the actual values for each of these individual parameters is relatively uncertain.

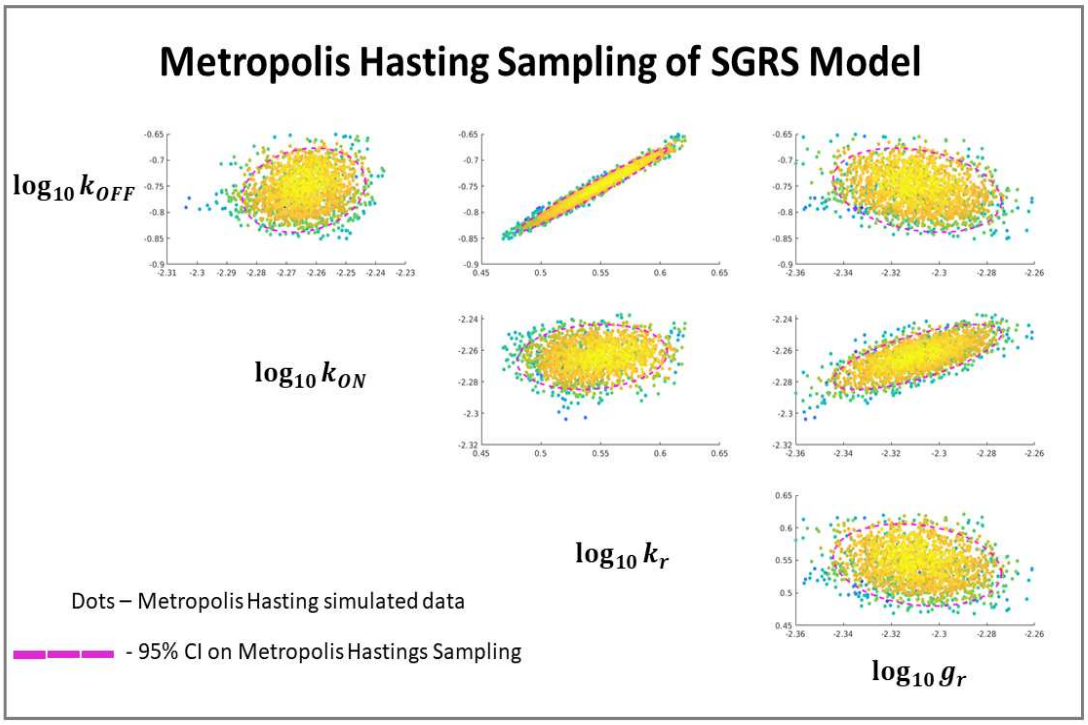

Figure 3.6: Metropolis Hastings (MH) analysis of parameter uncertainty given experimental data. The posterior parameter space of the SGRS model was sampled using MH to generate 15,000 parameter combinations depicted as individual dots. Colors denote the computed likelihood values from high (yellow) to low (blue). The dashed pink line denotes the 95% CI of the parameter estimates computed from the MH samples. Each panel shows the joint uncertainty of two parameters, and reveals that some combinations of parameters are highly correlated. For example, $\log_{10} k_{OFF}$ is linearly correlated to $\log_{10} k_r$, suggesting that the burst size k_r/k_{OFF} is tightly constrained by the data, but the actual values for each of these individual parameters is relatively uncertain.

The posterior parameter space of the both models was efficiently sampled to generate the data points depicted in the graph. The 95% CI, effectively encompassed the uncertainty associated with the parameter estimates. Each individual dot in the plot corresponds to a specific parameter within the set, thereby providing a comprehensive visualization of the parameter landscape. The parameter set for each model after fitting is shown in table 3.1.

Table 3.1: The resulting parameter set of the EGRNT and SGRS models. The expected value of each parameter is shown in the columns with the standard deviation presented after the parameter value.

Parameters	EGRNT	SGRS
$k_{nc}(min^{-1})$	25.8686 ±0.487	0.0538 ± 0.052
$k_{cn0}(min^{-1})$	0.0369 ± 0.0713	0.0220 ± 0.045
$k_{cn1}(min^{-1})$	1.2578 ± 0.0310	0.9369 ± 0.071
$r_1(min^{-1})$	0.0302 ± 0.0465	0.0538 ± 0.025
$k_{ON}(min^{-1})$	1.1079 ± 0.025	0.0055 ± 0.0247
$k_{OFF}(min^{-1})$	0.1023 ± 0.0705	0.1814 ± 0.0622
$k_r(min^{-1})$	2.1591 ± 0.0447	3.5961 ± 0.0456
$g_r(min^{-1})$	0.0037 ± 0.0517	0.0049 ± 0.0475

With the successful determination and preservation of a favorable parameter set for both models, the subsequent objective entailed the generation of the FIM for each model. This involved computation of the FIM, as described in the methods section.

3.3 The Fisher Information Matrix quantitatively predicts the variation in Maximum Likelihood Estimates among different simulated data set replicates.

Prior to proceeding with the utilization of the FIM for experiment design optimization, its validity and accuracy were confirmed through the MLE. Although the MLE can potentially yield a covariance estimate comparable to that derived from a sufficiently large number of samples, the FIM offers a significantly more efficient means of calculation. This expedites the acquisition of

valuable insights into the underlying model, rendering the FIM an indispensable tool for exploring diverse experiment designs within the EGRNT model.

The inverse determinant of the FIM, which is equal to $|COV|$, serves as a quantifiable measure of the information gained through each experiment design. Leveraging this metric, experiment designs can be optimized to strategically allocate cell population resources, further enhancing the efficiency of the overall experimental process.

Figure 3.6 showcases a comparative analysis of the variation observed in the MLE and the FIM for a specific scenario involving the observation of nuclear GR and DUSP1 mRNA within the EGRNT model. Additionally, figure 3.7 provides a complementary visual representation, contrasting the variance observed in the MLE and FIM specifically when solely observing DUSP1 mRNA. These comparative plots offer valuable insights into the accuracy and reliability of parameter estimation under different experimental conditions.

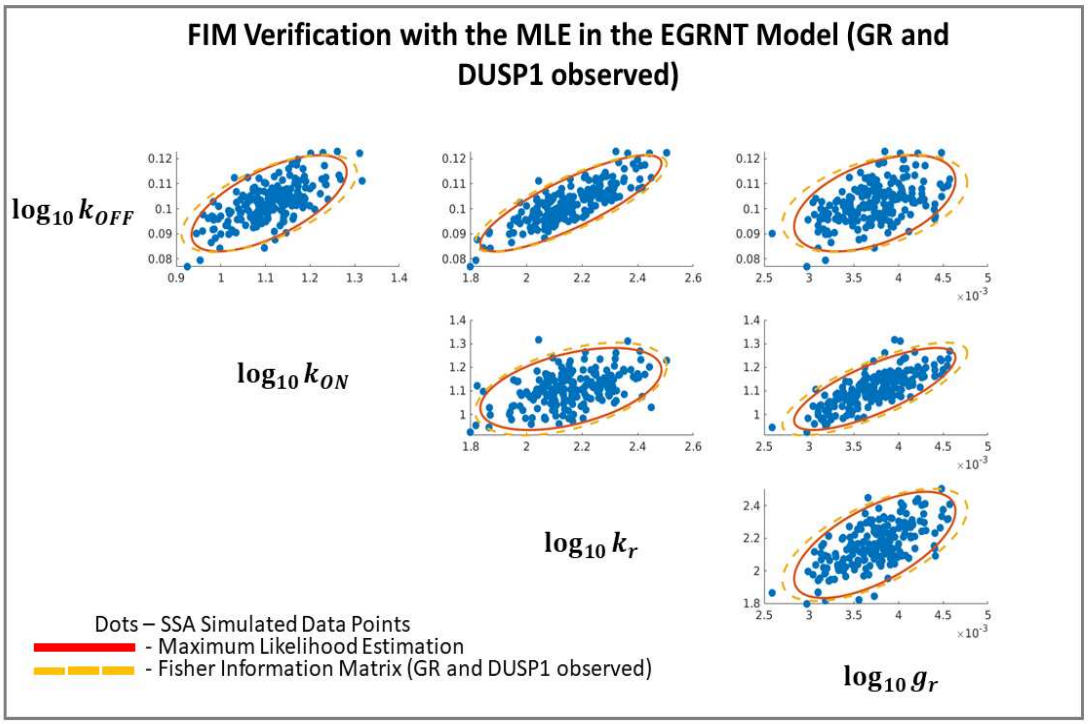

Figure 3.7: Comparison of spread of maximum likelihood estimates (MLE) to the predicted covariance from the FIM. Each of the 200 individual blue points is generated by fitting an independent simulated data set of 100 cells at each of 12 time points. The 95% CI of the MLE is shown in red and the EGRNT model's FIM when nuclear GR and DUSP1 mRNA are observed is shown in the dashed yellow line. The covariance is shown for each pairwise combination of the parameters k_{ON}, k_{OFF}, k_r, and g_r.

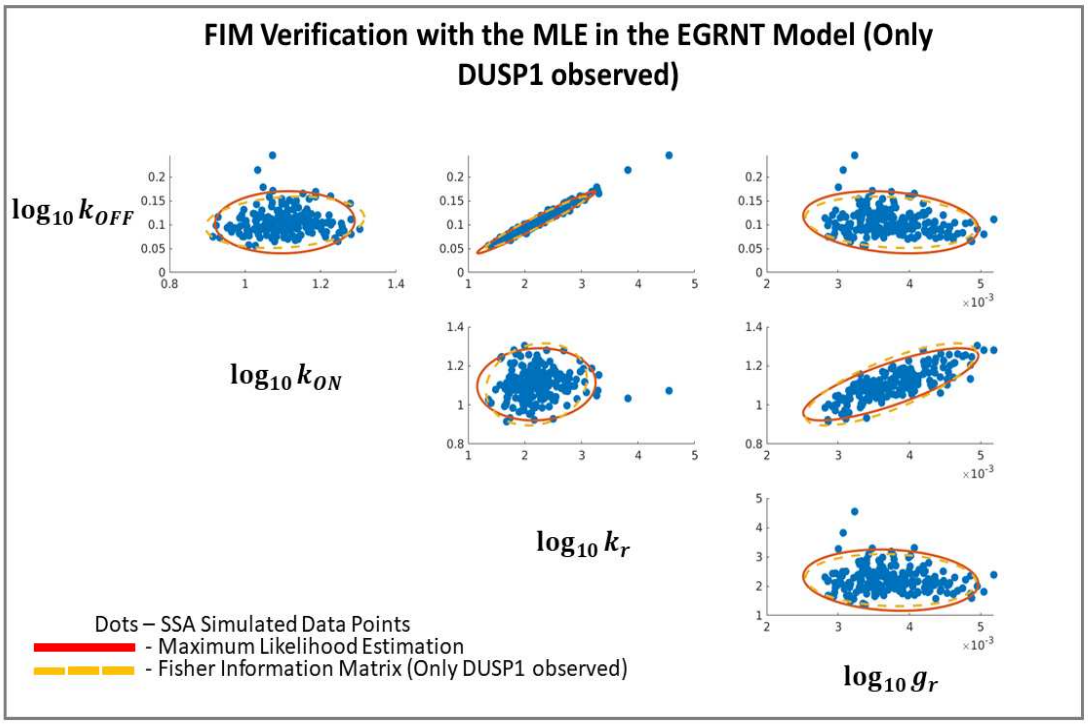

Figure 3.8: Comparison of spread of maximum likelihood estimates (MLE) to the predicted covariance from the FIM. Each of the 200 individual blue points is generated by fitting an independent simulated data set of 100 cells at each of 12 time points. The 95% CI of the MLE is shown in red and the EGRNT model's FIM when only DUSP1 mRNA is observed is shown in the dashed yellow line. The covariance is shown for each pairwise combination of the parameters k_{ON}, k_{OFF}, k_r, and g_r.

The congruence between the FIM and the MLE serves as an indicator of the FIM's efficacy in accurately capturing the covariance between parameters. In our analysis, although the fitted results obtained from the FIM closely approximated the MLE, they did not achieve an exact match. This discrepancy highlights the presence of a potential error within the SSIT code, prompting us to search the code for any potential issues. To validate this hypothesis, we conducted additional tests using alternative models, such as a birth decay process and a simple two-species gene expression model. Remarkably, in these cases, the FIM exhibited a near-perfect match with the MLE, further emphasizing the need to investigate the underlying cause of the mismatch observed specifically within the gene models under consideration.

3.4 Measuring GR and DUSP1 expression simultaneously is more informative than measuring these quantities in the same number cells divided into separate GR and DUSP1 experiments.

Three primary experiment designs are evaluated within this study. The first design involves measuring only the expression of DUSP1 mRNA through smFISH. The subsequent design entails partitioning an equal number of cells for measurements using both smFISH and ICC techniques. enabling the quantification of DUSP1 mRNA and GR. Lastly, a simultaneous measurement approach is employed, combining smFISH and ICC experiments within the same cell population to capture the expression profiles of both DUSP1 and GR.

Moreover, the EGRNT model is further stratified into two distinct cases. In the first case, the focus lies on identifying the gene regulation parameters, $[k_{ON}, k_{OFF}, k_r, g_r]$. Alternatively, the second case encompasses the identification of all model parameters, $[k_{ON}, k_{OFF}, k_r, g_r, k_{cn0}, k_{cn1}, k_{nc}, r_1]$.

To optimize the EGRNT DUSP1 model, an additional refinement is performed by strategically selecting informative time points. These time points are determined based on their high information content, enabling the allocation of more cells to be measured at those specific time points. Conversely, time points that are deemed less informative are deliberately excluded from the measurement design, ensuring efficient utilization of experimental resources.

The experimental data-set employed for model fitting comprised measurements from a total of 9639 cells obtained across various time points. Specifically, measurements were taken at 0, 10, 20, 30, 40, 50, 60, 120, 150, and 180 minutes following Dex stimulation. To streamline the experimental process and enhance the informativeness of the measurements, an optimization procedure was employed to determine the most informative time points for cell measurements. Consequently, the allocation of cells for the SGRS and EGRNT DUSP1 models was optimized,

resulting in the generation of the following figures, which provide a visual representation of the optimized experimental design as seen in figure 3.9.

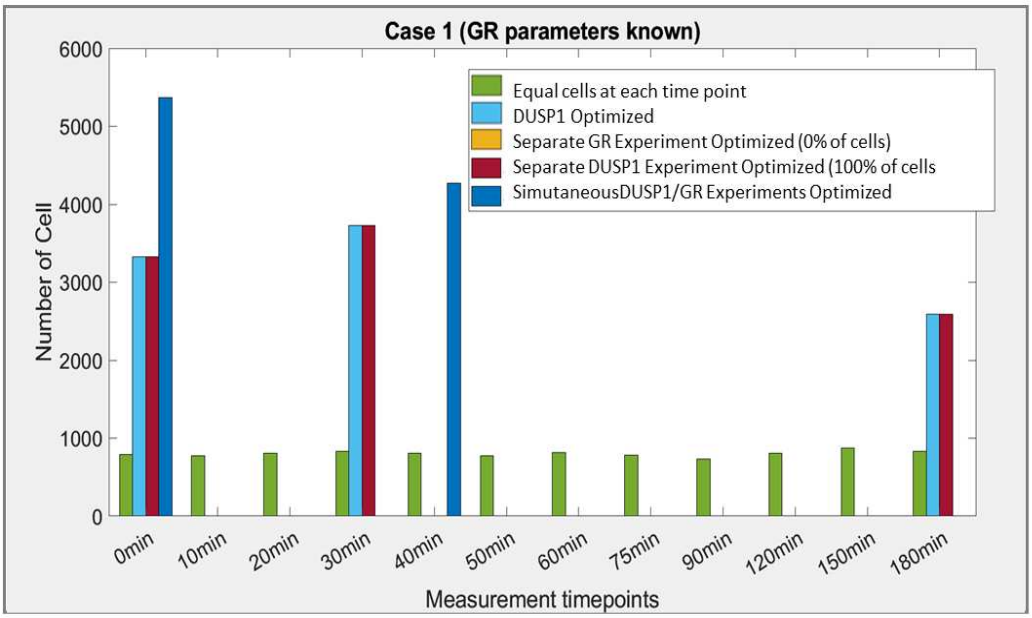

Figure 3.9: Optimal design for the number of cells to measure at each time point, assuming the EGRNT model and that GR parameters are known in advance. The FIM analysis was used to optimize the allocation of cells per time point for each experiment, resulting in a proposed number of cells to be measured at each specific time point. These proposed allocations are indicated in the figures, providing guidance for the experimental design. The optimal designs are compared to the original intuitive design that contained a roughly equal number of cells allocated to each time point (shown in green). The design focusing solely on DUSP1 measurements is depicted in cyan. The separate designs for DUSP1 and GR measurements are shown in maroon and yellow, respectively. Finally, the simultaneous measurement design for DUSP1 and GR is illustrated in dark blue. It's important to note that these optimized designs are specifically applied to Case 1, where the parameters related to nucleus GR translocation ($k_{nc}, k_{cn0}, k_{cn1}, r_1$) are already known.

In Case 1, the optimization results suggest that measuring cells at 3 or fewer time points would be sufficient. This reduction in the number of measurement time points from 12 to 3 significantly improves the workload for the experimenter. Notably, in this optimization, the number of cells allocated to the separate GR experiment is zero. This decision is based on the assumption that

the parameters related to GR translocation are already known from previous experiments or existing literature. Although the GR experiment could potentially provide information regarding GR translocation and the gene activation parameter k_{ON}, the optimization reveals that utilizing all the cells for the DUSP1 experiment design is preferable in this scenario.

For Case 2, where none of the parameters are known, the optimization results are presented in Figure 3.10 below. This figure showcases the optimized experimental design for simultaneous measurement of DUSP1 and GR, considering the comprehensive identification of all model parameters.

[hp]

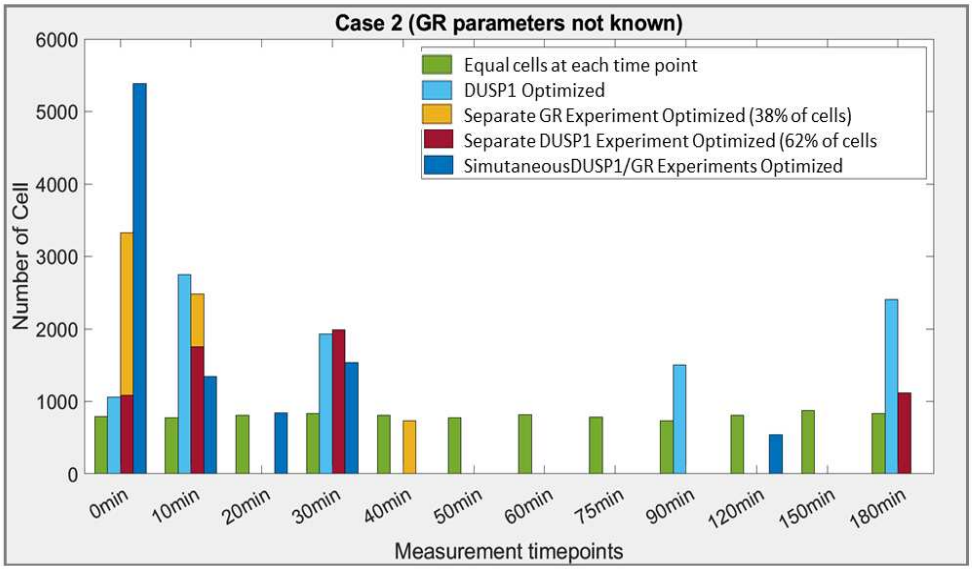

Figure 3.10: Optimal design for the number of cells to measure at each time point, assuming the EGRNT model and that GR parameters are *not* known in advance. The optimization process includes the allocation of an optimal number of cells for each experiment, resulting in a proposed number of cells to be measured at each specific time point. These proposed allocations are indicated in the figures, providing guidance for the experimental design. Additionally, a general design with a roughly equal number of cells allocated to each time point is represented by the color green. The design focusing solely on DUSP1 measurements is depicted in cyan. The separate designs for DUSP1 and GR measurements are shown in maroon and yellow, respectively. Finally, the simultaneous measurement design for DUSP1 and GR is illustrated in dark blue. It is important to note that this optimization pertains to Case 2, where none of the parameters are known. The figures serve as a comprehensive guide for the experimental design, facilitating informed decision-making and efficient resource allocation.

The optimization process for the measurement times revealed that the time points required to identify the model parameters vary among the different experimental designs. However, the optimization results indicate that only 5 time points or less depending on the experiment design are necessary to successfully identify the parameters in Case 2.

In the separate DUSP1/GR experiment design, the allocation of cells is divided between the ICC and smFISH experiments. A significant proportion of cells, 62%, is allocated to the smFISH

experiment, while the remaining 38% is allocated to the ICC experiment. This allocation strategy takes into consideration the fact that the parameters related to GR nuclear translocation are not known, and DUSP1 mRNA measurements alone do not provide information about nuclear GR translocation dynamics. This approach reduces the time required to perform single-cell experiments while simultaneously increasing the amount of information obtained from each experiment.

To compare the efficacy of each experimental design, the inverse determinant of the $|FIM^{-1}|$, which corresponds to the determinant of covariance matrix, $|COV|$, is plotted in Figure 3.11. This plot serves as a valuable tool for evaluating the information content and performance of each experimental design.

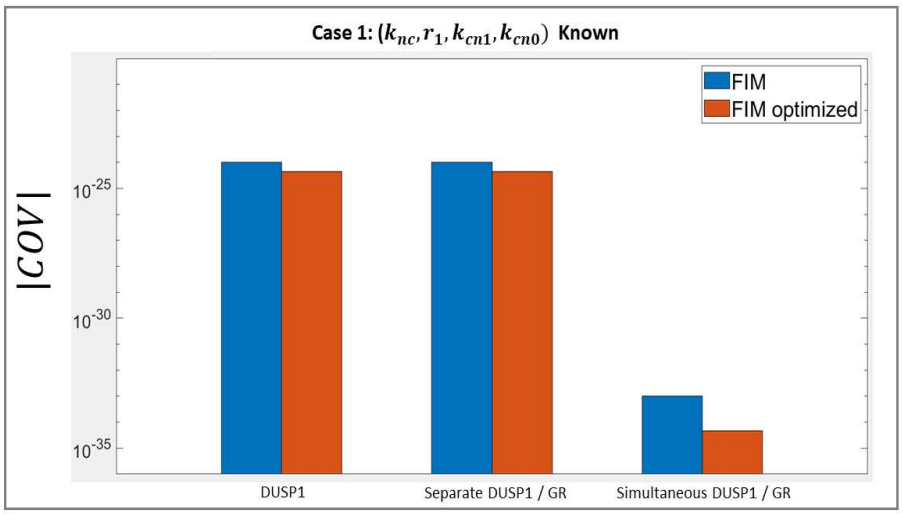

Figure 3.11: Expected uncertainty volume for different experiment types and using the EGRNT model and assuming that all GR parameters are known in advance. The expected determinant of the MLE covariance matrix, $|\Sigma|$, as predicted by the FIM for each experiment design for Case 1 (where the GR signalling parameters are known). The uncertainty for the original experiment design (12 time points with roughly equal numbers of cells per time point) is shown in blue, and the predicted uncertainty for the simpler optimized experiments (see Figure 3.9) are shown in red.

In Case 1, the optimal experimental design suggests that the simultaneous DUSP1/GR measurement provides the most valuable information. The separate DUSP1/GR experiment design,

despite allocating only half of the cell population for GR translocation measurements, yields a similar amount of information as solely measuring DUSP1. This outcome can be attributed to the fact that the GR translocation parameters are already known. In the optimized experiment design, none of the cells are allocated to GR measurements, while in the unoptimized design, approximately half of the cells are assigned to GR measurements at each time point. This finding implies that measuring GR does not provide additional information to the model unless the GR measurements are conducted in the same cells where DUSP1 mRNA is measured. To visually represent the comparison between different experiment designs, the same plot, illustrating the inverse determinant of the FIM, is generated for Case 2, and is presented in figure 3.12.

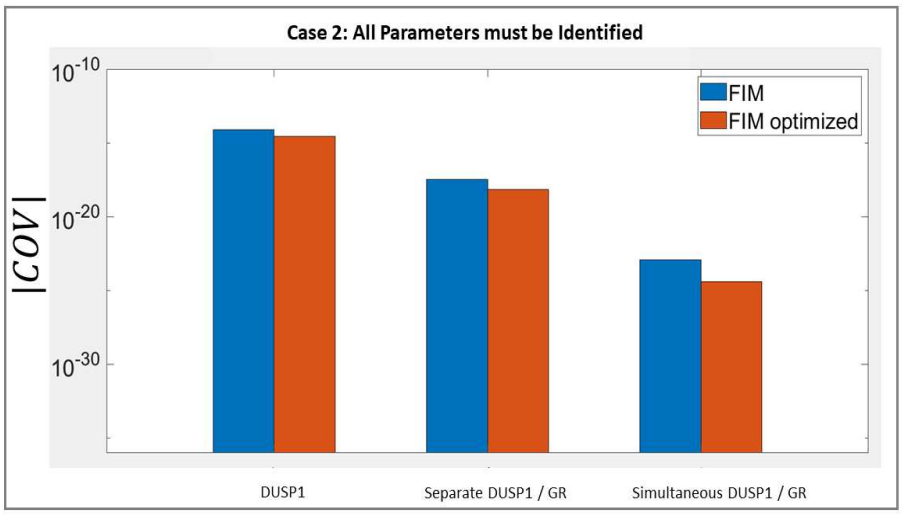

Figure 3.12: Expected uncertainty volume for different experiment types and using the EGRNT model and assuming that all GR parameters are *not* known in advance. The determinant of the MLE covariance matrix, $|\Sigma|$, as predicted by the FIM for each experiment design for Case 2 (where none of the parameters are known). The uncertainty for the original experiment design (12 time points with roughly equal numbers of cells per time point) is shown in blue, and the predicted uncertainty for the simpler optimized experiments (see Figure 3.10) are shown in red.

In both Case 1 and Case 2, the experiment design that yields the most informative results is the simultaneous DUSP1/GR measurement. If this design is feasible for the experimenter to

perform, it is recommended as the preferred choice. However, if conducting the simultaneous measurement is not practical, the experimenter may choose to perform the separate DUSP1 and GR experiments, particularly when the GR translocation parameters are unknown. In situations where the GR translocation parameters are already known, utilizing all the cells in a smFISH experiment solely for DUSP1 measurement provides an equivalent amount of information compared to conducting separate experiments.

At this stage, the experimenter can consider multiple factors, such as the optimal experiment design, cost implications, and the level of experimental complexity, to make an informed decision regarding the chosen experiment. It is important to strike a balance between maximizing information gain and considering practical constraints to ensure the experimental design aligns with the specific requirements and resources available to the experimenter.

The optimization process for the SGRS model follows a similar approach. The optimized cell allocation for the SGRS model experiment design is depicted in figure 3.13. This figure provides a visual representation of the proposed number of cells to be measured at each specific time point, taking into account the information content and efficiency of the experiment.

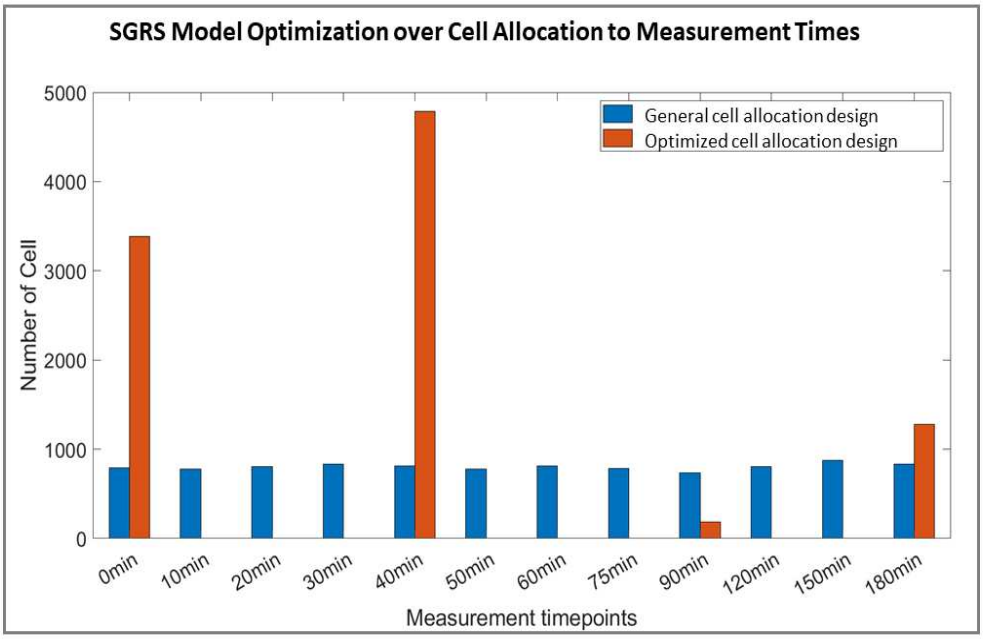

Figure 3.13: Optimal design for DUSP1 mRNA FISH experiments using the SGRS model and assuming that all GR parameters are known in advance. The optimization process includes the allocation of an optimal number of cells for each experiment, resulting in a proposed number of cells to be measured at each specific time point. These proposed allocations are indicated in the figures, providing guidance for the experimental design. Additionally, a general design with a roughly equal number of cells allocated to each time point is represented by the color blue. The optimized cell allocation design is illustrated in red. The figures serve as a comprehensive guide for the experimental design by suggesting efficient resource allocation in the SGRS model.

To evaluate the performance and information content of the experimental designs, Figure 3.14 displays the Fisher Information Matrix (FIM) for both the general experiment design and the optimized SGRS model design. By comparing these plots, researchers can assess the quality of the information obtained and the effectiveness of the optimized experimental design.

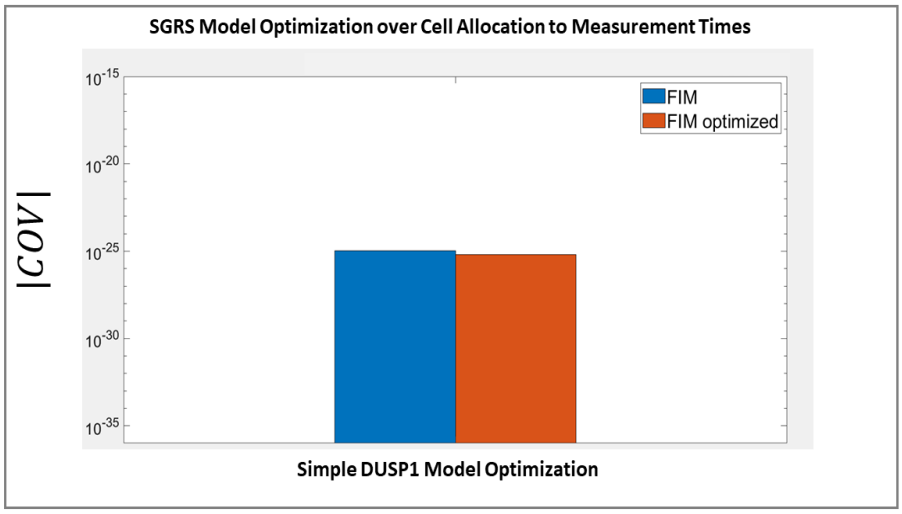

Figure 3.14: Expected uncertainty volume using the SGRS model and assuming that all GR parameters are *not* known in advance. The determinant of the MLE covariance matrix, $|\Sigma|$, as predicted by the FIM for the SGRS model experiment design. The uncertainty for the original experiment design (12 time points with roughly equal numbers of cells per time point) is shown in blue, and the predicted uncertainty for the simpler optimized experiments (see Figure 3.13) are shown in red.

The optimization of the experimental design for cell measurement resulted in a slight improvement in the amount of information gained, approximately 1.69 times higher compared to the initial design. This optimization also led to a reduction in the required number of measurement times, resulting in a more efficient experiment with a reduced workload.

Another approach to enhance parameter identification capabilities is by increasing the number of experimental replicates. To illustrate this effect, the covariance matrix $|COV|$ for each experiment design is plotted against the number of cells measured. The impact of the number of cells measured on the $|COV|$ can be quantified using Equation 3.1, which characterizes the relationship between the FIM and number of cells measured.

By examining the plots and $|COV|$, researchers can gain insights into the influence of sample size on the reliability of parameter estimation. Increasing the number of cells measured through replicates offers a potential avenue for refining parameter identification and improving the overall

robustness of the experimental results. The effect of increasing the number of cells measured is shown by Equation 3.1

$$|COV| = \det((n/N_{total}) * FIM)^{-1}). \tag{3.1}$$

The variable n is the proposed number of cells being measured and variable N_{total} is the actual number of cells that were measured in the original experiment performed. The comparison of increasing the number of cell measurements on each experiment to get a desired amount of information is shown in figure 3.15 under Case 1 and figure 3.16 under Case 2.

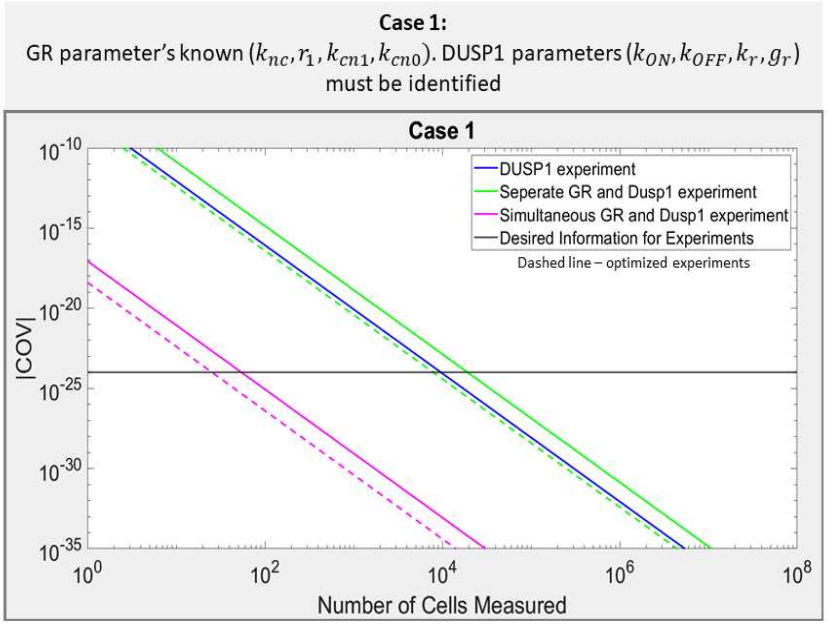

Figure 3.15: Comparison of the required number of cells to achieve equivalent information about the EGRNT model when using different experiment types and assuming that GR parameters are known in advance. The expected uncertainty volume $|FIM^{-1}|$ versus total number of cells is shown for each experiment design (assuming that all GR parameters are known in advance) is depicted using solid lines for the original experiment design (12 time points) and dashed lines for the optimized experiment design (3 or 4 times points)). The different experiment designs are visually differentiated as follows: the simultaneous DUSP1/GR design is shown in pink, the separate DUSP1/GR design is shown in green, and the DUSP1 experiment design is displayed in blue. To provide a reference for desired information content, a hypothetical target value of information is illustrated by the black line. This target value represents the desired level of precision and knowledge for parameter estimation.

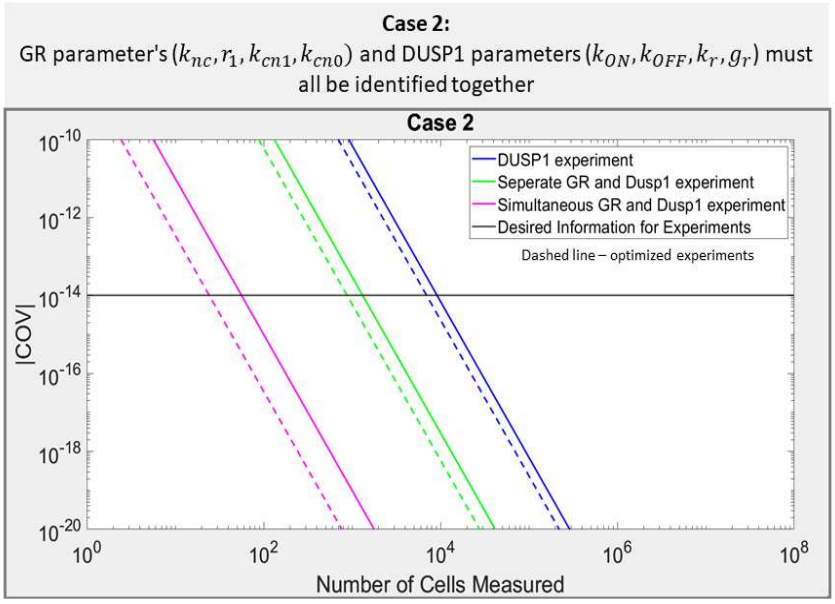

Figure 3.16: Comparison of the required number of cells to achieve equivalent information about the EGRNT model when using different experiment types and assuming that GR parameters are *not* known in advance. The FIM (Fisher Information Matrix) for each experiment design is depicted using solid lines, while the optimized FIM is represented by dashed lines. The different experiment designs are visually differentiated as follows: the simultaneous DUSP1/GR design is shown in pink, the separate DUSP1/GR design is shown in green, and the DUSP1 experiment design is displayed in blue. To provide a reference for desired information content, a hypothetical target value of information is illustrated by the black line. This target value represents the desired level of precision and knowledge for parameter estimation.

The analysis highlights the feasibility of increasing the number of cell measurements as an alternative to conducting more complex experiments to achieve a comparable amount of information. In Case 1, the DUSP1 measurement experiment requires approximately 229 times more cell measurements compared to the simultaneous measurement design. Similarly, the separate DUSP1/GR design necessitates around 250 times as many cells as the simultaneous measurement design to attain the desired information threshold.

Interestingly, the separate GR/DUSP1 measurement design may require more cells than the DUSP1 experiment due to the unoptimized design allocating half of the cells to measure GR,

even though GR measurements do not contribute significantly to the overall information. This suggests that allocating resources to measure GR in this particular scenario may not be as efficient as focusing solely on DUSP1 measurements.

In Case 2, the DUSP1 measurement experiment requires approximately 141 times as many cell measurements compared to the simultaneous measurement design. Similarly, the separate DUSP1/GR design necessitates roughly 13 times as many cells as the simultaneous measurement design to reach the desired information threshold.

These findings highlight the potential advantage of optimizing the allocation of cell measurements to maximize information content. By strategically designing experiments and efficiently allocating resources, researchers can achieve robust parameter estimation with a reduced experimental workload.

3.5 FIM analysis suggests the an optimal triptolide administration time for the EGRNT and SGRS models to identify a DUSP1 repression model.

In the previous studies conducted on the DUSP1 gene, the focus was primarily on understanding a mechanism that activate DUSP1 gene expression. However, in this paper, our objective shifts towards investigating a gene repression mechanism that acts upon DUSP1. Specifically, we explore the potential of triptolide as a repressive agent. Prior to conducting the experiments, we employ the FIM to determine the optimal timing for triptolide application following Dex stimulation. By calculating the FIM for various proposed triptolide application times, we can evaluate and identify the most effective experiment design. The resulting plots below depict the FIM analysis conducted for both the SGRS and EGRNT models.

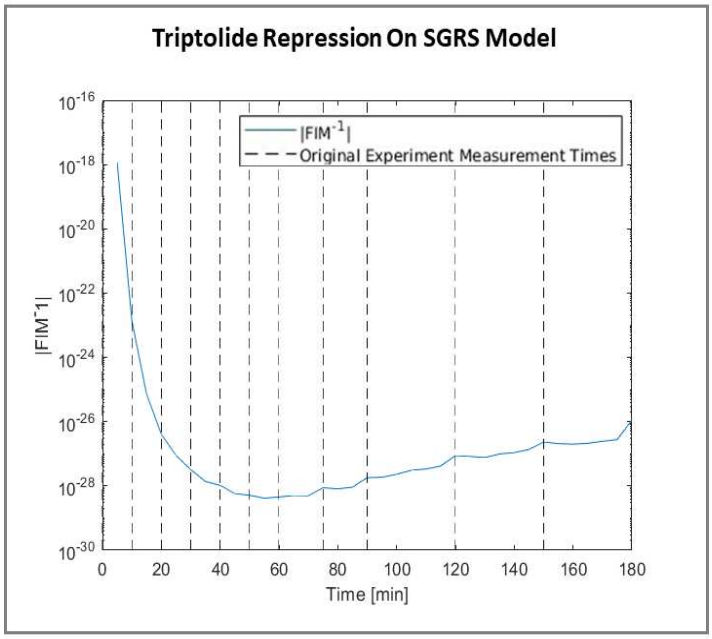

Figure 3.17: Prediction of information resulting from different transcription inhibition experiments, assuming the SGRS model. The predicted MLE uncertainty volume $|\Sigma| = |FIM^{-1}|$ for the the SGRS DUSP1 model is plotted as a function of a varying time for triptolide application (blue). Vertical dashed lines denote the measurement times at which smFISH measurements of the DUSP1 mRNA would be taken. The optimally informative experiment would apply triptolide at approximately t = 60 min.

For the SGRS DUSP1 model, the analysis reveals that the most optimal time to apply triptolide, in order to minimize $|FIM^{-1}|$, is approximately 55 minutes after Dex stimulation. At this specific time point, in experimental settings, the covariance between the model parameters is minimized, leading to an increased amount of valuable information for accurately identifying a model incorporating triptolide repression dynamics. The same analysis was also performed for the EGRNT model to create the corresponding, figure 3.18.

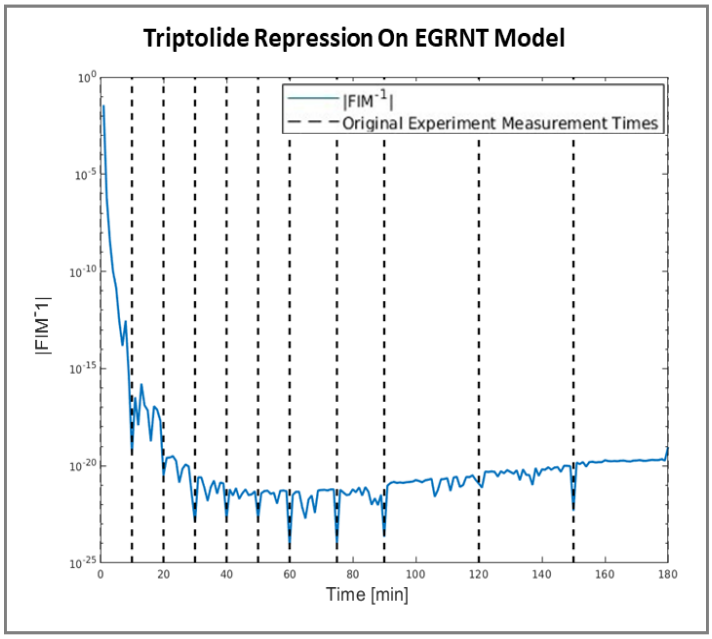

Figure 3.18: Prediction of information resulting from different transcription inhibition experiments, assuming the EGRNT model. The resulting $|FIM^{-1}|$ of the EGRNT model was plotted with varying triptolide application times shown in blue. The actual experiment measurement times for a previous DUSP1 experiment are shown in the dotted black line.

The analysis of the EGRNT model reveals that the $|FIM^{-1}|$ exhibits a higher degree of noise compared to the SGRS model. Notably, the noise appears to be concentrated around the time points where the original experimental measurements were taken. Despite this noise, the overall trend of the graph for the EGRNT model aligns with that of the SGRS model, suggesting that the optimal triptolide application time may likely be around a similar point. However, due to the presence of noise in the data, determining the precise optimal triptolide application time becomes challenging. Further analysis and careful consideration are required to ascertain the actual optimal triptolide application timing in the EGRNT model.

3.6 Bayesian Information Criterion Suggest the SGRS is a Better Model of DUSP1 Regulation

Both the EGRNT model and the SGRS model exhibited a good fit to the experimental data, as demonstrated in Figure 3.2 for the EGRNT model and Figure 3.4 for the SGRS model. The similarity observed in these fits prompted a comparative analysis of the models using the Bayesian Information Criterion (BIC), the log-likelihood when only DUSP1 mRNA is observed, and the computation time for the log-likelihood. To assess the quality of the parameter fit, the model with a higher log-likelihood is considered to provide a better fit. On the other hand, a lower BIC value generally indicates a superior model fit [46]. The BIC is calculated using Equation 3.2 [47].

$$BIC = k * ln(n) - 2ln(\hat{L}). \tag{3.2}$$

By evaluating these metrics, we can gain insights into the relative performance of the models and make informed comparisons to determine the most suitable model for the given experimental data. These evaluation criterion are outlined in table 3.2.

Table 3.2: The Log-likelihood, BIC, and computation time of the log-likelihood are calculated for the EGRNT and SGRS models and shown below.

Model	Log-likelihood	Bayesian Information Criterion	Computation Time (seconds
EGRNT	-4.503e4	90133	2.3889
SGRS	-4.4909e4	89891	0.2794

The comparison of the log-likelihood, BIC, and computation time between the EGRNT and SGRS models suggests that both models provide a good fit to the data. The log-likelihoods of the models are similar, indicating that both models capture the observed data well. However, the SGRS model shows slightly better fits based on the log-likelihood. In terms of computation time,

the SGRS model is faster than the EGRNT model for evaluating the log-likelihood. This indicates that the SGRS model is more computationally efficient.

The SGRS model has the smaller BIC indicating a better model fit for DUSP1 regulation. Also, the difference in BIC values was much larger than 10 inferring strong evidence of the SGRS model being better. Since both models have the same number of cells for fitting the data, n and the same number of parameters, k, the BIC value is only dependant on the likelihood values. Further more, the difference in the log-likelihood values of the models is less than 1%. Given that the BIC values rely solely on the log-likelihood, further investigation is needed to comprehensively evaluate the better model fit. Additional analysis or comparison metrics may be necessary to gain a more thorough understanding of the model fits and to make a more definitive conclusion regarding the superiority of one model over the other.

Chapter 4

Discussion

The design of single-cell microscopy experiments is a complicated task. However, because these experiments are often very expensive and there are many different experiments that one could choose to conduct. It is important to consider why these experiments are to be conducted and what is hoped to be achieved upon their analysis. In many research settings, quantitative experimental data is necessary in order to identify and constrain models that can then describe and predict how these biological systems will respond to novel stresses or environmental changes. The aim of this paper was to introduce a method to evaluate experiment designs through the use of the FIM to gain insight on how effective different experiments might be to achieve such goals and to determine how best to design new experiments. To do this, we proposed two models that capture the distribution of the real DUSP1 mRNA population over time after Dex stimulation. The first model, EGRNT, is a 3 species model that tries to capture the nuclear translocation dynamics of GR and the mRNA transcription dynamics. The second model, SGRS, is a simplification of the EGRNT model that includes the GR translocation dynamics as a time varying signal and reduces the species in the model to the gene state and the mRNA produced. These models were fit to real data to create models that accurately reproduce the DUSP1 transcription dynamics. The EGRNT and SGRS models were compared to one another using the log-likelihood, BIC, and computation time to evaluate which is the best model of DUSP1 transcription dynamics. The SGRS model proved to have a slightly larger log-likelihood and was substantially more computationally efficient. However, because the log-likelihood difference was less than 1% and both models have the same number of parameters, further validation would be needed to confirm the SGRS model as the better model to describe DUSP1 gene regulation, and we performed all remaining analyses using both models.

With the models defined, the FSP method [31] was used to generate the FSP based FIM [19]. According to the Cramer Rao lower bound (CRLB), the inverse of the FIM provides a lower bound

on the expected co-variance of maximum likelihood estimates collected from independent data generated by the process. As such, we verified the CRLB for the FSP-FIM for our models by generating many simulated data sets from the models and then finding MLE parameter sets, as demonstrated previously [12, 19, 48]. We found that the FIM matched closely to the statistics of the MLE as shown in figure 3.7 and figure 3.8, although we observed that in some cases the FIM predicted a larger variance than that observed in the MLE. Although this may suggest that there is an issue in the SSIT code, which is currently being investigated, this observation is likely due to the possibility that MLE fits had not been given sufficient time to fully converge.

Having demonstrated close agreement of the FIM and the spread of the MLE for simulated data, we then used the FSP-FIM approach to explore the optimization of the experiment design. The first such optimization was performed to find the optimal time points to measure cells. This optimization reduced the expected covariance of the parameters (i.e., it would make the parameters identifiable to greater precision) for every model and experiment design under consideration. Moreover, the selected experiments were all much simpler and required fewer time points than the original, intuitive experiment. This analysis showed that model-driven experiment design could lead to more informative experiments, that are actually easier and less expensive to perform than those that would be designed using intuition alone.

The second optimization analysis compared the expected uncertainty volume for 3 different experiment designs that use different combinations of smFISH to measure the DUSP1 mRNA expression or ICC to measure nuclear GR over time. Through this method, as expected, the most informative experiment was revealed to be a simultaneous measurement of GR and DUSP1 mRNA in the same cells. The next informative experiment was splitting the cells into two separate ICC and smFISH experiment, although this approach is only necessary in case the nuclear GR translocation parameters are not known. Beyond confirming the somewhat obvious fact that the most complicated experiment would reveal the most information about the parameters, the FSP-FIM analysis provided a rigorous means to quantify the relative value of these experiments, and how many measurements would be needed to achieve the same amount of information using each of

these approaches. For example, in the situation where a complex experiment requiring measurement of both DUSP1 and GR at the same time in the same cells could not be accomplished or was prohibitively expensive, we found that the same information could be obtained using 13 times as many cells if the two species could be measured separately in different cells or 141 times as many cells if one only measured the DUSP1 mRNA (see Figure 3.16). Having such insight before choosing an experiment could provide valuable guidance on what experiments are absolutely necessary, and which could be replaced using a greater number of replicas of simpler or less expensive approaches.

The last optimization we performed was to use the FIM to infer the best time to apply a second drug (i.e., triptolide) to the process so that we could learn even more about the parameters of the DUSP1 gene. The FIM showed that the best time to apply triptolide was at 55 minutes after Dex stimulation in the SGRS model as shown in figure 2.8. According to the FIM analysis for the SGRS model (figure 3.17), this application would reduce the uncertainty volume by a factor of 294. Assuming that all eight free parameters (including those describing the GR dynamics) needed to be identified, this increase in information per measurement would effectively reduce the total number of cells needed to achieve the same information by a factor ($294^{1/8}$= 2.04). The EGRNT model showed a similar trend as the SGRS for the optimal triptolide application time but the analysis was noisier which makes it hard to confirm the best application time.

In conclusion, modern single-cell microscopy technology now allows for the measurement of various different fluorescent probes in single cell experiments, and these techniques are rapidly evolving. Over time, this progress will certainly increase the types of model and experiments that can be combined to describe and predict biological processes. However, as these experiment become increasingly complex and expensive to do, it is important to develop new tools to choose how best to apply these experimental efforts. To aid in this process, the methods introduced in this study demonstrate the capabilities of the FIM to quantify the value for different designs for single-cell experiments. With the FIM, one may seek to optimize established designs and change conditions (e.g., time points or numbers of cells) to increase the identifiability of the model param-

eters from experimental data. However, in other cases, the FIM analysis can also provide advance knowledge that such identification may be inefficient or impossible using available experimental approaches (i.e., when the FIM has low rank or very small eigenvalues). Armed with this capability, researchers can hypothesize different potential future experiments and ask what new insights or benefits could reasonably be expected, or if the same amount of insight could be achieved using additional replicas of a simpler, less exact experimental approach. By leveraging the power of the FIM to assess the potential of different experimental designs, researchers can prioritize more informative experiments, avoid unnecessarily complex or expensive experiments, and expedite their understanding of gene expression and the underlying mechanisms of biological control.

Bibliography

[1] A. S. Ribeiro, "Stochastic and delayed stochastic models of gene expression and regulation," *Mathematical biosciences*, vol. 223, no. 1, pp. 1–11, 2010.

[2] B. Munsky, Z. Fox, and G. Neuert, "Integrating single-molecule experiments and discrete stochastic models to understand heterogeneous gene transcription dynamics," *Methods*, vol. 85, pp. 12–21, 2015.

[3] D. Zenklusen, D. R. Larson, and R. H. Singer, "Single-rna counting reveals alternative modes of gene expression in yeast," *Nature structural & molecular biology*, vol. 15, no. 12, pp. 1263–1271, 2008.

[4] G. Neuert, B. Munsky, R. Z. Tan, L. Teytelman, M. Khammash, and A. Van Oudenaarden, "Systematic identification of signal-activated stochastic gene regulation," *Science*, vol. 339, no. 6119, pp. 584–587, 2013.

[5] C. Zimmer, "Experimental design for stochastic models of nonlinear signaling pathways using an interval-wise linear noise approximation and state estimation," *PloS one*, vol. 11, no. 9, p. e0159902, 2016.

[6] N. Tsanov, A. Samacoits, R. Chouaib, A.-M. Traboulsi, T. Gostan, C. Weber, C. Zimmer, K. Zibara, T. Walter, M. Peter, *et al.*, "smifish and fish-quant–a flexible single rna detection approach with super-resolution capability," *Nucleic acids research*, vol. 44, no. 22, pp. e165–e165, 2016.

[7] T. Antakly, E. B. Thompson, and D. O'Donnell, "Demonstration of the intracellular localization and up-regulation of glucocorticoid receptor by in situ hybridization and immunocytochemistry," *Cancer Research*, vol. 49, no. 8_Supplement, pp. 2230s–2234s, 1989.

[8] L. S. Forero-Quintero, W. Raymond, T. Handa, M. N. Saxton, T. Morisaki, H. Kimura, E. Bertrand, B. Munsky, and T. J. Stasevich, "Live-cell imaging reveals the spatiotempo-

ral organization of endogenous rna polymerase ii phosphorylation at a single gene," *Nature Communications*, vol. 12, no. 1, p. 3158, 2021.

[9] T. Fukaya, B. Lim, and M. Levine, "Enhancer control of transcriptional bursting," *Cell*, vol. 166, no. 2, pp. 358–368, 2016.

[10] D. M. Suter, N. Molina, D. Gatfield, K. Schneider, U. Schibler, and F. Naef, "Mammalian genes are transcribed with widely different bursting kinetics," *science*, vol. 332, no. 6028, pp. 472–474, 2011.

[11] J. Ruess, F. Parise, A. Milias-Argeitis, M. Khammash, and J. Lygeros, "Iterative experiment design guides the characterization of a light-inducible gene expression circuit," *Proceedings of the National Academy of Sciences*, vol. 112, no. 26, pp. 8148–8153, 2015.

[12] Z. R. Fox and B. Munsky, "The finite state projection based fisher information matrix approach to estimate information and optimize single-cell experiments," *PLoS computational biology*, vol. 15, no. 1, p. e1006365, 2019.

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

[14] L. Pronzato and É. Walter, "Robust experiment design via stochastic approximation," *Mathematical Biosciences*, vol. 75, no. 1, pp. 103–120, 1985.

[15] H. D. Vo, L. Forero, L. Aguilera, and B. Munsky, "Analysis and design of single-cell experiments to harvest fluctuation information while rejecting measurement noise," *bioRxiv*, pp. 2021–05, 2021.

[16] R. N. Gutenkunst, J. J. Waterfall, F. P. Casey, K. S. Brown, C. R. Myers, and J. P. Sethna, "Universally sloppy parameter sensitivities in systems biology models," *PLoS computational biology*, vol. 3, no. 10, p. e189, 2007.

[17] M. Komorowski, M. J. Costa, D. A. Rand, and M. P. Stumpf, "Sensitivity, robustness, and identifiability in stochastic chemical kinetics models," *Proceedings of the National Academy of Sciences*, vol. 108, no. 21, pp. 8645–8650, 2011.

[18] J. Ruess, A. Milias-Argeitis, and J. Lygeros, "Designing experiments to understand the variability in biochemical reaction networks," *Journal of The Royal Society Interface*, vol. 10, no. 88, p. 20130588, 2013.

[19] Z. Fox, G. Neuert, and B. Munsky, "Finite state projection based bounds to compare chemical master equation models using single-cell data," *The Journal of chemical physics*, vol. 145, no. 7, p. 074101, 2016.

[20] J. Hoppstädter and A. J. Ammit, "Role of dual-specificity phosphatase 1 in glucocorticoid-driven anti-inflammatory responses," *Frontiers in immunology*, vol. 10, p. 1446, 2019.

[21] A. Plotnikov, E. Zehorai, S. Procaccia, and R. Seger, "The mapk cascades: signaling components, nuclear roles and mechanisms of nuclear translocation," *Biochimica et Biophysica Acta (BBA)-Molecular Cell Research*, vol. 1813, no. 9, pp. 1619–1633, 2011.

[22] L. Chang, "karin m," *Mammalian MAP kinase signalling cascades. Nature*, vol. 410, pp. 37–40, 2001.

[23] M. Nixon, R. Andrew, and K. E. Chapman, "It takes two to tango: dimerisation of glucocorticoid receptor and its anti-inflammatory functions," *Steroids*, vol. 78, no. 1, pp. 59–68, 2013.

[24] D. P. Shepherd, N. Li, S. N. Micheva-Viteva, B. Munsky, E. Hong-Geller, and J. H. Werner, "Counting small rna in pathogenic bacteria," *Analytical chemistry*, vol. 85, no. 10, pp. 4938–4943, 2013.

[25] A. Senecal, B. Munsky, F. Proux, N. Ly, F. E. Braye, C. Zimmer, F. Mueller, and X. Darzacq, "Transcription factors modulate c-fos transcriptional bursts," *Cell reports*, vol. 8, no. 1, pp. 75–83, 2014.

[26] A. Imbert, W. Ouyang, A. Safieddine, E. Coleno, C. Zimmer, E. Bertrand, T. Walter, and F. Mueller, "Fish-quant v2: a scalable and modular tool for smfish image analysis," *RNA*, vol. 28, no. 6, pp. 786–795, 2022.

[27] C. Stringer, T. Wang, M. Michaelos, and M. Pachitariu, "Cellpose: a generalist algorithm for cellular segmentation," *Nature methods*, vol. 18, no. 1, pp. 100–106, 2021.

[28] J. Paulsson, "Models of stochastic gene expression," *Physics of life reviews*, vol. 2, no. 2, pp. 157–175, 2005.

[29] N. G. Van Kampen, *Stochastic processes in physics and chemistry*, vol. 1. Elsevier, 1992.

[30] D. A. Mcquarrie and D. T. Gillespie, "Stochastic theory and simulations of chemical kinetics," *J. Appl. Probab*, vol. 478, no. 413-478, p. 1140, 1967.

[31] B. Munsky and M. Khammash, "The finite state projection algorithm for the solution of the chemical master equation," *The Journal of chemical physics*, vol. 124, no. 4, p. 044104, 2006.

[32] S. Asmussen and P. W. Glynn, *Stochastic simulation: algorithms and analysis*, vol. 57. Springer, 2007.

[33] M. Gómez-Schiavon, L.-F. Chen, A. E. West, and N. E. Buchler, "Bayfish: Bayesian inference of transcription dynamics from population snapshots of single-molecule rna fish in single cells," *Genome biology*, vol. 18, pp. 1–12, 2017.

[34] H. Xu, S. O. Skinner, A. M. Sokac, and I. Golding, "Stochastic kinetics of nascent rna," *Physical review letters*, vol. 117, no. 12, p. 128101, 2016.

[35] L. A. Sepúlveda, H. Xu, J. Zhang, M. Wang, and I. Golding, "Measurement of gene regulation in individual cells reveals rapid switching between promoter states," *Science*, vol. 351, no. 6278, pp. 1218–1222, 2016.

[36] C. A. Kastner, A. Braumann, P. L. Man, S. Mosbach, G. P. Brownbridge, J. Akroyd, M. Kraft, and C. Himawan, "Bayesian parameter estimation for a jet-milling model using metropolis–hastings and wang–landau sampling," *Chemical Engineering Science*, vol. 89, pp. 244–257, 2013.

[37] F. Liu, X. Li, and G. Zhu, "Using the contact network model and metropolis-hastings sampling to reconstruct the covid-19 spread on the "diamond princess"," *Science Bulletin*, vol. 65, no. 15, pp. 1297–1305, 2020.

[38] S. Gupta, L. Hainsworth, J. Hogg, R. Lee, and J. Faeder, "Evaluation of parallel tempering to accelerate bayesian parameter estimation in systems biology," in *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, pp. 690–697, IEEE, 2018.

[39] A. Degasperi and S. Gilmore, "Sensitivity analysis of stochastic models of bistable biochemical reactions," in *Formal Methods for Computational Systems Biology: 8th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM 2008 Bertinoro, Italy, June 2-7, 2008 Advanced Lectures 8*, pp. 1–20, Springer, 2008.

[40] M. Morshed, B. Ingalls, and S. Ilie, "An efficient finite-difference strategy for sensitivity analysis of stochastic models of biochemical systems," *Biosystems*, vol. 151, pp. 43–52, 2017.

[41] R. Gunawan, Y. Cao, L. Petzold, and F. J. Doyle, "Sensitivity analysis of discrete stochastic systems," *Biophysical journal*, vol. 88, no. 4, pp. 2530–2540, 2005.

[42] G. Casella and R. L. Berger, "Statistical inference. wadsworth & brooks," *Cole, Pacific Grove, CA*, 1990.

[43] S. M. Kay, *Fundamentals of statistical signal processing: estimation theory*. Prentice-Hall, Inc., 1993.

[44] S. Ziaei and R. Halaby, "Immunosuppressive, anti-inflammatory and anti-cancer properties of triptolide: A mini review," *Avicenna journal of phytomedicine*, vol. 6, no. 2, p. 149, 2016.

[45] Y. Wang, J.-j. Lu, L. He, and Q. Yu, "Triptolide (tpl) inhibits global transcription by inducing proteasome-dependent degradation of rna polymerase ii (pol ii)," *PloS one*, vol. 6, no. 9, p. e23993, 2011.

[46] G. Schwarz, "Estimating the dimension of a model," *The annals of statistics*, pp. 461–464, 1978.

[47] A. R. Liddle, "Information criteria for astrophysical model selection," *Monthly Notices of the Royal Astronomical Society: Letters*, vol. 377, no. 1, pp. L74–L78, 2007.

[48] H. D. Vo and B. Munsky, "Designing single-cell experiments to harvest fluctuation information while rejecting measurement noise," *bioRxiv*, pp. 2021–05, 2022.

Appendix A

Cell Visualization and Experiment Verification

The smFISH data utilized in this study was acquired by my colleague, Eric Ron. This data-set played a crucial role in identifying the underlying models presented in this paper. During the data acquisition process, it was imperative to address potential limitations arising from the microscope. To ensure the integrity of the subsequent image processing pipeline, I created a visualization code developed in Python. This code facilitated the examination of cells alongside the corresponding detected spots extracted using the BIG-FISH image processing toolkit. The visualization function served as a quality control step, enabling the identification of any potential errors that may have arisen during the image processing phase. This script was implemented after the image processing stage, which involved the combined utilization of BIG-FISH [26] and Cellpose [27]. The integration of these tools allowed for the automated processing of smFISH images, further enhancing the reliability and efficiency of the analysis workflow.

Following the completion of the visualization process, the resulting data is saved as a CSV file. This file encompasses crucial details pertaining to spot detection, cell labeling, and cell locations. To ensure ease of use and flexibility, the visualization script is developed as a function that accepts parameters such as the experiment name and cell ID. By specifying these inputs, researchers gain access to the original image, annotated with the detected spots within the requested cell. In certain experiments, the introduction of multiple smFISH probes enables the tracking of multiple targets within the same population of cells, facilitating comprehensive analyses and insights. To accommodate this, the function offers the ability to apply various filtering options to refine the displayed information. Notably, the visualization includes two distinct types of spots.

The Python function only requires the cell ID and experiment name as inputs, but it also has other options to customize the visualization of the requested cell. The variable "csv_cell_ID" represents the associated cell ID, while the variable "string_name" corresponds to the experiment name of the saved CSV file. The width and height of the selected area to view the cell can be manu-

ally adjusted from their default values using the variables "x_width" and "y_height." Additionally, the z-stack of the image can be chosen for viewing by setting the "z_slice" variable. The spots displayed in the image are only those detected in the selected z-stack, but an option to view spots detected in nearby z-stacks is provided through the "delta_z" variable. If the spot ID is known, the "spot_show" variable allows the user to view specific spots in a given z-stack. The "filter" variable represents the filter applied to the images, while the "spot_filter" variable denotes the filter applied to the individual spot crops. An example of the input for the function is provided below. When no value is defined for the additional variables in the function or a value of -1 is given, the default value is used.

```
cell_ID=45
string_name = 'MS2-CY5_Cyto543_560_woStim___nuc_70__cyto_0__psfz_350__psfyx_160__ts_550_400'

find_cell(csv_cell_ID,string_name,x_width=150,y_height=150,z_slice=-1,spots_show = [-1],filter='None',delta_z=0,spot_filter='None')
```

Figure A.1: Command script to run the cell visualization. An example of how to run the function with the all the function variables included.

The function allows for precise customization of the visualization parameters. The variable "string_name" serves to specify the experiment name, ensuring the retrieval of the corresponding data. By setting the x width and y width as 150, the visualization focuses on a rectangular region centered around the cell's coordinates. To visualize the central z plane, a value of -1 is assigned to the z slice parameter. By providing the value [-1] for the "spot_to_show" variable, all spots detected in the selected z plane are displayed. The absence of any filtering criteria on the cell image is indicated by assigning the 'None' value to the "filter" variable. To restrict the display to spots solely from the selected z plane, the "delta_z" variable is set to zero, thereby excluding spots from other z planes. Similarly, by setting the "spot_filter" variable to zero, no spot crop filters are applied, allowing the visualization of all detected spots within the specified region.

The first output provided is the original image, with each image channel represented by a different color in a single image. An image showing the segmented area is presented next to the

original image, allowing the user to match the cell ID to the cell being visualized. An example is given below. Information about the image, such as the image ID, the cell, and the shape of the original image, is shown in the image below.

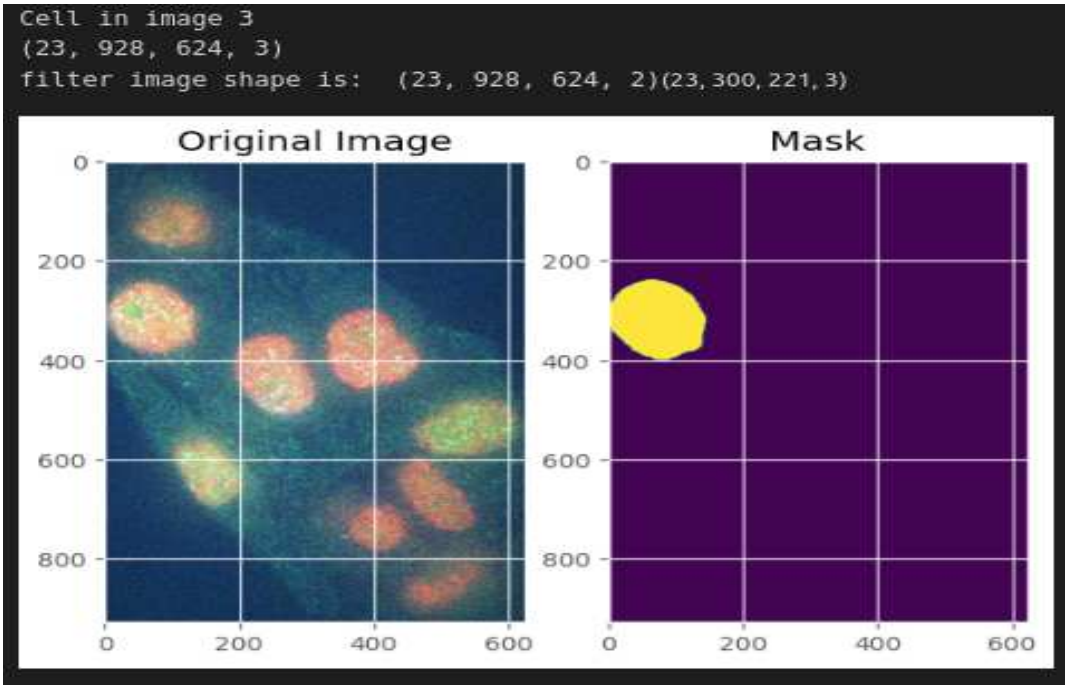

Figure A.2: Cell visualization of a cell in image 3 of the data set. The original cell microscope image, shown on the right, is plotted next to the image of the cell mask corresponding to the selected cell.

In this specific example, the third cell image in the data-set is selected for visualization. The original image is represented as a tensor with dimensions (21, 924, 624, 3). This implies that there are 22 z-plane images, each with a resolution of 925 by 625 pixels, and four channels corresponding to different color channels. The true number of z-planes, resolution, and number of channels is off by 1 due to python's indexing starting at 0.

The next outputs are individual crops of the cell with each image channel shown in an individual plot. The location of spots are outlined with squares and triangles to indicate their location within the cell. The different shapes indicates multiple types of spots in the image detection. The function allows for up to two spot detection channels. An example of this output is shown below.

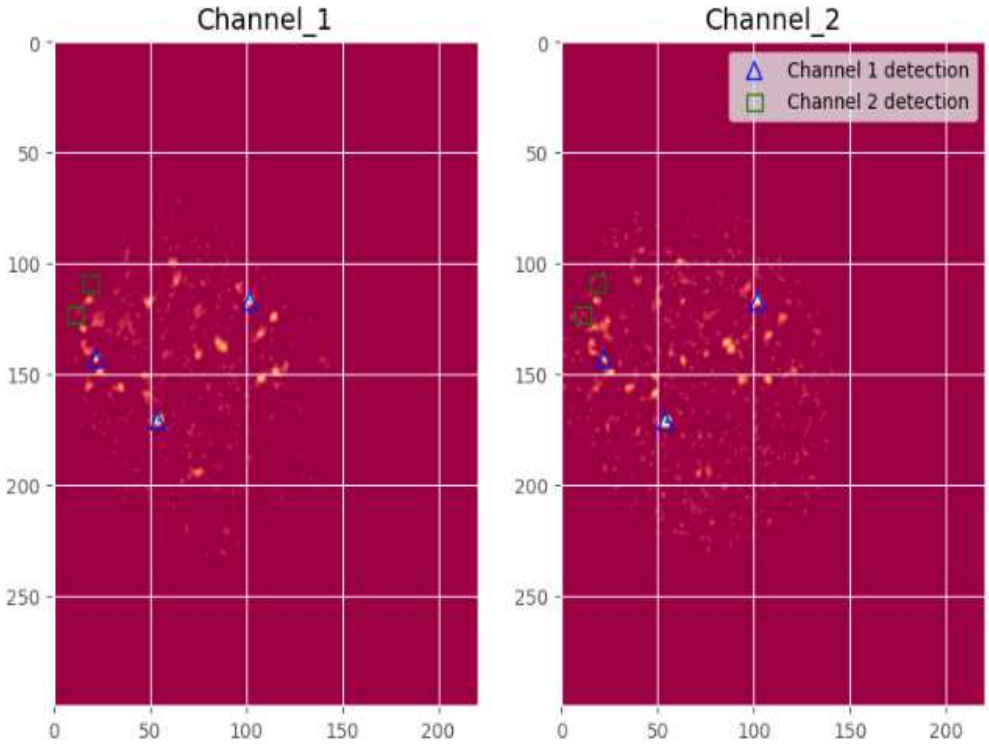

Figure A.3: Cell channel visualization with applied filters to view spots. The channels of the smFISH image with spots detected in the channel are shown in individual cell crops for each channel. The detected spot locations are shown in blue triangles for the first spot detection and green squares for the second spot detection.

The next output is a cell crop of the original image with the spot locations shown by triangles and squares. An example of this output is shown below.

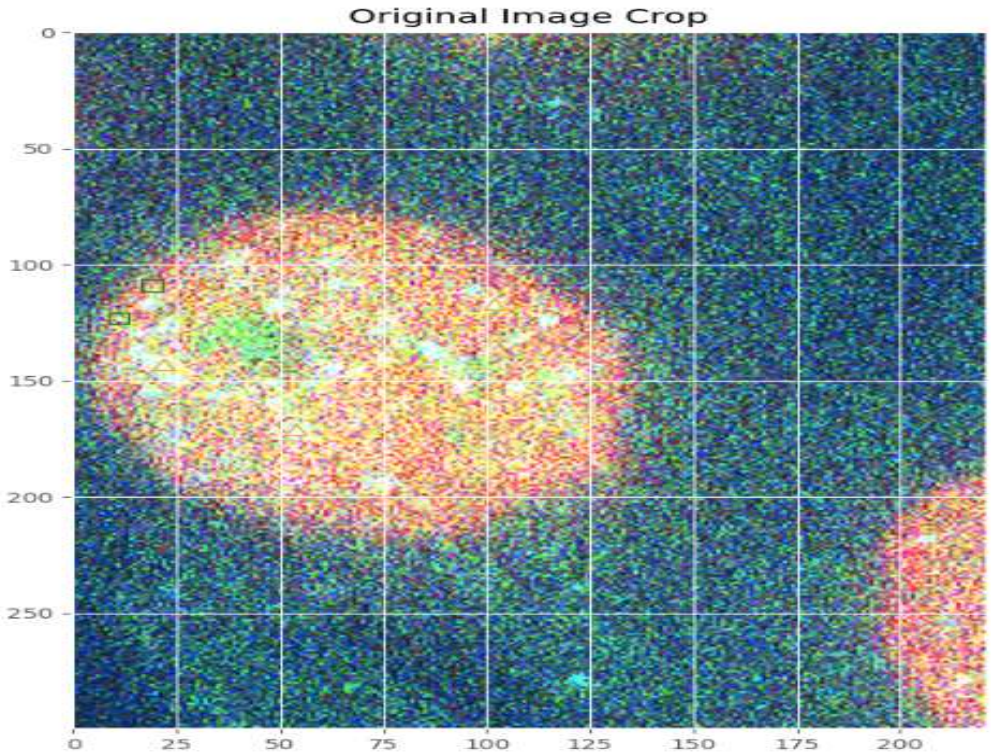

Figure A.4: Original image cropped to the cell of interest with marker showing spot locations. The original image of the smFISH image are shown in an individual cell crop. The detected spot locations are shown in orange triangles for the first channel spot detection and green squares for the second channel spot detection.

The final output of the function includes individual spot crops of the detected spots. Each spot crop shows the specific region where the spot was detected in each channel, and the spot ID is provided for each detected spot. This output is useful for identifying and verifying the spots detected by the BIG-FISH image processing pipeline. It allows the experimenter to visually assess whether the detected spots correspond to actual biological signals. An example of the output is shown in the figure below.

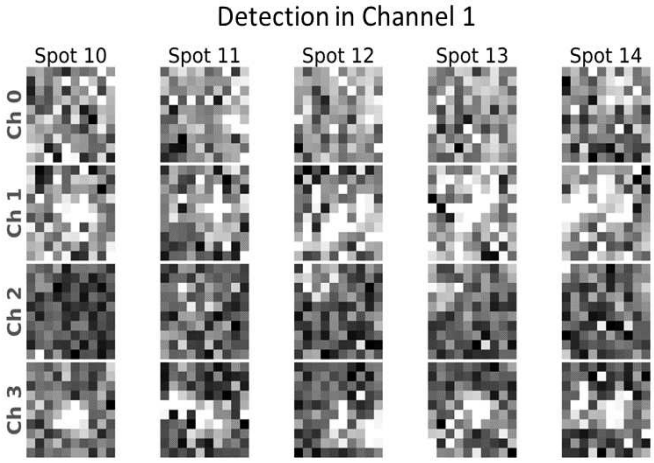

Figure A.5: Individual spot crops of the detected spots are shown in each channel. The spot IDs are given by the columns and the image channels are presented in the rows.

Appendix B

Stochastic Systems Identification Toolkit tutorial

The main function of the SSIT is to set up discrete models, run Stochastic Simulation Algorithm (SSA) trajectories, perform FSP calculations, conduct sensitivity analysis, load and fit data, and calculate Fisher Information. Tutorials and examples of each function are provided below.

B.1 Model Set Up

The first function of the SSIT is to set up a discrete stochastic model. This involves initializing the SSIT, defining the species in the model, specifying the initial conditions, defining the parameters, setting up the stoichiometry matrix, and establishing the propensity functions. The example demonstrated in this section will be based on the EGRNT model discussed in the paper. The model is initialized by the following code:

```
Model = SSIT;
```

Here I call the the SSIT code into the variable "Model" to make changes to the model without changing any parameters in the SSIT code itself. Any changes from this point will be made to above defined variable "Model." The species are defined in the line:

```
Model.species = {'x1';'x2';'x3'};
```

In this case, I define three species in the model. Adding another species would be done by adding x4 or others species separated by a comma within the curly brackets. The initial conditions are defined using the line:

```
Model.initialCondition = [0;0;0];
```

The number of values in this vector must be equal to the number of species. In this species, I set the initial species count for all species to 0. The propensity function is defined in the following line.

```
Model.propensityFunctions = {'(kcn0+kcn1*IDex)';'knc*x1';...
  'kon*x1*(2-x2)';'koff*x2';'kr*x2';'gr*x3'};
```

All of the variables must be defined before moving on to any of the other features. This is done in the following line below.

```
Model.parameters = ({'koff',0.14;'kon',0.01;'kr',1;'gr',0.01;...
  'kcn0',0.01;'kcn1',0.1;'knc',1;'r1',0.01});
```

Each parameter is written as a string for the variable name. Immediately after defining the variable name, the value of the variable needs to be given by separating the string with a comma. The initial variables can be a guess that can be updated in the fitting section later. Any variable that starts with an "i" character will be treated as an time varying input. Input variables are defined in the following line below.

```
Model.inputExpressions = {'IDex','(t>0)*exp(-r1*t)'};
```

Lastly, the stochiometry matrix is set up using the following code.

```
Model.stoichiometry = [ 1,-1, 0, 0, 0, 0;...
                        0, 0, 1,-1, 0, 0;...
                        0, 0, 0, 0, 1,-1];
```

It is important that the number of rows matches the number of species and the number of columns matches the number of reactions.

B.2 Running SSA trajectories

Running SSA trajectories can be done in a few lines. First, set the solution scheme to the SSA method by using the line:

```
Model.solutionScheme = 'SSA';
```

The number of simulations is controlled by using the following codes.

```
Model.ssaOptions.nSimsPerExpt = 200;
```

In this case I am running 200 trajectories. These trajectory can be done and saved in a CSV file using the code below.

```
Model.solve([],'Example.csv');
```

B.3 Finite State Projection Calculation

The FSP section of he codes allow the user to calculate the FSP which is a truncation of the CME. The FSP solution is used later on in the calculation of the sensitivity matrix so this must be run first. An example of running the FSP calculation for the EGRNT model is shown below.

```
Model.solutionScheme = 'FSP';
Model.fspOptions.fspTol = 1e-4;
Model.fspOptions.bounds=[];
[fspSoln,Model.fspOptions.bounds] = Model.solve;
```

The solution scheme is set to the FSP method, with an error tolerance of $1e-4$, and no bounds are specified for the FSP. Bounds can be set if the user has knowledge of the potential bounds of the FSP solution. Setting appropriate bounds can expedite the calculation time. The final line of code executes the FSP calculation using the defined FSP options from the previous line.

B.4 Sensitivity calculation and Analysis

This section calculates the sensitivity matrix of the model and enables the user to observe the sensitivity of parameters on the model fit. This is achieved using the following code:

```
% Set solutions scheme to FSP Sensitivity
Model.solutionScheme = 'fspSens';
% Solve the sensitivity problem
[sensSoln] = Model.solve(FSPsoln.stateSpace);
```

The solution scheme is set to the FSP sensitivity method and the last line computes the sensitivity matrix. To view the sensitivity plots, the following code is used.

```
% Plot marginal sensitivities
Model.makePlot(sensSoln,'marginals')
```

This example code will create the sensitivity plots. For each species in the model, the marginal distributions at the desired time point will be shown. The joint distribution of the selected species will also be plotted. An example of this plot for the EGRNT model is shown in the Figure B.1.

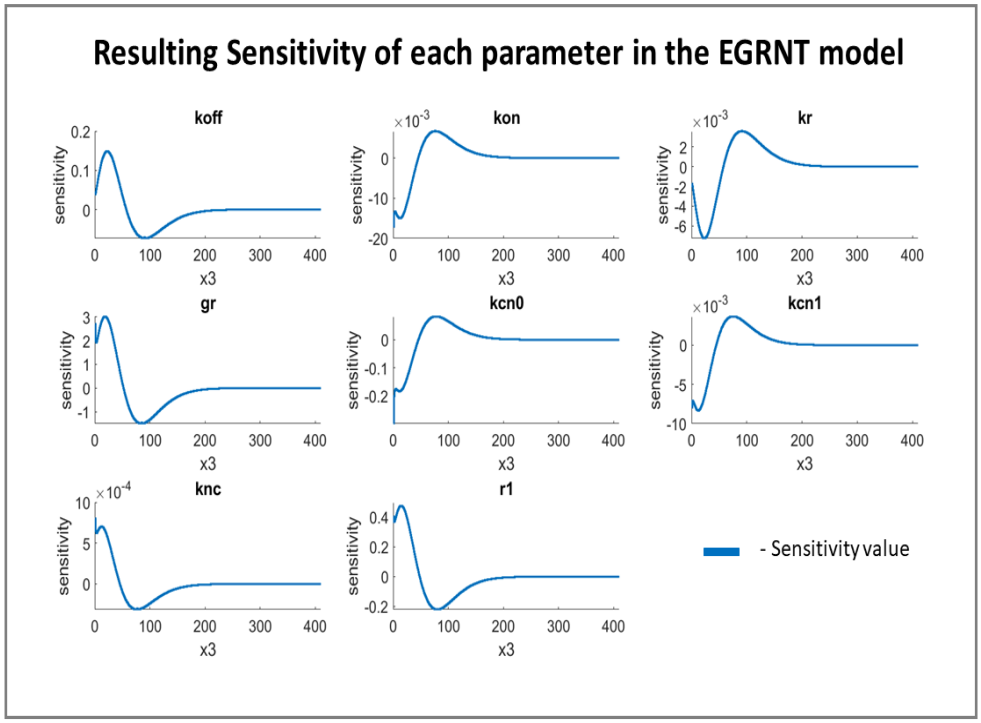

Figure B.1: Sensitivity analysis for the Chemical Master Equation The sensitivity of each parameter's effect on the distribution of DUSP1 mRNA is shown as calculated using the FSP approach in the SSIT. The x axis shows the count of mRNA, (species 'x3'). The y-axis shows the sensitivity of the distribution at the specific time (t=180 min) to the parameter listed in the subplot title. For example, an increase in the mRNA degradation parameter 'gr' leads to more probability mass (positive sensitivity) at low mRNA levels and less probability mass (negative sensitivity) at higher values of mRNA.

B.5 Fisher Information Computation

In this section the fisher information matrix will be computed from the sensitivity matrix. This is done with the following code:

```
fimResults = Model.computeFIM(sensSoln.sens);
[FIM,sFIMcov,fimMetrics] = Model.evaluateExperiment(fimResults,...
                                cellCounts);
```

The results of the FIM at each individual time point is calculated in the variable "fimResults." The total FIM is calculated with the amount of cells at each measurement point taken into account and is saved as the variable "FIM."

B.6 Data loading and fitting

The last section of the SSIT is the data loading and fitting section. This sections loads data from a CSV file. The columns of the file are the species and experiment replicas. The rows of the CSV file are the measurement time points of the experiment with its associated species count. Example code of loading the data into the model is shown by the code below.

```
Model = Model.loadData('../Example.csv',{'x1',...
                        'Example_Column_Name'});
```

The column name is required for the SSIT to associate that model with the species being fit. In the example above the species name would be "Example_Column_Name." Once the data is loaded into the model, the parameters can be fit to using the FSP method or Metropolis Hastings method. The FSP fit is done in the following code:

```
Model.fspOptions.fspTol = inf;
Model.fittingOptions.modelVarsToFit = 1:8;
fitOptions = optimset('Display','iter','MaxIter',100);
```

```
Model.parameters(1:8,2) = num2cell(Model.maximizeLikelihood(...
                          [],fitOptions));
```

The tolerance for the FSP is set to infinite. The model parameters that will be fit is given as 1 through 8. The number of attempts for fitting is set to a maximum of 100 iterations. The last line calculates the likelihood of each fit attempt and picks the parameter set that maximizes the likelihood. The two lines of codes below will allow the user to visualize the FSP fit and the the data.

```
Model.solutionScheme = 'FSP';
Model.makeFitPlot;
```

This will create plots for the distribution of the data and fit at each individual time-point as shown in figure B.2. A plot showing the model fit with the mean and standard deviation will be create as presented in figure B.3. An example of the figures generated from the EGRNT model is given in the figures below.

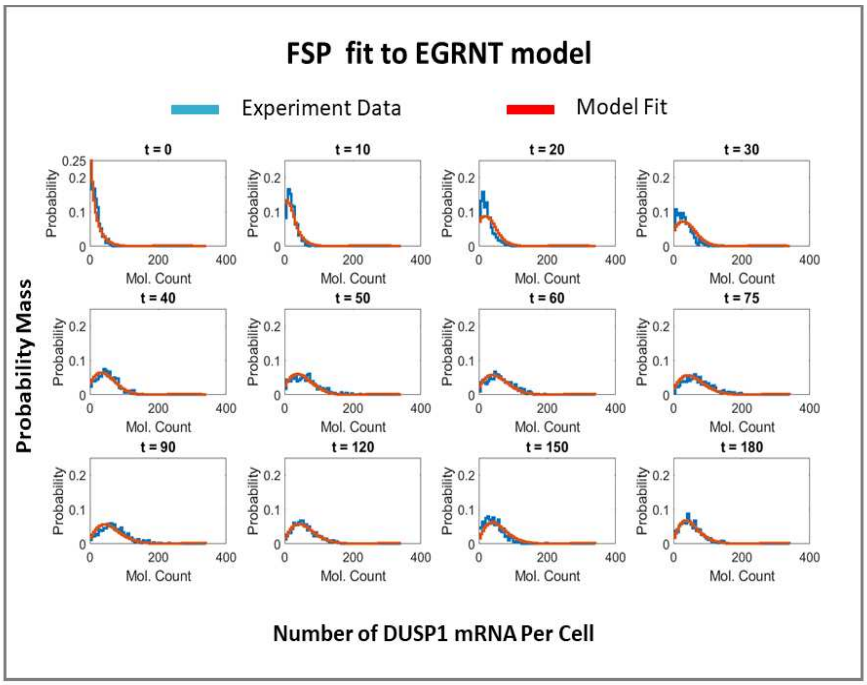

Figure B.2: Fit of the EGRNT model to smFISH data at 12 time points. The EGRNT gene regulatory model is parameterized and calibrated to accurately capture the temporal distribution of the DUSP1 mRNA population in smFISH experimental data. Leveraging the FSP methodology, a comprehensive model fit is generated to depict the dynamics of DUSP1 mRNA across various time points, where the mRNA species is visually represented in blue. The fitted EGRNT model is visually showcased in red, demonstrating the model's aptitude in reproducing the observed experimental outcomes. the data shown here consist of 9,639 cells over 12 different time points.

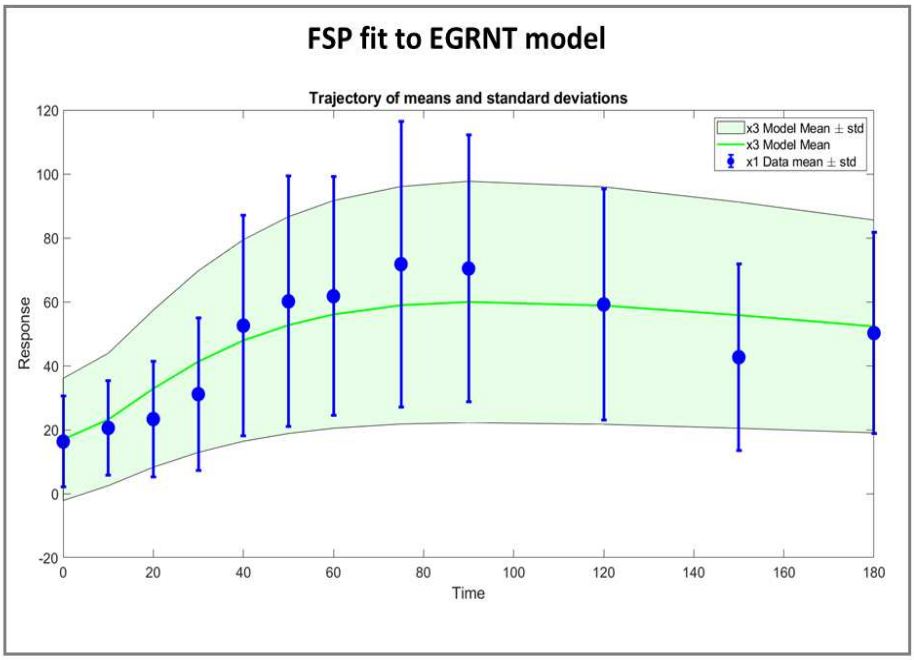

Figure B.3: DUSP1 mean and variance over time after Dex-stimulation. The mRNA mean (circles) and standard deviation (bars) from the experiment data are shown in blue and compared to the mean (green line) and standard deviation (green shading) from the model.

Running a fit with Metropolis Hastings is done in the following lines example code given below.

```
Model.fittingOptions.modelVarsToFit = 1:4;


MHOptions = struct('numberOfSamples',15000,'burnin',...
   100,'thin',1,...
  'useFIMforMetHast',true,'suppressFSPExpansion',true);


[bestParsFound,~,mhResults] = Model.maximizeLikelihood(...
[Model.parameters{Model.fittingOptions.modelVarsToFit,2}]',...
  MHOptions,'MetropolisHastings');
```

```
Model.parameters(Model.fittingOptions.modelVarsToFit,2) =...
num2cell(bestParsFound);
```

in this example, the model is being fit to parameters 1 through 4, the number of samples being run is 15000, the FIM is used in the calculation of the Metropolis Hastings samples and, the FSP is prevented from expanding in the calculation of the FIM. After generating all the samples, the best parameter set that maximizes the likelihood function is saved to the model. The below codes are used to create a plot of the MH samples with a 95% CI around the points.

```
Model.plotMHResults(mhResults);
```

This code produces figure B.4.

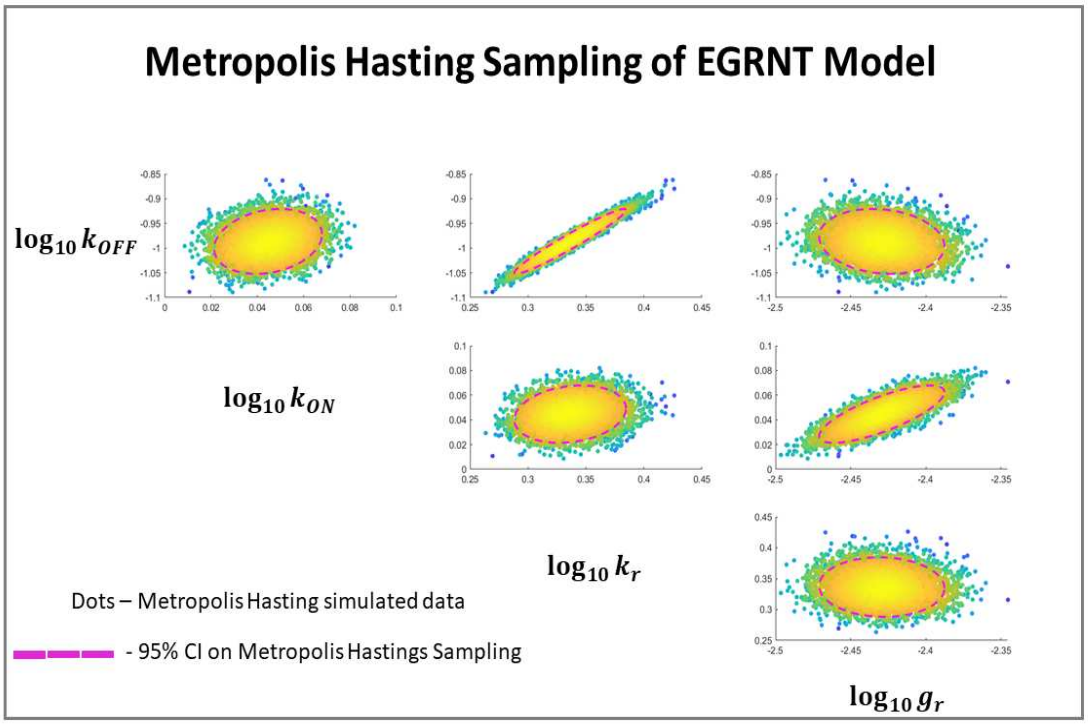

Figure B.4: Metropolis Hastings (MH) analysis of parameter uncertainty given experimental data. The posterior parameter space of the EGRNT model was sampled using MH to generate 15,000 parameter combinations depicted as individual dots. Colors denote the computed likelihood values from high (yellow) to low (blue). The dashed pink line denotes the 95% CI of the parameter estimates computed from the MH samples. Each panel shows the joint uncertainty of two parameters, and reveals that some combinations of parameters are highly correlated. For example, $\log_{10} k_{OFF}$ is linearly correlated to $\log_{10} k_r$, suggesting that the burst size k_r/k_{OFF} is tightly constrained by the data, but the actual values for each of these individual parameters is relatively uncertain.