Identification of Gene Regulation Models from Single-Cell Data

Lisa Weber[1], **William Raymond**[1,2], **Brian Munsky**[1,2,†]

[1] Department of Chemical and Biological Engineering, Colorado State University, Fort Collins, CO

[2] School of Biomedical Engineering, Colorado State University, Fort Collins, CO

† To whom correspondence should be addressed: munsky@colostate.edu

Abstract. In quantitative analyses of biological processes, one may use many different scales of models (e.g., spatial or non-spatial, deterministic or stochastic, time varying or at steady state) or many different approaches to match models to experimental data (e.g., model fitting or parameter uncertainty/sloppiness quantification with different experiment designs). These different analyses can lead to surprisingly different results, even when applied to the same data and the same model. We use a simplified gene regulation model to illustrate many of these concerns, especially for ODE analyses of deterministic processes, chemical master equation and finite state projection analyses of heterogeneous processes, and stochastic simulations. For each analysis, we employ MATLAB and PYTHON software to consider a time-dependent input signal (e.g., a kinase nuclear translocation) and several model hypotheses, along with simulated single-cell data. We illustrate different approaches (e.g., deterministic and stochastic) to identify the mechanisms and parameters of the same model from the same simulated data. For each approach, we explore how uncertainty in parameter space varies with respect to the chosen analysis approach or specific experiment design. We conclude with a discussion of how our simulated results relate to the integration of experimental and computational investigations to explore signal-activated gene expression models in yeast [1] and human cells [2]‡.

PACS numbers: 87.10.+e, 87.15.Aa, 05.10.Gg, 05.40.Ca,02.50.-r

Submitted to: *Phys. Biol.*

Keywords: gene expression models, stochastic gene expression, single-cell data, model inference

‡ A substantially reduced version of this manuscript will appear as an extended exercise in the textbook *Quantitative Biology: Theory, Computational Methods and Examples of Models*, edited by Brian Munsky, Lev S. Tsimring, and William S. Hlavacek, to be published by MIT Press, [3].

1. Introduction

Time-varying and bursty gene expression has been observed in many single-cell analyses, and many models have been developed to capture these dynamics (e.g., see recent reviews at [4, 5]). In this article, we define a small class of single-cell models which could potentially describe such processes, and we employ several computational analyses in attempts to identify these models and their parameters from various sets of simulated data. The specific analyses that we explore are introduced in next subsection. All simulated data and MATLAB codes used throughout this article can be downloaded from GITHUB at:

`https://github.com/MunskyGroup/qbio_Textbook`

The folder "CH30_Weber" within this repository includes all files referred to throughout this article.

1.1. A standard computational toolbox for quantitative biology

In the following paragraphs, we introduce a handful of computational tools that are frequently used in the analysis and fitting of quantitative gene regulation models.

Ordinary Differential Equations. One of the most common approaches to describe the dynamic behavior of gene regulation is to use coupled sets of Ordinary Differential Equations (ODEs). For example, the ODE related to mRNA transcription is a mathematical equation which describes the rate of change in the number of mRNA over time, taking into consideration both production and degradation of mRNA molecules. For most biological processes of interest, analytical solutions to the corresponding ODEs may not be possible, and numerical techniques may be necessary. In this article, we will deal with reaction rates that change over time, for which numerical integration is required.

Parameter fitting and the Metropolis-Hastings Algorithm. A common task in quantitative biology is to fit parameters of a model to match experimental data. Due to the stochastic nature of biological systems, in addition to the fact that experimental data is noisy and limited in quantity, exact estimation of parameters is impossible. Having a quantitative understanding of parameter uncertainties in light of existing data is crucial for predictive modeling. To accomplish this goal, Markov Chain Monte Carlo analyses, such as the Metropolis-Hastings (MH) algorithm, are frequently used to provide unbiased samples of parameter space and to estimate parameter uncertainties. In particular, the MH algorithm samples parameter space by adding random perturbations to parameters and then calculating the likelihood of the data, given the new parameter set. The new parameter set is then discarded unless the likelihood exceeds an appropriately chosen (albeit) random acceptance threshold. See [3] for a more in depth discussion of Monte Carlo analyses (Chapters 10 and 16), parameter estimation (Chapter 13), and sensitivity analyses (Chapter 14) as applied to models in quantitative biology.

The Stochastic Simulation Algorithm. Unlike most dynamical problems in science and engineering, gene regulation is inherently subject to discrete stochastic behavior arising from the single-molecule nature of genes within individual cells. The standard approach to analyze such discrete stochastic processes is to apply the kinetic Monte Carlo algorithm known as (Gillespie's) *Stochastic Simulation Algorithm* (SSA, [6, 7]), which uses a combination of stochastic reaction rates (i.e., propensity functions) and reaction rules (i.e., stoichiometry vectors) to simulate the stochastic trajectories. This algorithm updates the state of the system after each random reaction and assumes exponentially-distributed waiting times between every event. In this article, we apply the SSA to simulate many possible stochastic trajectories for the switching of gene activity states coupled with mRNA production and degradation.

The Finite State Projection Method. In addition to using the SSA to generate stochastic trajectories for the gene regulation processes, we also explore direct computations of the underlying probability distributions for these processes. These distributions evolve according to the infinite dimensional linear equation known as the Chemical Master Equation (CME, [8, 9]). Because the CME has no exact solution for most models, we employ the finite state projection (FSP) method to approximate the CME solution and to estimate the full joint probability distributions for gene switching behavior and transcript accumulation [10]. Several recent reviews of the FSP approach and its applications can be found at [11, 12, 13, 14]. A key advantage of the CME and FSP formulation is that it provides a systematic approach to compute and compare the likelihood of data arising from multiple models [1, 11]. With this in mind, we use the Metropolis-Hastings algorithm in conjunction with the FSP analysis to attain more precise constraints on the parameters when fit to simulated data.

Experiment Design. The successful identification of a model from data requires not just good data and appropriate modeling tools, but the experiment itself also needs to be conducted using a well-designed and informative condition [14, 15, 16, 17, 18, 19, 20]. With this in mind, we will explore the importance of experiment design, and we will show that different experiments (e.g., different input signals and different measurement times) and different analyses (e.g., ODEs or the FSP) yield different results for the constraint of system parameters.

In the following section, we introduce a specific gene regulation model and its underlying assumptions. Then, in Section 3, we demonstrate the implementation of the above tools to this model and different sets of simulated data. Finally, in Section 4, we provide a brief summary of our results and conclusions, and we introduce a user-friendly GUI toolbox to allow readers to reproduce all of the described analyses.

2. Gene Regulation Model Description

We begin by specifying a simple semi-mechanistic model for gene regulation. Figure 1 illustrates this model, which is a generalization of the standard bursting gene expression model discussed in [4, 5, 21]. The model consists of a single gene with three possible

states $\{S_1, S_2, \text{and } S_3\}$, each of which represents a different configuration of transcription factor binding or chromatin conformation. Similar 3-state models have been used previously in the context of time varying inputs to capture and predict signal-activated single-cell gene expression in yeast [1] and human [22] cells. The model assumptions are as described with additional detail in Subsection 2.1:

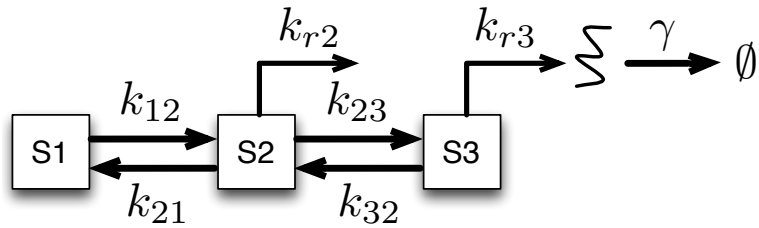

Figure 1. Schematic of a three-state gene regulation model. State 1 is transcriptionally inactive, and states 2 and 3 are transcriptionally active with transcription rates k_{r2} and k_{r3}. Parameters k_{ij} denote the (potentially time-varying) transition rates states S_i and S_j.

2.1. Assumptions of the 3-State Bursting Gene Expression Model

The assumed class of potential gene regulation models is defined as follows.

First, we define the conformational states $\{S_1, S_2, \text{ and } S_3\}$ of the gene and the production and degradation of mRNA. In this model, S_1 is a transcriptionally inactive ('OFF') state, whereas S_2 and S_3 are transcriptionally active ('ON') states. The transcription rates for S_2 and S_3 are k_{r2} and k_{r3}, respectively. Degradation of mRNA species, R, is assumed to be a first order process with rate γ. At the initial time ($t = 0$), we assume that all cells are in state S_1 and that there are no mRNA ($R = 0$).

Next, we define the transition dynamics that allow the gene to switch among these states. In the three-state model, there are four state transition events: $S_i \to S_j$, each with a corresponding transition rate of $k_{ij}(t)$. Three of the four transition rates are assumed to be constant with respect to time and the activating kinase: $k_{ij}(t) = k_{ij}(0)$. The gene is activated by a time-varying kinase, $Y_1(t)$, which acts as an input to the system. In practice, such signals can be controlled experimentally through modulation of external stimuli [1, 22, 23, 24]. For illustrative purposes, we assume a sinusoidal input signal given by:

$$Y_1(t) = \left\{ \begin{array}{ll} 0 & \text{for } (t \leq 5)(t \geq 70) \\ 1 - \cos\left(\frac{2\pi t}{30}\right) & \text{for } t \in (5, 70) \end{array} \right\}, \tag{1}$$

where the input begins at time $t = 5$ min and ends at time $t = 70$ min. We assume exactly one of the four transition rates is linearly dependent upon the activating kinase as follows:

$$k_{ij}(t) = \max\left\{0, k_{ij}(0) + \beta Y(t)\right\}. \tag{2}$$

For the combination $i < j$ (i.e., when the reaction leads to a state with a higher index), the value of $k_{ij}(0)$ is negative and β is positive. Conversely, when $i > j$ (i.e., when the reaction leads to a state with a lower index), the value of $k_{ij}(0)$ is positive and β is negative. This choice gives rise to a thresholded activation response in which $k_{ij}(t)$ is zero until the kinase exceeds or drops below a specific level. This definition leads to activation, either through direct activation when $i < j$ or indirect loss of repression when $i > j$.

One of our goals in this study is to determine if it is possible to identify the mechanism of action, i.e., to determine which k_{ij} depends upon the time-varying kinase signal. We define the model number, N_m, as the integer that specifies which mechanism depends upon the activating kinase. The model numbers and the corresponding input-dependent transition rates are as follows:

Model 1: $k_{12}(t)$; Model 2: $k_{23}(t)$; Model 3: $k_{21}(t)$; Model 4: $k_{32}(t)$.

Now that we have specified the assumptions needed to define the gene states, mRNA levels and the associated dynamics, we can write a set of four ODEs to describe this model:

$$\frac{dS_1}{dt} = -k_{12}S_1 + k_{21}S_2, \tag{3}$$

$$\frac{dS_2}{dt} = k_{12}S_1 - (k_{21} + k_{23})S_2 + k_{32}S_3, \tag{4}$$

$$\frac{dS_3}{dt} = k_{23}S_2 - k_{32}S_3, \tag{5}$$

$$\frac{dR}{dt} = k_{r2}S_2 + k_{r3}S_3 - \gamma R. \tag{6}$$

These equations are easily integrated in MATLAB using the stiff ODE solver, ODE15S.

Finally, for parameter estimation, we assume that all rates have an independent prior given by a lognormal distribution with a logarithmic mean of $\mathbb{E}\{\log_{10}(\lambda)\} = 0$ and a wide logarithmic variance of $\mathbb{E}\{(\log_{10}(\lambda))^2\} = 4$. Having defined the reactions and priors on all parameters, in the next section we will generate some simulated data and attempt to fit our model, explore uncertainty in parameter space, and make model predictions in light of these datasets.

3. Results

The goal of our study is to determine if simulated data is sufficient to identify the model and its parameters, and if the identified parameters can be used to predict the system's response under new experimental conditions. All parameter sets are defined in Section 2.1 and are presented in the following vector form:

$$\mathbf{\Lambda} = [k_{12},\ k_{23},\ k_{21},\ k_{32},\ k_{r2},\ k_{r3},\ \beta,\ \gamma].$$

3.1. Simulation of ODE Models

We first used a deterministic approach and solved the simple system of ODEs listed in Eqs. 3 - 6 to capture the bulk gene regulation dynamics. For a given model and parameter set, these ODEs generates the mean gene expression as a function of time. The input signal for these analyses is given in Eqn. 1, and the two arbitrary parameter sets are given in the caption of Fig. 2.

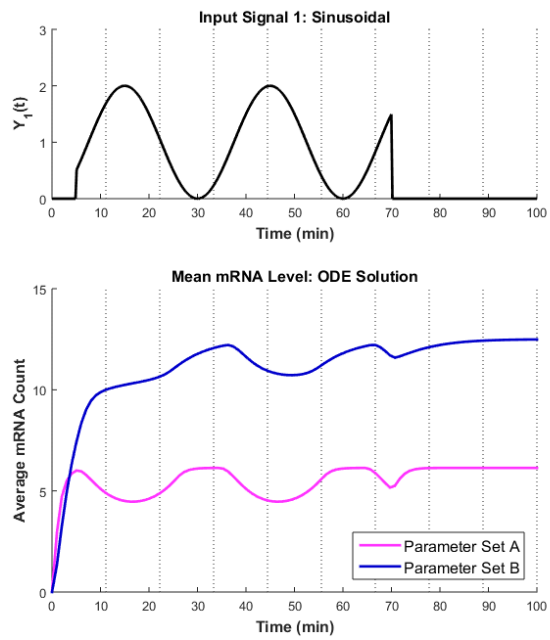

Figure 2. (Top) Input signal, $Y_1(t)$ as defined in Eqn. 1 versus time. (Bottom) Deterministic ODE solution for the mean gene expression as a function of time for the three-state gene regulation model shown in Fig. 1, as outlined in Section 3.1. Analyses are shown for Model 2 (k_{23} is time dependent), and the two illustrated parameter sets are $\mathbf{\Lambda_A} = [3,\ -0.4,\ 2.5,\ 1,\ 9,\ 2,\ 1,\ 0.8]$ and $\mathbf{\Lambda_B} = [1,\ -0.6,\ 1,\ 0.2,\ 5,\ 2,\ 1,\ 0.2]$.

The mean gene expression as a function of time for Model 2 (i.e., where k_{23} is time-dependent) is shown in Fig. 2. This analysis can reveal qualitative insight about how rate parameters control the underlying process dynamics. For example, Fig. 2 illustrates that faster rates for transcription and degradation (k_{r2} and γ) enable gene expression to track the input signal more closely (Fig. 2, magenta line), whereas slower rates result in greater averaging and lag behind the input signal (Fig. 2, blue line). Biologically, evolution could tune the rates of these reactions to allow cells to respond more quickly to important fluctuations (e.g., sustained stresses) while rejecting faster environmental disturbances (e.g., short-lived transient fluctuations). In addition, such dependence of response dynamics on parameters makes it feasible to perform an inverse analysis (i.e., to estimate parameters from quantitative observations of response dynamics) as discussed in the following subsection.

3.2. Maximum Likelihood Estimation for ODE Models

Next, we use the ODE model defined in Eqns. 3-6 and search parameter space to find a parameter set that maximizes the likelihood of a simulated dataset. This simulated data can be found in the GitHub folder at "CH30_Weber/Matlab_Codes." This data has been generated under a specific input signal, $Y_1(t)$, which is given in Eqn. 1 and shown in Fig. 2(A). Additionally, the simulated data has been 'measured' at ten different time points, where all measurements were generated from independent single-cell trajectories. From this mRNA trajectory data, we obtained the mean, standard deviation, and standard error of the mean, as illustrated in Fig. 3.

Before we can fit the model to the data, we must first compute the likelihood of the data given the model's parameters. Consider a set of experimental data, including N_i cells at each time t_i. Let $\tilde{R}_{ij}$ denote the number of mRNA in the jth cell at the ith time point. The measured sample mean at each time is given by:

$$\mu_S(t_i) = \frac{1}{N_i}\sum_{j=1}^{N_i}\tilde{R}_{ij}, \tag{7}$$

and the squared standard error (i.e., standard variance of the sample mean) can be measured as:

$$\sigma^2_{SEM}(t_i) = \frac{1}{N_i(N_i-1)}\sum_{j=1}^{N_i}\left(\tilde{R}_{ij} - \mu_S(t_i)\right)^2. \tag{8}$$

Under an assumption of independent Gaussian errors for all cells, the likelihood to observe the measured sample means can be estimated using the normal distribution:

$$Prob(\mu_S(t_i) \mid \mathbf{\Lambda}) = \frac{1}{\sqrt{2\pi\sigma^2_{SEM}(t_i)}}\exp\left(-\frac{(\mu_S(t_i) - R(t_i,\mathbf{\Lambda}))^2}{2\sigma^2_{SEM}(t_i)}\right). \tag{9}$$

Our assumed lognormal prior distribution ($\mathbb{E}\{log_{10}(\lambda)\} = 0$, $\mathbb{E}\{(log_{10}(\lambda))^2\} = 4$) for each parameter can be written as:

$$Prob(\lambda_k) = A\exp\left(-\frac{\log_{10}(\lambda_k)^2}{8}\right), \tag{10}$$

where A is a normalization constant. The probability of a parameter set $\mathbf{\Lambda}$, given the data, can be written according to Bayes' rule, (i.e., $P(\mathbf{\Lambda}|\mathbf{D}) = P(\mathbf{D}|\mathbf{\Lambda})P(\mathbf{\Lambda})/P(\mathbf{D})$, where $\mathbf{\Lambda}$ denotes the model and $\mathbf{D}$ denotes the data) as:

$$Prob(\mathbf{\Lambda}|\mathbf{D}) = \frac{1}{P(\mathbf{D})}\left(\prod_{i=1}^{N_t} Prob(\mu_S(t_i) \mid \mathbf{\Lambda}) \cdot \prod_{k=1}^{8} Prob(\lambda_k)\right). \tag{11}$$

Here, N_t is the number of time points, and $P(\mathbf{D})$ is a normalization constant. Taking the logarithm of $Prob(\mathbf{\Lambda}|\mathbf{D})$ and substituting in the expressions for $Prob(\mu_S(t_i) \mid \mathbf{\Lambda})$ and $Prob(\mathbf{\Lambda}|\mathbf{D})$ above yields:

$$\log(Prob(\mathbf{\Lambda}|\mathbf{D})) = C - \sum_{i=1}^{N_t}\frac{(\mu_S(t_i) - R(t_i,\mathbf{\Lambda}))^2}{2\sigma^2_{SEM}(t_i)} - \sum_{k=1}^{8}\frac{\log_{10}(\lambda_k)^2}{8}, \tag{12}$$

where the first summation is the chi-squared function (i.e., the likelihood of the data given the model) and the second summation relates to the prior probability to select those parameters in the absence of any data.

For small data sets, the second term will dominate, but as more data is collected, the first summation will take on greater importance. We fit all four possible models (each with a different input-dependent transition rate) by maximizing the likelihood of the model given the data as specified in Eqn. 12. For this, we use standard optimization routines including MATLAB's *fminsearch*, which attempts to find the unconstrained minimum of the objective function using a Nelder-Mead simplex algorithm [25, 26]. We have also utilized a customized simulated annealing algorithm. To improve efficiency, we converted all parameters to positive values (since the signs of all parameters are known for each model), and we conducted all searches in logarithmic space. Moreover, we restarted all searches using multiple random initial guesses (thirty per model), and we selected the local minima that provide the greatest likelihoods of the model given the data. Figure 3 shows the result for such a fit using the model in which the rate k_{23} was assumed to be time-dependent (Model 2). The corresponding best-fit parameter set for Model 2 is given in the caption to Fig. 3.

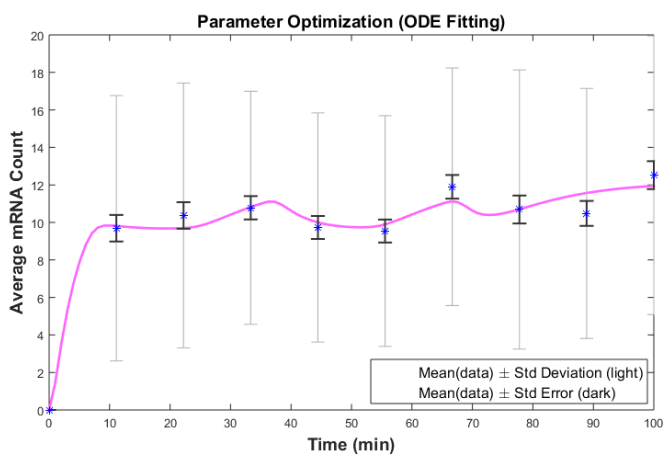

Figure 3. Mean gene expression as a function of time for Model 2 after performing optimization to determine the parameter set with the best fit to the averaged data. The light error bars represent the mean of the data ± one standard deviation. The dark error bars represent the mean of the data ± the standard error (100 independent samples per time point). The resulting parameter set with the best fit to the data was found to be: $\mathbf{\Lambda_{Fit}} = [1.7254,\ -1.0095,\ 0.1070,\ 0.0998,\ 3.0449,\ 2.2485,\ 1.6607,\ 0.2360]$.

3.3. Parameter Uncertainty Estimation for ODE Models

When maximizing the likelihood of the model given the data, we saw that it was possible to find many combinations of models and parameters that match exceptionally well to the simulated data. We next explored the uncertainty in parameter space in

order to understand how well the averaged experimental data was able to constrain the parameters of the chosen model. To do this, we implemented the Metropolis-Hastings (MH) algorithm to explore posterior parameter uncertainty in the model, given the simulated experimental data and the likelihood function in Eqn. 12. In the MH proposal function, we select each parameter with probability of 0.5, and for these we change their logarithmic value by adding a Gaussian-distributed perturbation with a mean of zero and a standard deviation of 0.2. In MATLAB, this can be scripted as:

*x = x + 0.2*randn(size(x)).*(rand(size(x))>0.5)*

where 'x' is the vector of parameters in logarithmic space.

Figure 4 shows an example of the scatter plot for k_{r2} and γ for Model 2. In this case, we used a total chain length of 750,000 with a burn-in of 250,000 steps and a thin rate of 10. Since the MH algorithm is random, each finite length run of this algorithm produces different results. In practice, the MH algorithm should be run until convergence (e.g., until multiple independent chains with different initial starting parameter guesses converge, within some specified metric and tolerance). Our analysis yielded the following average parameter values:

$$\mathbf{\Lambda}_{\mathrm{MH}} = [7.2673,\ 0.1012,\ 0.3457,\ 0.2072,\ 7.2478,\ 0.0706,\ 2.1825,\ 0.0923]$$

and the following large covariances for the parameters:

$$\Sigma_{\mathbf{\Lambda}} = \begin{bmatrix} 1.5475 & 0.3667 & 0.0827 & -0.0145 & -0.3329 & -0.0032 & 0.5451 & -0.2689 \\ 0.3667 & 1.9240 & -0.0690 & -0.1318 & -0.1888 & -0.2652 & 0.9059 & -0.6151 \\ 0.0827 & -0.0690 & 2.3328 & -0.3474 & 0.3830 & -0.3328 & 0.2021 & -0.3072 \\ -0.0145 & -0.1318 & -0.3474 & 2.7010 & 0.1564 & 0.1150 & -0.6002 & 1.5233 \\ -0.3329 & -0.1888 & 0.3830 & 0.1564 & 0.3285 & 0.0851 & -0.1926 & 0.2883 \\ -0.0032 & -0.2652 & -0.3328 & 0.1150 & 0.0851 & 1.5912 & -0.1730 & 0.6291 \\ 0.5451 & 0.9059 & 0.2021 & -0.6002 & -0.1926 & -0.1730 & 1.6170 & -0.9621 \\ -0.2689 & -0.6151 & -0.3072 & 1.5233 & 0.2883 & 0.6291 & -0.9621 & 1.6560 \end{bmatrix}. \tag{13}$$

In Fig. 4, the ellipse depicts the 90% confidence interval in the parameters, with the (dubious) assumption that the posterior parameter distribution is a multivariate lognormal distribution with the mean equal to $\mathbf{\Lambda}_{\mathrm{MH}}$ and a covariance matrix equal to $\Sigma_{\mathbf{\Lambda}}$. Clearly, with the data available, *the model is not well identified despite excellent fits to the average data*. This is an illustration of the effect of model sloppiness, where parameters are highly uncertain in some directions, yet potentially well-constrained in other directions within parameter space [27, 28]. In Section 3.5 we will explore how stochastic analyses can dramatically improve model identification even when using the same model and the same data.

3.4. Simulation of Stochastic Gene Expression Models

Next, we simulate many stochastic trajectories for the gene regulation model, thus generating potential single-cell mRNA distributions from the model. To do this, we define the stoichiometry matrix and the propensity functions for each of the four possible

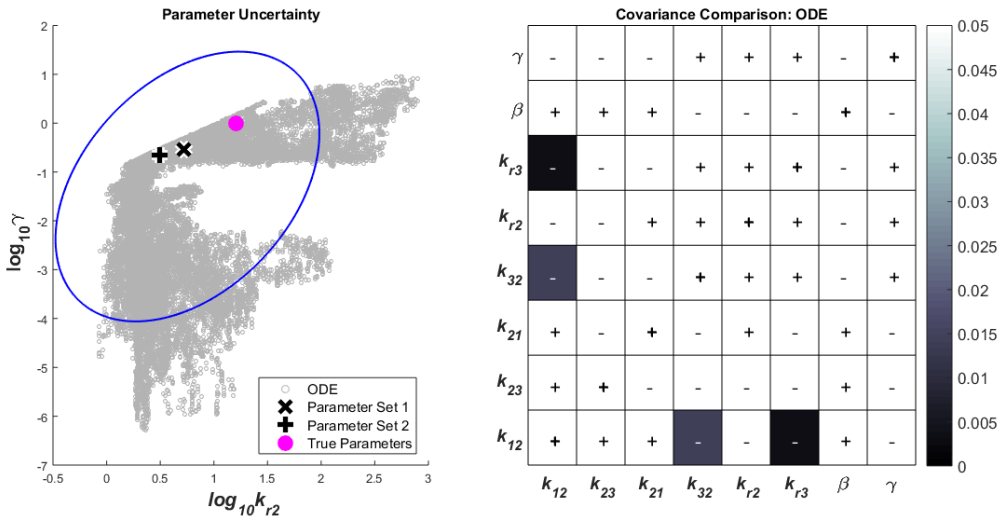

Figure 4. (Left) Parameter uncertainty in logspace for parameters k_{r2} and γ, as determined by applying the Metropolis-Hastings algorithm to the ODE likelihood function (Eqn. 12). The ellipse indicates a 90% confidence interval, assuming a log Gaussian posterior. The magenta circle indicates the true k_{r2} and γ parameter values. Two similarly good parameter sets were selected from this chain: $log_{10}(\mathbf{\Lambda_1}) = [0.0705,\ -2.2859,\ -0.1497,\ -0.3444,\ 0.7181,\ -1.9591,\ -1.1805,\ -0.5304]$ and $log_{10}(\mathbf{\Lambda_2}) = [0.1797,\ 0.5551,\ -0.7058,\ -1.1098,\ 0.4949,\ 0.3303,\ 0.6639,\ -0.6531]$. For these parameters sets, k_{r2} and γ for are represented by the black $\mathbf{x}$ and $+$, respectively. (Right) Comparison of the *thresholded* covariances for each parameter combination from the ODE MH search, with the covariance values thresholded at ± 0.05. All covariance values whose magnitudes exceed 0.05 are shown in white. The **(+)** and **(-)** symbols indicate positive and negative covariances, respectively.

models. Unlike the standard SSA, one of the propensity functions in these models is a time-varying function. In this case, it is necessary to adapt the SSA in a manner similar to that discussed in [29, 30]. This modification corresponds to adding a fast reaction to the system, which has a large propensity function ($w_1 = 100/\text{min}$), but which does not change the state of the system ($\mathbf{S}(:,1) = \mathbf{0}$). We ran 1000 SSA trajectories for each of the four possible models, and we made histograms of the number of mRNA in different cells at each point in time (using the resulting best fit parameters from Section 3.2).

A single SSA trajectory is shown in the left panel of Fig. 5. In the middle panel of Fig. 5, we plot the average of multiple sets of 1000 SSA simulations versus time along with the ODE solution. In addition, the right panel of Fig. 5 shows different FSP distributions for the two parameter sets (given in the caption for Fig. 4) and the simulated data at the same, specific time point. Despite producing excellent fits to the averaged data (Fig. 5, middle), neither parameter set captures the bimodal behavior of the data as shown in Fig. 5, right. This again shows that fitting the ODE was not a sufficient strategy to extract all available information from our simulated single-cell data

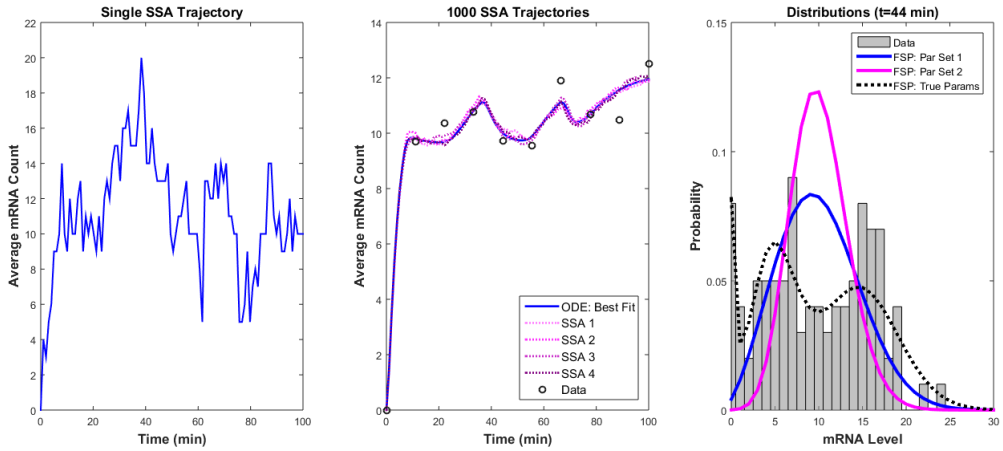

Figure 5. (Left) Results from a single SSA trajectory for Model 2 using the best fit parameters, $\mathbf{\Lambda_{Fit}}$, obtained in Section 3.2. (Middle) Mean gene expression as a function of time from multiple runs of 1000 SSA trajectories using the best fit parameters compared to the deterministic solution and the experimental data. (Right) Full distributions at 44 minutes for the two parameter sets, $\mathbf{\Lambda_1}$ and $\mathbf{\Lambda_2}$, compared to the data and the exact distribution for the true parameters at the same time point. The true parameters are $\mathbf{\Lambda_{True}} = [0.20,\ -0.076,\ 0.050,\ 0.050,\ 16.0,\ 5.0,\ 0.10,\ 1.0]$ or equivalently $\mathbf{\log_{10}(\Lambda_{True})} = [-0.6990,\ -1.1192,\ -1.3010,\ -1.3010,\ 1.2041,\ 0.6990,\ -1.0000,\ 0]$.

set, and the result suggests that fitting full distributions could lead to improvements in parameter identification. We examine this possibility further in the next subsection.

3.5. Parameter Uncertainty Estimation for FSP Models

To improve upon the parameter uncertainty estimation results, we next utilized the Finite State Projection (FSP) approach to re-explore the uncertainty in parameter space, given the single-cell data. To accomplish this, we write the CME in the matrix ODE form:

$$\frac{d\mathbf{P}}{dt} = \mathbf{A}(t)\mathbf{P}(t), \tag{14}$$

where $\mathbf{P} = [p_1, p_2, \ldots]$ is the vector of probabilities of each state of the stochastic process, and $\mathbf{A}(t)$ is known as the *infinitesimal generator matrix*. See [11, 12, 13, 14] for examples on the construction and evaluation of this FSP formulation (Eqn. 14) and the GitHub folder at "CH30_Weber/Matlab_Codes" for specific FSP codes for this section. Equation 14 can then be integrated in time to compute full probability distributions for the gene regulation dynamics. With these probabilities, the likelihood of the data given the model is easily computed using the expression:

$$\log\left(L_{\text{FSP}}(\mathbf{D}|\mathbf{\Lambda})\right) = \sum_{i=1}^{N_t}\sum_{j=0}^{N_r} n_{ij}\log(p(j|t_i,\mathbf{\Lambda})), \tag{15}$$

where N_r is the maximum number of mRNA observed in a single cell, n_{ij} is the number of cells with exactly j mRNA at the ith timepoint, and $p(j|t_i, \mathbf{\Lambda})$ is the probability to have j mRNA at that time given the model parameters in $\mathbf{\Lambda}$, as discussed in [11, 14]. The corresponding log-likelihood of the model given the data can be written (using our previous definition of the prior) as:

$$\log\left(L_{\text{FSP}}(\mathbf{\Lambda}|\mathbf{D})\right) = C - \log(P(\mathbf{D})) + \sum_{i=1}^{N_t}\sum_{j=0}^{N_r} n_{ij}\log(p(j|t_i, \mathbf{\Lambda})) - \sum_{k=1}^{8}\frac{\log_{10}(\lambda_k)^2}{8}. \quad (16)$$

With this new likelihood function for the model given the data, we again implemented the MH algorithm to explore parameter uncertainty in the model, given the full information within the simulated single-cell data. In comparison to the parameter uncertainty estimation using the ODE analysis, our analysis using the FSP approach yielded substantially lower covariances for the parameters, as seen in the covariance matrix:

$$\Sigma_{\mathbf{\Lambda}} = \begin{bmatrix} 0.0079 & 0.0049 & 0.0084 & -0.0008 & 0.0099 & 0.0105 & 0.0007 & 0.0100 \\ 0.0049 & 0.5976 & 0.0097 & -0.0133 & 0.0010 & 0.0022 & 0.0837 & 0.0008 \\ 0.0084 & 0.0097 & 0.0148 & 0.0021 & 0.0036 & 0.0042 & 0.0026 & 0.0034 \\ -0.0008 & -0.0133 & 0.0021 & 0.0089 & -0.0064 & -0.0067 & 0.0022 & -0.0067 \\ 0.0099 & 0.0010 & 0.0036 & -0.0064 & 0.0314 & 0.0328 & -0.0012 & 0.0322 \\ 0.0105 & 0.0022 & 0.0042 & -0.0067 & 0.0328 & 0.0348 & -0.0011 & 0.0336 \\ 0.0007 & 0.0837 & 0.0026 & 0.0022 & -0.0012 & -0.0011 & 0.0223 & -0.0014 \\ 0.0100 & 0.0008 & 0.0034 & -0.0067 & 0.0322 & 0.0336 & -0.0014 & 0.0330 \end{bmatrix} \quad (17)$$

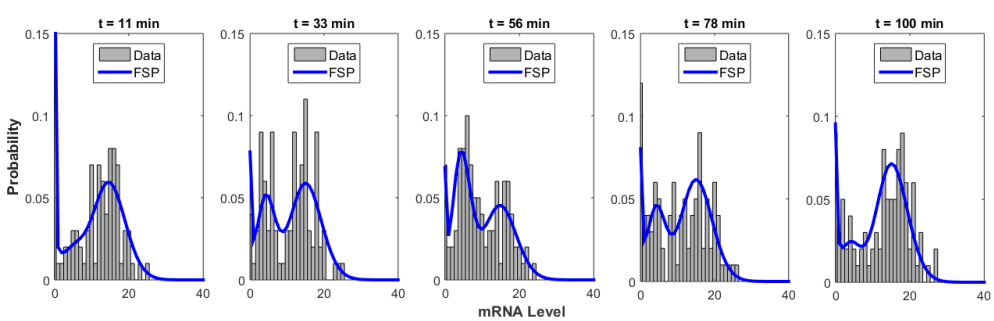

Figure 6. Data distributions compared with FSP distributions for Model 2 (using the FSP best fit parameter values) at specific time points.

The FSP distribution results are presented in Figs. 6 and 7. Figure 6 shows that the FSP distributions resulting from the identified parameters match closely to the data distributions at the time points presented and capture the bimodal behavior of the data. Figure 7 shows that there is substantially less parameter uncertainty when the Metropolis-Hastings search is performed with the FSP instead of the chi-squared function. Figure 8 shows the true parameter values compared to the average estimated parameter values for Model 2 from the ODE and FSP searches of the parameter space.

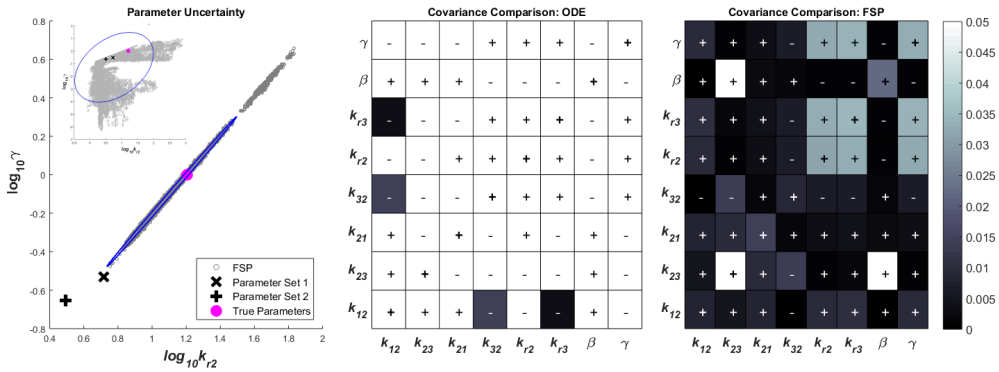

Figure 7. (Left) Parameter uncertainty for k_{r2} and γ, as determined using the FSP likelihood function (Eqn. 16) in the Metropolis-Hastings analysis of Model 2. The ellipse indicates a 90% confidence interval. The magenta circle indicates the true k_{r2} and γ values. (Left Inset) To provide a scale reference, the FSP-MH results are compared to the ODE search results for the same data set. (Middle and Right) Comparison of the *thresholded* covariances for each parameter combination from the ODE MH search (Middle) and the FSP MH search (Right), using the thresholding procedure described in the caption of Fig. 4 caption. The **(+)** indicates a positive covariance and **(-)** indicates a negative covariance.

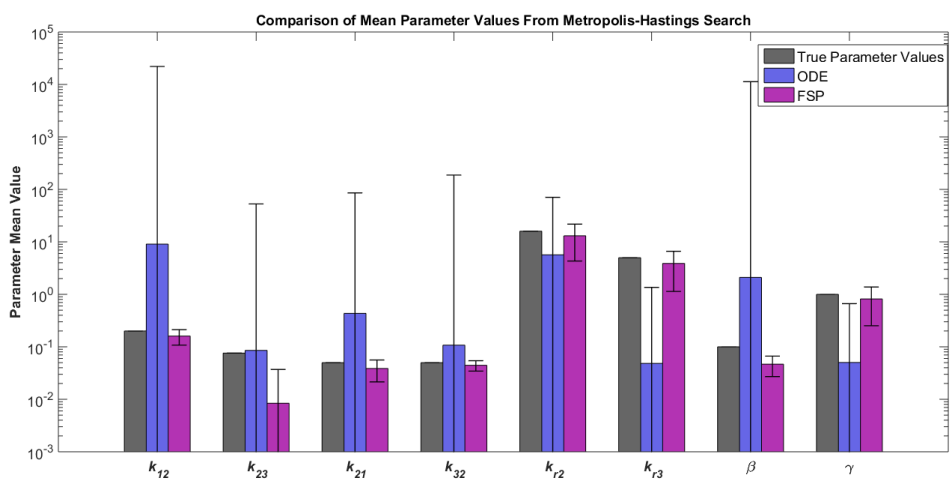

Figure 8. Bar plot showing the true parameter values compared to the mean parameter values from the ODE MH search and the FSP MH search for Model 2. The error bars represent the mean parameter value ± one standard deviation.

For nearly all of the parameters, the FSP search yields substantially closer estimates of the true parameter values and has much lower standard deviations for the parameter estimates as compared to ODE-based search. The only exception to this observation, for parameter $k_{23}(0)$, is easily explained by the the fact that this reaction (Eqn. 2) is saturated and therefor unobservable for much of the system trajectory. It is also worth noting that, in addition to providing better parameter estimates and lower uncertainty,

the FSP-based analysis required far shorter MH chains to explore parameter space before converging around the global minimum.

3.6. Importance of Experiment Design

Up to this point, we have demonstrated that the type of analysis chosen (i.e., an ODE- or FSP-based likelihood function) to examine experimental data is crucial in model and parameter identification. We now briefly explore how different experimental designs can affect parameter identification results. To demonstrate this concern, we fit the data and performed MH searches of parameter space for multiple different experiments (i.e., different input signals). Specifically, we compared the original input signal, given in Eqn. 1, with two additional input signals, which are as follows:

$$Y_2(t) = \left\{ \begin{array}{ll} 0 & \text{for } (t \leq 5)(t \geq 70) \\ 2 & \text{for } t \in (5, 70) \end{array} \right\}, \tag{18}$$

$$Y_3(t) = \left\{ \begin{array}{ll} 0 & \text{for } (t \leq 5)(t \geq 70) \\ \left(\frac{2}{65}t\right) - \left(\frac{2}{13}\right) & \text{for } t \in (5, 70) \end{array} \right\}, \tag{19}$$

where both additional input signals begin at time $t = 5$ min and end at time $t = 70$ min, similar to $Y_1(t)$.

Figures 9 and 10 illustrate how the three different experimental designs (i.e., three different input signals) lead to dramatically different parameter uncertainties, using the ODE and the FSP likelihood functions, respectively. In all figures, we assume the exact same amount of data, but the effects on which parameters are tightly identified and which remain uncertain is non-trivially dependent on the input signal. Moreover, input signals that are best for one type of analysis may not be the best choice for a different analysis (for example, compare the right panels of Fig. 9 and 10).

To further demonstrate how important experimental design considerations are to obtain accurate prediction of gene regulation dynamics, we make model predictions of FSP distributions at different times and for multiple different experiments (i.e., different input signals). In these predictions, we use the best-fit parameter sets from the FSP-based searches for all four models (each with a different time-dependent transition rate and obtained using the original input signal given in Eqn. 1). The potential input signals are $Y_2(t)$ (Eqn. 18), $Y_3(t)$ (Eqn. 19), and $Y_4(t)$, given by:

$$Y_4(t) = \left\{ \begin{array}{ll} 0 & \text{for } (t \leq 5)(t \geq 70) \\ 1 - \cos\left(\frac{2\pi t}{15}\right) & \text{for } t \in (5, 70) \end{array} \right\}. \tag{20}$$

Figure 11 shows predictions for these 3 potential input signals (left, middle, right columns) for two potential time points (middle, bottom rows), assuming the 4 different models that were identified using the original experiment (different colors). By examining these predictions, we can gain insight into which experiments might be more informative. For example, at $t = 30$ min (middle row), an experiment with input signal

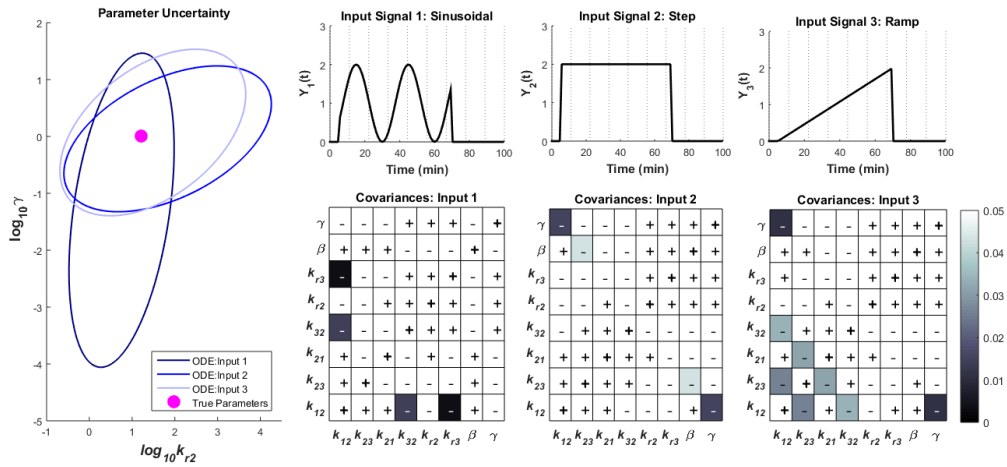

Figure 9. (Left) Uncertainty in k_{r2} and γ given three different input signals, as determined by the ODE-based MH analysis of Model 2. The ellipses indicate a 90% confidence interval. The magenta circle indicates the true k_{r2} and γ parameter values. (Top right) Three different input signals and (Bottom right) the corresponding comparison of the *thresholded* covariances for each parameter combination from the ODE MH analysis. The **(+)** indicates positive covariances and **(-)** indicates negative covariances.

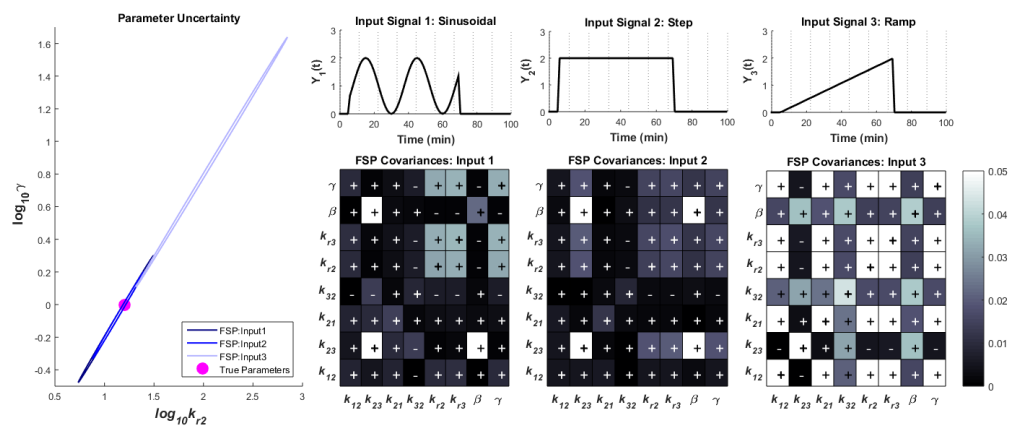

Figure 10. (Left) Uncertainty in k_{r2} and γ given three different input signals, as determined by the FSP-based MH analysis of Model 2. The ellipses indicate a 90% confidence interval. The magenta circle indicates the true k_{r2} and γ parameter values. (Top right) Three different input signals and (Bottom right) the corresponding comparison of the *thresholded* covariances for each parameter combination from the FSP MH analysis. The **(+)** indicates positive covariances and **(-)** indicates negative covariances.

$Y_2(t)$ (left column) or $Y_3(t)$ (middle column) would enable us to differentiate easily between the models, whereas input signal $Y_4(t)$ (right column) is unlikely to differentiate between the models. Furthermore, measuring the output at $t = 30$ min is clearly more informative than at $t = 100$ min (compare middle and bottom rows, Fig. 11).

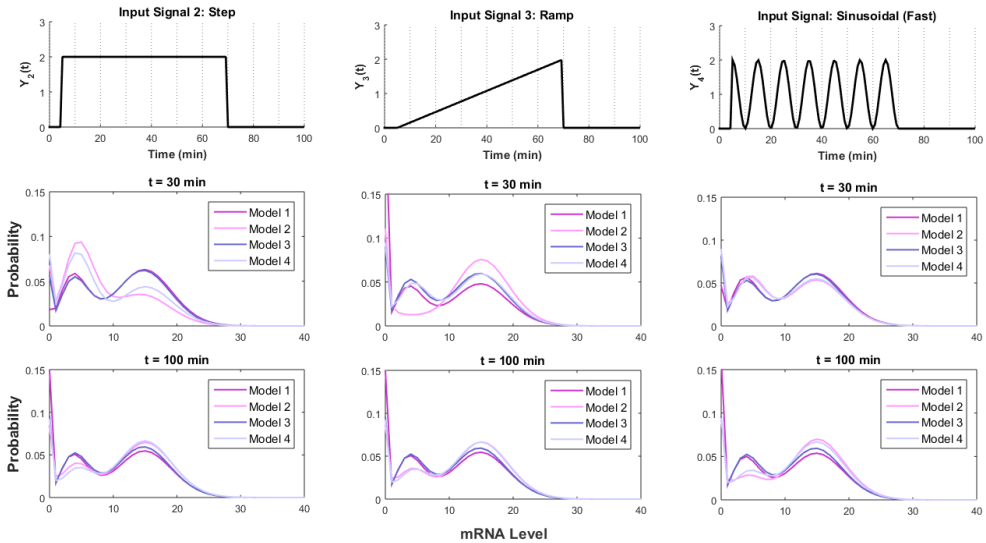

Figure 11. Predictions of the FSP distributions for all four model numbers at two different time points for three different experiments (with input signals shown in Eqn. 18 - 20). Each model uses the best parameter set found during the fit to the original input signal in Eqn. 1.

4. Summary and Conclusions

In this article, we explored the application of various computational techniques (such as ODE analyses [31, 32, 33, 34, 35, 36], stochastic simulations [37, 7, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48], and finite state projections [14, 11, 12, 10, 49, 50, 8, 4, 51, 52]) to understand the dynamics of single-cell gene regulation. In particular, we focussed on a simplified three-state gene regulation model, and we used a combination of simulated single-cell data and different modeling procedures to explore how well one could identify models and parameter uncertainties from discrete stochastic data. Additionally, we discussed the importance of experiment design and illustrated how different experiments can be more informative than others for model identification. Moreover, our analyses show that fitting ODEs to dynamical data may not be adequate for parameter identification and discerning parameter uncertainty (even when models are relatively simple), and we discussed the fact that fitting full distributions (e.g., using an FSP likelihood function) may lead to substantial improvements, even when considering the same model and same data. These results argue that the integration of discrete stochastic models and single-cell data could open new possibilities for improved model identification, even in cases where traditional OIDE analyses have failed due to excessive parameter sloppiness.

To make these approaches more accessible, we have developed a GUI, which is freely available on GitHub in the folder "CH30_Weber/GUI". The GUI, shown in Fig. 12, provides an opportunity for users to specify any input and model number and to perform the analyses outlined throughout, evaluate results, and generate figures. In

addition, the GUI enables the automatic generation of simulated data using SSA for user-specified parameter sets and model, or import a previously generated data file to perform the analyses outlined in this exercise. The GUI requires PYTHON and the packages: NUMPY, SCIPY, and MATPLOTLIB.

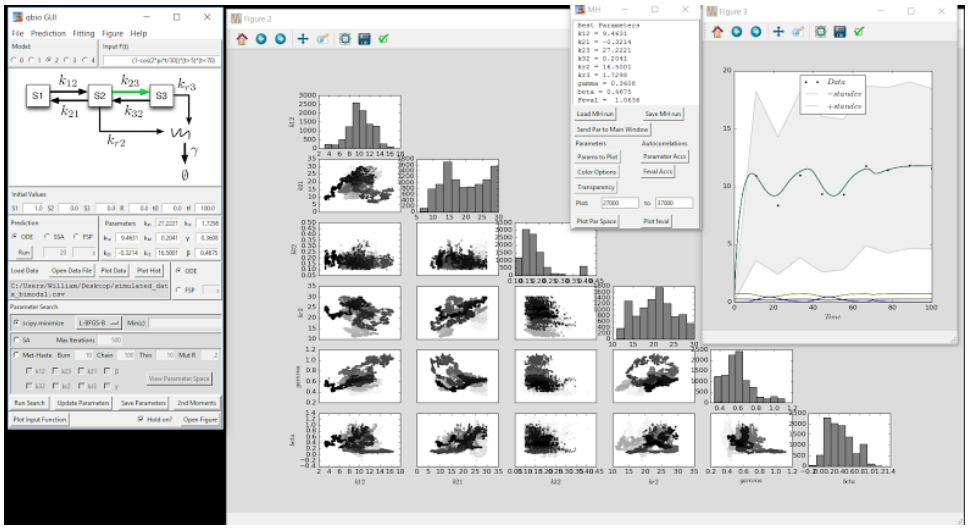

Figure 12. Screenshot of the provided GUI, which can be utilized to perform all analyses outlined throughout, including generating simulated data sets and producing desired plots.

[1] 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, pp. 584–587, Jan. 2013.

[2] 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.

[3] B. Munsky, W. S. Hlavacek, and L. S. Tsimring, *Quantitative Biology: Theory, Computational Methods and Examples of Models*. MIT Press, 1st ed., In Press.

[4] B. Munsky, G. Neuert, and A. van Oudenaarden, "Using Gene Expression Noise to Understand Gene Regulation," *Science*, vol. 336, pp. 183–187, Apr. 2012.

[5] B. Munsky and G. Neuert, "From analog to digital models of gene regulation," *Physical Biology*, vol. 12, no. 4, p. 045004, 2015.

[6] D. T. Gillespie, "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions," *Journal of Computational Physics*, vol. 22, no. 4, pp. 403–434, 1976.

[7] D. T. Gillespie, "Exact stochastic simulation of coupled chemical reactions," *The journal of physical chemistry*, vol. 81, no. 25, pp. 2340–2361, 1977.

[8] D. A. McQuarrie, "Stochastic approach to chemical kinetics," *Journal of Applied Probability*, vol. 4, no. 3, pp. 413–478, 1967.

[9] N. van Kampen, *Stochastic Processes in Physics and Chemistry*. Elsevier, 3 ed., 2007.

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

[11] 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, p. 074101, Aug. 2016.

[12] 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, Sept. 2015.

[13] K. N. Dinh and R. B. Sidje, "Understanding the finite state projection and related methods for solving the chemical master equation," *Physical biology*, vol. 13, no. 3, p. 035003, 2016.

[14] Z. Fox and B. Munsky, "Stochasticity or noise in biochemical reactions," in *Quantitative Biology: Theory, Computational Methods and Examples of Models* (B. Munsky, L. S. Tsimring, and W. S. Hlavacek, eds.), ch. 5, Cambridge, Massachusetts: MIT Press, 2018.

[15] K. NienaĆtowski, T. Jetka, and M. Komorowski, "Sensitivity analysis," in *Quantitative Biology: Theory, Computational Methods and Examples of Models* (B. Munsky, L. S. Tsimring, and W. S. Hlavacek, eds.), Cambridge, Massachusetts: MIT Press, 2018.

[16] T. Jetka, K. NienaĆtowski, and M. Komorowski, "Experimental design," in *Quantitative Biology: Theory, Computational Methods and Examples of Models* (B. Munsky, L. S. Tsimring, and W. S. Hlavacek, eds.), Cambridge, Massachusetts: MIT Press, 2018.

[17] B. C. Daniels, M. Dobrzynski, and D. Fey, "Parameter estimation, sloppiness, and model identifiability," in *Quantitative Biology: Theory, Computational Methods and Examples of Models* (B. Munsky, L. S. Tsimring, and W. S. Hlavacek, eds.), Cambridge, Massachusetts: MIT Press, 2018.

[18] J. Bernie J. Daigle, "Bayesian parameter estimation and markov chain monte carlo," in *Quantitative Biology: Theory, Computational Methods and Examples of Models* (B. Munsky, L. S. Tsimring, and W. S. Hlavacek, eds.), Cambridge, Massachusetts: MIT Press, 2018.

[19] B. Mélykúti, E. August, A. Papachristodoulou, and H. El-Samad, "Discriminating between rival biochemical network models: three approaches to optimal experiment design," *BMC Systems Biology*, vol. 4, no. 1, p. 1, 2010.

[20] J. F. Apgar, J. E. Toettcher, D. Endy, F. M. White, and B. Tidor, "Stimulus Design for Model Selection and Validation in Cell Signaling," *PLoS Computational Biology*, vol. 4, p. e30, Feb. 2008.

[21] J. Peccoud and B. Ycart, "Markovian Modeling of Gene-Product Synthesis," *Theoretical Population Biology*, vol. 48, pp. 222–234, Oct. 1995.

[22] 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, pp. 75–83, July 2014.

[23] D. Muzzey, C. A. Gómez-Uribe, J. T. Mettetal, and A. van Oudenaarden, "A Systems-Level Analysis of Perfect Adaptation in Yeast Osmoregulation," *Cell*, vol. 138, pp. 160–171, July 2009.

[24] J. T. Mettetal, D. Muzzey, C. Gomez-Uribe, and A. van Oudenaarden, "The Frequency Dependence of Osmo-Adaptation in Saccharomyces cerevisiae," *Science*, vol. 319, pp. 482–484, Jan. 2008.

[25] J. Nelder and R. Mead, "A Simplex Method for Function Minimization," *The Computer Journal*, vol. 7, no. 4, p. 308, 1965.

[26] G. B. Dantzig, *Linear Programming and Extensions.* Princeton, NJ: Princeton University Press, 1963.

[27] 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, pp. 1871–78, Oct. 2007.

[28] B. C. Daniels, Y.-J. Chen, J. P. Sethna, R. N. Gutenkunst, and C. R. Myers, "Sloppiness, robustness, and evolvability in systems biology.," *Current Opinion in Biotechnology*, vol. 19, no. 4, pp. 389–395, 2008.

[29] M. Voliotis, P. Thomas, C. G. Bowsher, and R. Grima, "The extra reaction algorithm for stochastic simulation of biochemical reaction systems in fluctuating environments," in *Quantitative Biology: Theory, Computational Methods and Examples of Models* (B. Munsky, L. S. Tsimring, and W. S.

Hlavacek, eds.), Cambridge, Massachusetts: MIT Press, 2018.

[30] M. Voliotis, P. Thomas, R. Grima, and C. G. Bowsher, "Stochastic simulation of biomolecular networks in dynamic environments," *PLoS computational biology*, vol. 12, no. 6, p. e1004923, 2016.

[31] D. Fey, M. Dobrzynski, and B. N. Kholodenko, "Modeling with ordinary differential equations," in *Quantitative Biology: Theory, Computational Methods and Examples of Models* (B. Munsky, L. S. Tsimring, and W. S. Hlavacek, eds.), Cambridge, Massachusetts: MIT Press, 2018.

[32] E. D. Conrad and J. J. Tyson, *System Modeling in Cellular Biology: From Concepts to Nuts and Bolts*, ch. Modeling Molecular Interaction Networks with Nonlinear Ordinary Differential Equations, p. 97ff. The MIT Press, 2006.

[33] B. N. Kholodenko, O. V. Demin, G. Moehren, and J. B. Hoek, "Quantification of short term signaling by the epidermal growth factor receptor.," *J Biol Chem*, vol. 274, pp. 30169–30181, Oct 1999.

[34] E. Klipp, W. Liebermeister, C. Wierling, A. Kowald, H. Lehrach, and R. Herwig, *Systems Biology.* Wiley, 2013.

[35] B. Ingalls, *Mathematical Modeling in Systems Biology: An Introduction.* MIT Press, 2013.

[36] J. Keener and J. Sneyd, *Mathematical Physiology*, vol. 8 of *Interdisciplinary Applied Mathematics.* New York: Springer-Verlag, second ed., 2001.

[37] R. Bertolusso and M. Kimmel, "Kinetic monte carlo analyses of discrete biomolecular events," in *Quantitative Biology: Theory, Computational Methods and Examples of Models* (B. Munsky, L. S. Tsimring, and W. S. Hlavacek, eds.), Cambridge, Massachusetts: MIT Press, 2018.

[38] M. A. Gibson and J. Bruck, "Efficient exact stochastic simulation of chemical systems with many species and many channels," *Journal of Physical Chemistry A*, vol. 104, pp. 1876–1889, 2000.

[39] A. Slepoy, A. P. Thompson, and S. J. Plimpton, "A constant-time kinetic Monte Carlo algorithm for simulation of large biochemical reaction networks.," *J Chem Phys*, vol. 128, p. 205101, May 2008.

[40] D. Gillespie, "Approximate accelerated stochastic simulation of chemically reacting systems," *The Journal of Chemical Physics*, vol. 115, no. 4, pp. 1716–1733, 2001.

[41] D. Gillespie, "The chemical Langevin equation," *The Journal of Chemical Physics*, vol. 113, no. 1, pp. 297–306, 2000.

[42] D. Gillespie and L. Petzold, "Improved leap-size selection for accelerated stochastic simulation," *The Journal of Chemical Physics*, vol. 119, no. 16, pp. 8229–8234, 2003.

[43] Y. Cao, D. Gillespie, and L. Petzold, "Efficient step size selection for the tau-leaping simulation method," *The Journal of Chemical Physics*, vol. 124, no. 4, p. 044109, 2006.

[44] Y. Cao, D. Gillespie, and L. Petzold, "Adaptive explicit-implicit tau-leaping method with automatic tau selection," *The Journal of Chemical Physics*, vol. 126, no. 22, p. 224101, 2007.

[45] Y. Cao, D. Gillespie, and L. Petzold, "The Slow-Scale Stochastic Simulation Algorithm," *The Journal of chemical physics*, vol. 122, Jan. 2005.

[46] C. Rao and A. P. Arkin, "Stochastic Chemical Kinetics and the Quasi-Steady-State Assumption: Application to the Gillespie Algorithm," *The Journal of chemical physics*, vol. 118, pp. 4999–5010, 2003.

[47] L. Harris and P. Clancy, "A "partitioned leaping" approach for multiscale modeling of chemical reaction dynamics," *J. Chem. Phys.*, vol. 125, p. 144107, 2006.

[48] Y. Cao, D. T. Gillespie, and L. R. Petzold, "Efficient step size selection for the tau-leaping simulation method," *The Journal of chemical physics*, vol. 124, no. 4, p. 044109, 2006.

[49] K. Burrage, M. Hegland, S. Macnamara, and R. Sidje, "A Krylov-Based Finite State Projection Algorithm for Solving the Chemical Master Equation Arising in the Discrete Modelling of Biological Systems," *Proc. of The A.A.Markov 150th Anniversary Meeting*, pp. 21–37, 2006.

[50] B. Munsky and M. Khammash, "The Finite State Projection Approach for the Analysis of Stochastic Noise in Gene Networks," *IEEE Trans. Automat. Contr./IEEE Trans. Circuits and Systems: Part 1*, vol. 52, pp. 201–214, Jan. 2008.

[51] B. Munsky and M. Khammash, "Transient Analysis of Stochastic Switches and Trajectories with Applications to Gene Regulatory Networks," *IET Systems Biology*, vol. 2, no. 5, pp. 323–333, 2008.

[52] B. Munsky, "Modeling cellular variability," *Quantitative Biology: From Molecular to Cellular Systems*, p. 233, 2012.