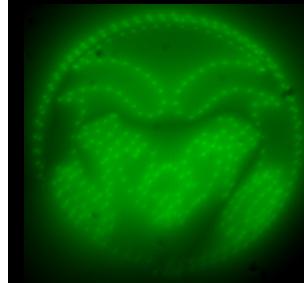


The Stochastic system Identification Toolkit (SSIT): Efficient, accessible quantitative modeling of discrete stochastic biochemical processes



Brian Munsky
Chemical and Biological Engineering
School of Biomedical Engineering
Colorado State University

ACS Fall 2024 Meeting

Denver CO, USA

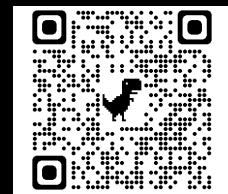
August 22, 2024

munsky@colostate.edu



To REPRODUCE THIS PRESENTATION,
FOLLOW THESE STEPS:

- (1) Clone GitHub Repository:
<https://github.com/MunskyGroup/SSIT>
- (2) Launch MATLAB R2021a-R2023b
- (3) Open/Run script “**Examples/example_DUSP1_Regulation.m**” in Matlab Editor



Summary of what has been seen in the past using good experiments:



- * Transcription and Translation occur in **stochastic bursts** that can be measured at single-molecule resolution.
- * Different bursting mechanisms alter the **temporal and spatial statistics** of cellular heterogeneity in subtle ways.
- * By testing multiple models in different stress or drug response conditions, fluctuation fingerprints reveal insight into which bursting mechanisms are affected under what perturbations.

This is great, but...



Single-cell experiments are **expensive**, **noisy**, and there are **vast numbers** of possible experiment designs or user-supplied inputs.



Experiment Design Considerations

- Number of cells
- Sampling times or periods
- Choice of fluorophore(s)
- Number and placement of probes
- Choice of which genes, mRNA, or protein to measure
- Inducer/drug concentrations and delivery times

Measurement Error Considerations

- Microscope resolution
- Image processing errors (segmentation, spot detection, track linking)
- Photobleaching
- Autofluorescence
- Camera exposure time
- Light source power and wavelength and optical filters
- Delays due to drug/inducer diffusion or nuclear import

1. Models need to account for ***cellular heterogeneities*** and ***experimental uncertainties***.
2. Experiments should be ***designed to minimize uncertainty*** about mechanisms or parameters of interest.



Single-cell experiments are **expensive**, **noisy**, and there are **vast numbers** of possible experiment designs or user-supplied inputs.



Experiment Design Considerations

- Number of cells
- Sampling times or periods
- Choice of fluorophore(s)
- Number and placement of probes
- Choice of which genes, mRNA, or protein to measure
- Inducer/drug concentrations and delivery times

Measurement Error Considerations

- Microscope resolution
- Image processing errors (segmentation, spot detection, track linking)
- Photobleaching
- Autofluorescence
- Camera exposure time
- Light source power and wavelength and optical filters
- Delays due to drug/inducer diffusion or nuclear import

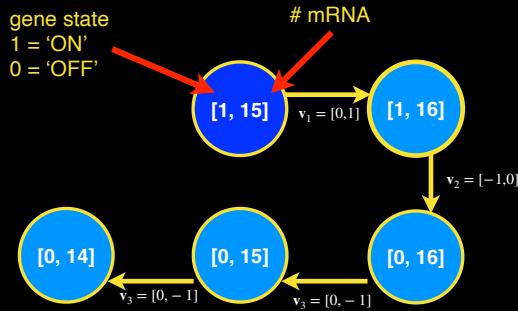
1. Models need to account for ***cellular heterogeneities*** and ***experimental uncertainties***.
2. Experiments should be ***designed to minimize uncertainty*** about mechanisms or parameters of interest.



The Markov Model description of gene expression



- Non-stationary distributions of stochastic gene regulation can often be reproduced using discrete state Markov processes.
- At any time, the state of the system is defined by an integer population vector: $\mathbf{x} \in \mathbb{Z}^N$
- Reactions are represented by propensity (rate) functions $\{w_\mu\}$ and stoichiometry vectors $\{\mathbf{v}_\mu\}$.



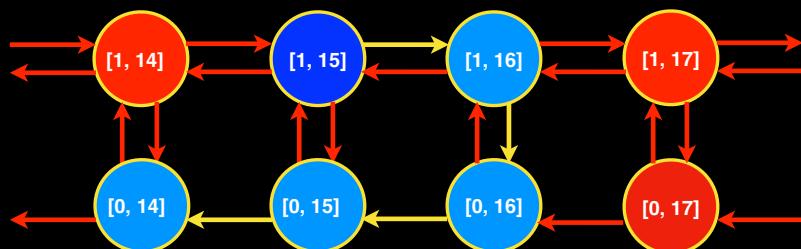
<https://github.com/MunskyGroup/SSIT>



The Markov Model description of gene expression



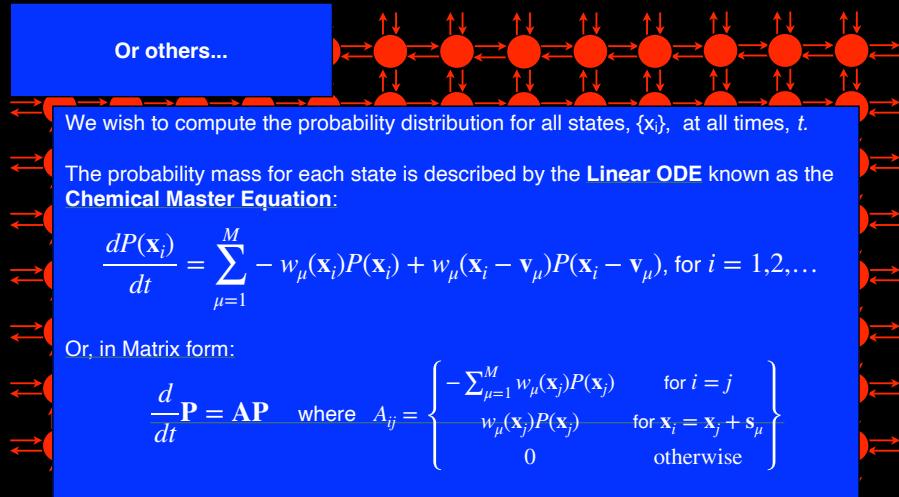
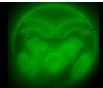
- Non-stationary distributions of stochastic gene regulation can often be reproduced using discrete state Markov processes.
- At any time, the state of the system is defined by an integer population vector: $\mathbf{x} \in \mathbb{Z}^N$
- Reactions are represented by propensity (rate) functions $\{w_\mu\}$ and stoichiometry vectors $\{\mathbf{v}_\mu\}$.
- These reactions are random, others could have occurred:



<https://github.com/MunskyGroup/SSIT>



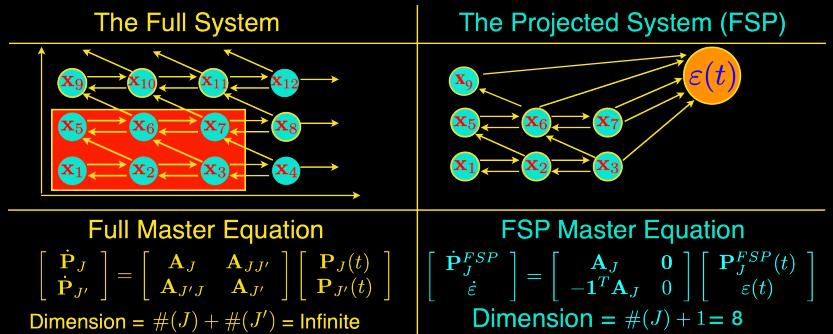
The Markov Model description of gene expression



<https://github.com/MunskyGroup/SSIT>



The Finite State Projection (FSP) approach to solve the CME



1) **Strict lower bound** on solution: $\mathbf{P}_J(t) \geq \mathbf{P}_J^{FSP}(t)$

2) **Exact error** of solution: $\left\| \begin{bmatrix} \mathbf{P}_J(t) \\ \mathbf{P}_{J'} \end{bmatrix} - \begin{bmatrix} \mathbf{P}_J^{FSP}(t) \\ \mathbf{0} \end{bmatrix} \right\|_1 = \varepsilon_J(t)$

3) **Monotonic** convergence: $\varepsilon_{J_1}(t) \geq \varepsilon_{J_2}(t)$ for any $J_1 \subseteq J_2$

4) Convergence of **Likelihood Function**:

$$\sum d_j \log P_j^{FSP}(\theta) \leq \log L(D; \theta) \leq \max_{\substack{|\mathbf{f}|_1 = 1 \\ f_j \geq 0}} \left(\sum d_j (\log P_j^{FSP}(\theta) + f_j \varepsilon) \right)$$

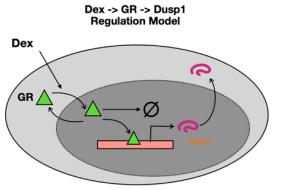
<https://github.com/MunskyGroup/SSIT>

(1-3) Munsky et al, JCP 2006
(4) Fox et al, JCP 2016

Example — FSP Analysis of Bursting DUSP1 mRNA Expression



Dex > GR > Dusp1 Regulation Model



Propensities and Stoichiometries

$$w_1 = k_{ON}I_{GR}(t)g_{OFF}, \quad s_1 = [-1, 1, 0]$$

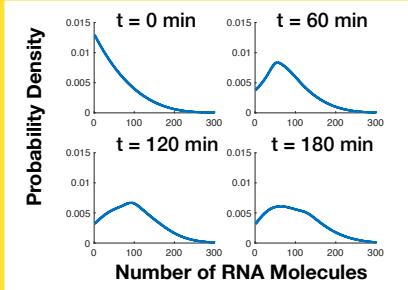
$$w_2 = k_{OFF}g_{ON}, \quad s_2 = [1, -1, 0]$$

$$w_3 = k_rg_{ON}, \quad s_3 = [0, 0, 1]$$

$$w_4 = \gamma_r r, \quad s_4 = [0, 0, -1]$$

Time Varying Input Signal:

$$I_{GR}(t) = \begin{cases} 1, & \text{for } t < 0 \\ 1 + a_1 e^{-r_1 t} (1 - e^{-r_2 t}), & \text{for } t \geq 0 \end{cases}$$



MATLAB CODE

```

%% Create SSIT Model
Modell = SSIT; % Create blank SSIT model.
Modell.species = {'offGene';'onGene';'rna'}; % Set species names.
Modell.initialCondition = [2;0;0]; % Set initial condition

%% Define propensity functions and input signals:
Modell.propensityFunctions = {'(kOn*kOff*offGene)';'kOff*onGene';'kr*onGene';'gr*rna'};
Modell.inputExpressions = {'IGR', '1+al*exp(-r1*t)*(1-exp(-r2*t))*((t>0))'};
Modell.stoichiometry = [-1,1,0;1,-1,0;0,0,1]; % Define stoichiometry
Modell.parameters = {[kOff',0.014; kOn',0.002; kr',1; gr',0.004;...
    'al',20;r1',0.04;r2',0.1]}; % Specify parameter guesses
Modell.fspOptions.initApproxSS = true; % Set Initial Distribution to Steady State.
Modell.summarizeModel % Print visual summary of model
Modell.formPropensitiesGeneral('ToyDUSP1Model'); % Generate model codes

%% Solve using the FSP approach
Modell.solutionscheme = 'FSP'; % Select FSP solution Scheme
Modell.fspOptions.fpTol = 1e-4; % Set FSP 1-norm error tolerance.
Modell.fspOptions.bounds(4:6) = [2,2,400]; % Guess initial bounds on FSP StateSpace
Modell.tSpan = linspace(0,180,301);
[ModelfPSoln,Modell.fspOptions.bounds] = Modell.solve; % Solve Model
Modell.makePlot(ModelfPSoln,'marginal1',[1:100:301],false,[1,2,3],{'linewidth',2}); % Plot distributions
Modell.makePlot(ModelfPSoln,'margmove',[1],false,[101],{'linewidth',2},{'movie.mp4',[1,1,0.015],[2,3]}); % Generate Movie of Distributions.

```

<https://github.com/MunskyGroup/SSIT>

Using FSP to Compute Sensitivity to Model Parameters



- To find the sensitivities to a parameter θ_i , we can differentiate the FSP:

$$\frac{d}{d\theta_i} \left(\frac{d\mathbf{P}(\theta, t)}{dt} \right) = \frac{d}{d\theta_i} (\mathbf{A}(\theta)\mathbf{P}(\theta, t))$$

- Switch the derivatives on the left and apply product rule on the right:

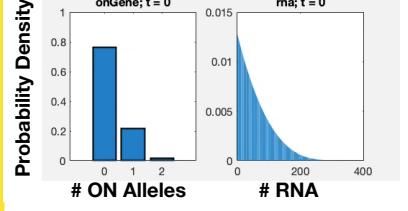
$$\frac{d}{dt} \left(\frac{d\mathbf{P}(\theta, t)}{d\theta_i} \right) = \frac{d\mathbf{A}(\theta)}{d\theta_i} \mathbf{P}(\theta, t) + \mathbf{A}(\theta) \frac{\partial \mathbf{P}(\theta, t)}{\partial \theta_i}$$

- We are then left with a coupled set of **Linear ODES** for the FSP and its sensitivity:

$$\frac{\partial}{\partial t} \begin{bmatrix} \mathbf{P}(\theta, t) \\ \frac{\partial \mathbf{P}(\theta, t)}{\partial \theta_i} \end{bmatrix} = \begin{bmatrix} \mathbf{A}(\theta) & \mathbf{0} \\ \frac{\partial \mathbf{A}(\theta)}{\partial \theta_i} & \mathbf{A}(\theta) \end{bmatrix} \begin{bmatrix} \mathbf{P}(\theta, t) \\ \frac{\partial \mathbf{P}(\theta, t)}{\partial \theta_i} \end{bmatrix}$$

- or more simply:

$$\frac{\partial}{\partial t} \begin{bmatrix} \mathbf{P} \\ \mathbf{s}_i \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \partial_{\theta_i} \mathbf{A} & \mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{P} \\ \mathbf{s}_i \end{bmatrix}$$

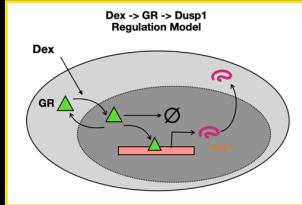




Example – FSP Sensitivity Analysis for Bursting DUSP1 mRNA Expression



We calculate the sensitivity of all species to all parameters and at all time points.



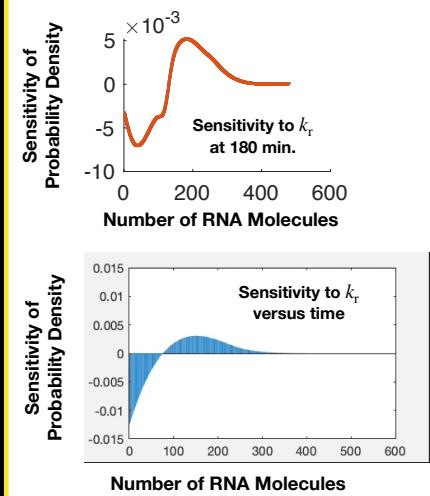
Propensities $w_1 = k_{ON}I_{GR}(t)g_{OFF}$, $w_2 = k_{OFF}g_{ON}$, $w_3 = k_rg_{ON}$, $w_4 = \gamma_r r$,	Stoichiometries $s_1 = [-1, 1, 0]$, $s_2 = [1, -1, 0]$, $s_3 = [0, 0, 1]$, $s_4 = [0, 0, -1]$
---	---

Time Varying Input Signal:

$$I_{GR}(t) = \begin{cases} 1, & \text{for } t < 0 \\ 1 + a_1 e^{-r_1 t} (1 - e^{-r_2 t}), & \text{for } t \geq 0 \end{cases}$$

MATLAB CODE

```
% Solve Sensitivity using FSP
Modell.solutionScheme = 'fspSens'; % Set solutions scheme to FSP Sensitivity
ModellSensSoln = Modell.solve(ModellFSPsoln.stateSpace); % Solve the sensitivity problem
Modell.makePlot(ModellSensSoln,'marginals',[1],false,[4,5,6],{'linewidth',2}); % Plot marginal sensitivities
Modell.makePlot(ModellSensSoln,'marginMovie',[1],false,[101],{'linewidth',2}, 'sensMovie.mp4',
[-1,-1,-0.015;1,1,0.015],[3],[3]) % Plot sensitivity movie
```



<https://github.com/MunskyGroup/SSIT>

Measurements are Uncertain

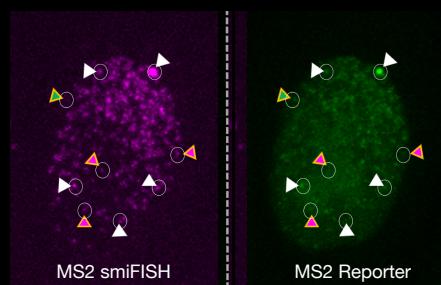
No labeling strategy, microscope, or image processing tool is perfect.

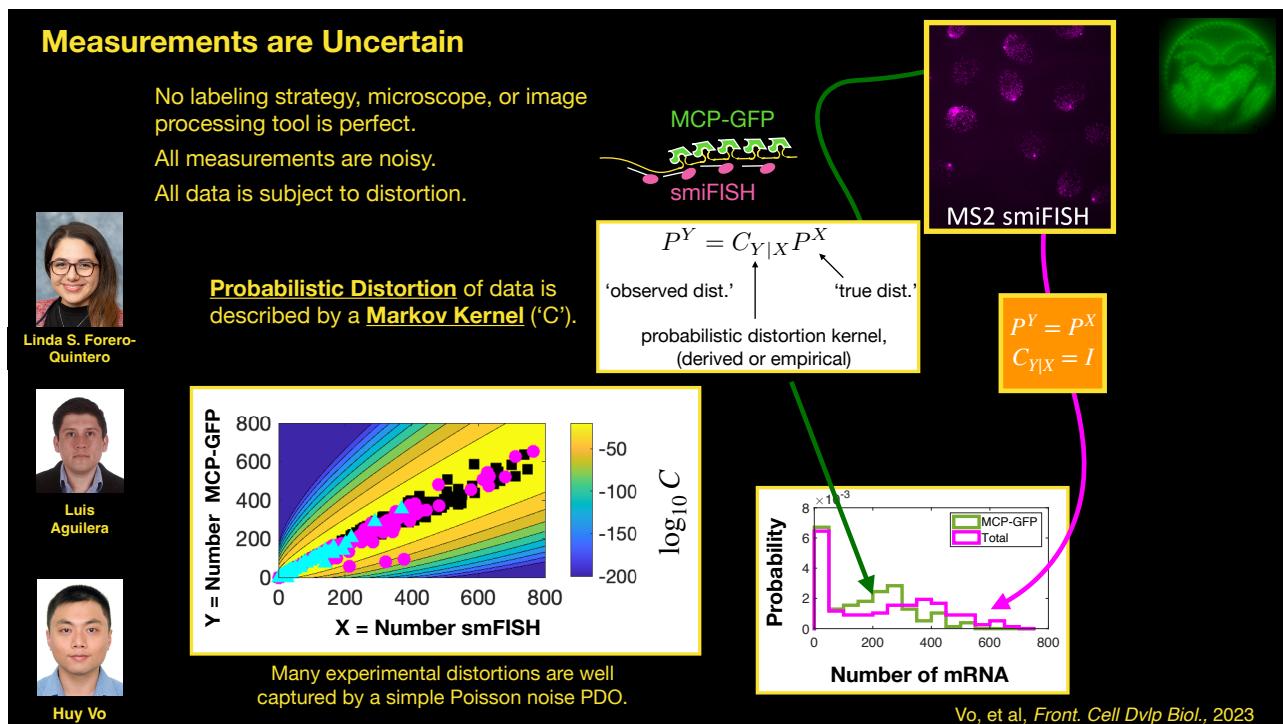
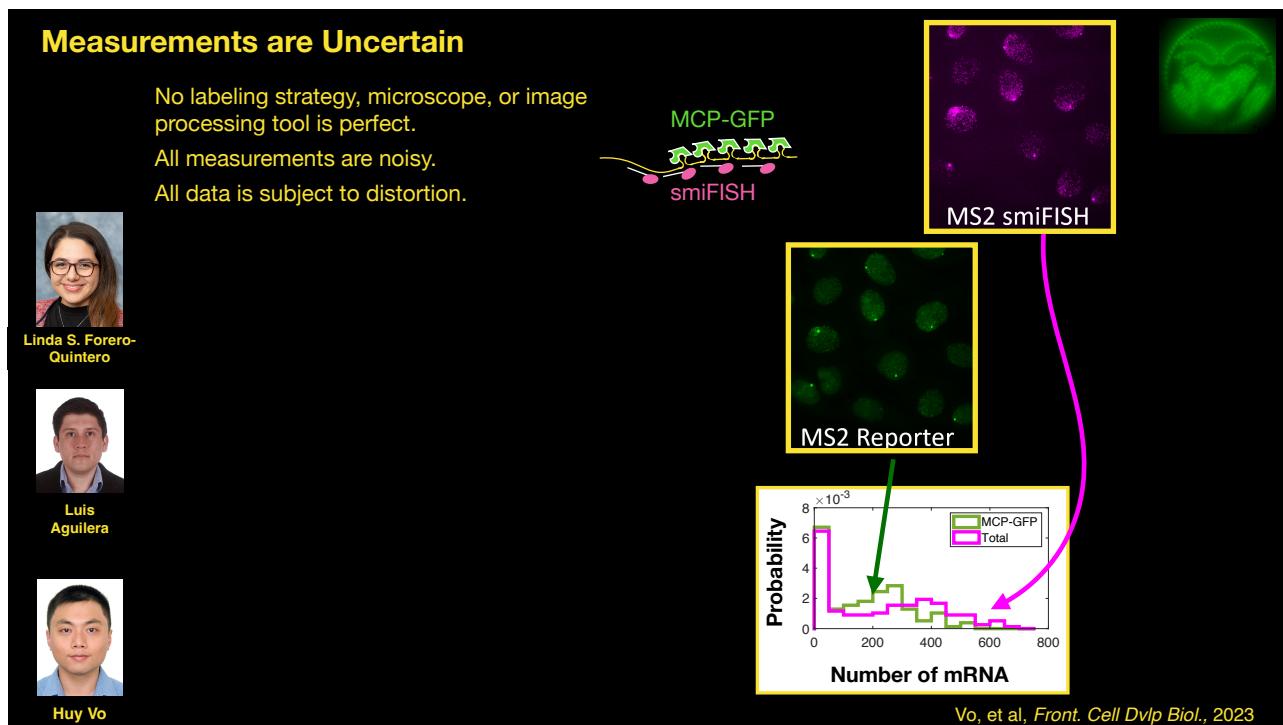
All measurements are noisy.

All data is subject to distortion.



Linda S. Forero-Quintero







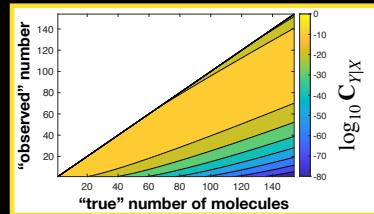
Example – Distortion Effects for Bursting DUSP1 mRNA Expression



Spots are often lost (e.g., due to inefficient labeling, out of focus imaging, spatial overlap).

Here we model this loss using a binomial distortion with a 70% detection rate.

PDO



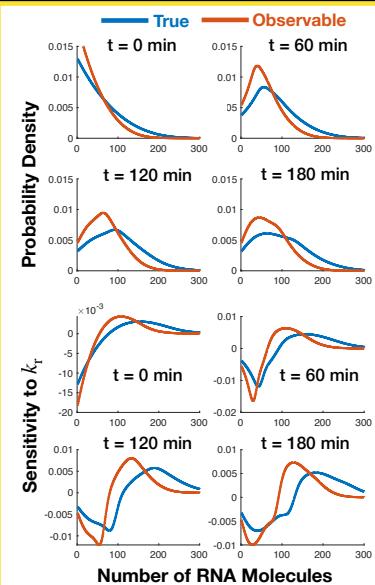
MATLAB CODE

```
%% Define a Binomial PDO
Model2 = Model1; % Make a copy of the original model
Model2.pdoOptions.type = 'Binomial';
Model2.pdoOptions.props.CaptureProbabilityS1 = 0; % Distortion for OFF species (unobserved)
Model2.pdoOptions.props.CaptureProbabilityS2 = 0; % Distortion for ON species (unobserved)
Model2.pdoOptions.props.CaptureProbabilityS3 = 0.7;% Distortion for RNA species
Model2.pdoOptions.PDO = Model2.generatePDO(Model2.pdoOptions,[],Mod1SensSoln.sens.data,true);

%% Plot the PDO
figure(20); contourf(log10(Model2.pdoOptions.PDO.conditionalPmf{3}),30); colorbar
xlabel('true number of mRNA'); ylabel('observed number of mRNA'); set(gcf,'fontsize',15);

%% Calculate and plot FSP and Sensitivity
Model2.solutionsScheme = 'FSP'; % Set solution scheme to FSP.
Model2.makePlot(Mod1FSPsoln,'marginals',[1:100:301],true,[1,2,3],{'linewidth',2}); % Plot Marginals
Model2.solutionsScheme = 'fSPsens'; % Set solution scheme to Sensitivity
Model2.makePlot(Mod1SensSoln,'marginals',[1:100:301],true,[3+(1:12)],{'linewidth',2}) % Plot Sensitivities
```

<https://github.com/MunskyGroup/SSIT>



Computing and Maximizing the Likelihood of Single-Cell Data



- Suppose that we measure N_c cells in a given experiment and quantify their expression as: $\mathbf{D} = \{d_1, d_2, \dots, d_{N_c}\}$
- The likelihood (and log-likelihood) to observe this dataset given our model can be calculated using the FSP solution:

$$L(\mathbf{D}) = \prod_{c=1}^{N_c} P_{d_c}(\theta) \quad \text{or} \quad \log L(\mathbf{D}) = \sum_{d_c \in \mathbf{D}} \log P_{d_c}(\theta)$$

- We can calculate the sensitivity of the log-probability of state \mathbf{x} :

$$\frac{d}{d\theta_i} \log P_{\mathbf{x}} = \frac{1}{P_{\mathbf{x}}} \frac{dP_{\mathbf{x}}}{d\theta_i} = \frac{\mathbf{s}_{i\mathbf{x}}}{P_{\mathbf{x}}}$$

which gives us the gradient of the log-likelihood of all data:

$$\frac{d}{d\theta_i} \log L(\mathbf{D}) = \sum_{d_c \in \mathbf{D}} \frac{\mathbf{s}_{id_c}}{P_{d_c}}$$

- With the likelihood function and its derivative, we can now search parameter space to maximize the likelihood or sample the posterior parameter distribution.

<https://github.com/MunskyGroup/SSIT>



Computing and Maximizing the Likelihood of Distorted Single-Cell Data



- Suppose that we measure N_c cells in a given experiment and quantify their expression as: $\mathbf{D} = \{d_1, d_2, \dots, d_{N_c}\}$
- The likelihood (and log-likelihood) to observe this dataset given our model (and image distortion effects) can be calculated using the FSP solution:

$$L(\mathbf{D}) = \prod_{d_c \in \mathbf{D}} [\mathbf{C}(\theta)\mathbf{P}(\theta)]_{d_c} \quad \text{or} \quad \log L(\mathbf{D}) = \sum_{d_c \in \mathbf{D}} \log[\mathbf{C}(\theta)\mathbf{P}(\theta)]_{d_c}$$

- We can also calculate the sensitivity of the log-probability of state \mathbf{x} :

$$\frac{d}{d\theta_i} \log[\mathbf{CP}]_{\mathbf{x}} = \frac{[\mathbf{Cs}_i]_{\mathbf{x}}}{[\mathbf{CP}]_{\mathbf{x}}} + \frac{[(d\mathbf{C}/d\theta_i)\mathbf{P}]_{\mathbf{x}}}{[\mathbf{CP}]_{\mathbf{x}}}$$

which gives us the gradient of the log-likelihood of all data:

$$\frac{d}{d\theta_i} \log L(\mathbf{D}) = \sum_{d_c \in \mathbf{D}} \left(\frac{[\mathbf{Cs}_i]_{d_c}}{[\mathbf{CP}]_{d_c}} + \frac{[d\mathbf{C}/d\theta_i]\mathbf{P}]_{d_c}}{[\mathbf{CP}]_{d_c}} \right)$$

- With the likelihood function and its derivative, we can now search parameter space to maximize the likelihood or sample the posterior parameter distribution.

<https://github.com/MunskyGroup/SSIT>



Example — Generating simulated data for Bursting DUSP1 mRNA Expression



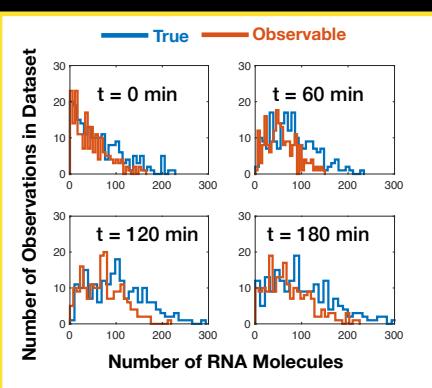
We simulate 50 sets of data, each with 200 independent cells at 4 time points.

MATLAB CODE

```
%% Simulate Data for Subsequent Fitting
Model2.tSpan = [0,60,120,180]; % Set the times at which to generate data.
Model2.solutionScheme = 'FSP'; % Set solution scheme to FSP.
Mod2FSPsoln = Model2.solve; % Solve Model
Model2.ssaOptions.Nexp = 50; % Number of independent data sets to generate.
Model2.ssaOptions.NSimsPerExpt = 200; % Number of cells to include at each time point for each data set.
Model2.ssaOptions.applyPDD = true; % Include the distortion in the data.
dataTable = Model2.sampleDataFromFSP(Mod2FSPsoln, 'DUSP1SSAdata50Expts.csv'); % Generate and save data.

% Plot data as histograms
tPlot = [0,60,120,180];
for i = 1:4
    subplot(2,2,i) % Switch to current subplot.
    histogram(dataTable.expl_s3(dataTable.time==tPlot(i)),30,"DisplayStyle","stairs"); hold on;
    histogram(dataTable.expl_s3_distorted(dataTable.time==tPlot(i)),30,"DisplayStyle","stairs");
end
```

The SSIT automatically saves simulated results with and without the distortion.

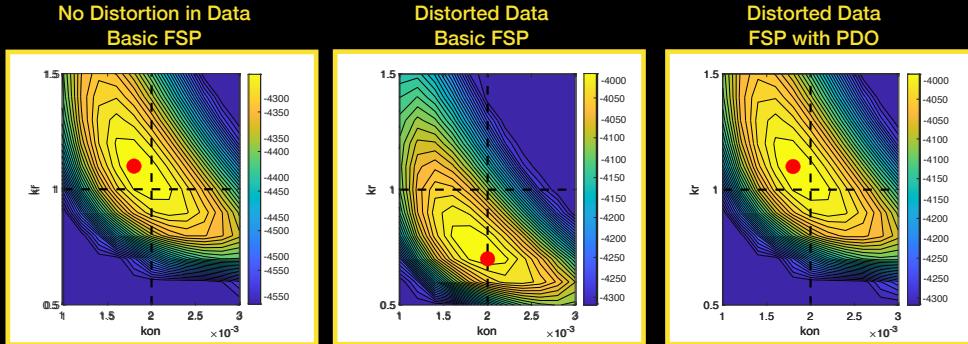


<https://github.com/MunskyGroup/SSIT>

Example — Likelihood Function for Bursting DUSP1 mRNA Expression



We sweep over model parameters to plot contours of the likelihood function with and without distortions.



Including the PDO in the parameter inference allows for excellent recovery of parameters.

MATLAB CODE

```
%> Create Model copies with and without distortion
Model2.solutionScheme = 'FSP'; % Set solution scheme back to FSP.
B{1} = Model2.loadData('DUSP1SSAData50Expts.csv',{ 'rna','exp2_a3' });
B{1}.pdoOptions.pdo = [ ]; % Do not use PDO.
B{2} = Model2.loadData('DUSP1SSAData50Expts.csv',{ 'rna','exp2_s3_Distorted' });
B{2}.pdoOptions.pdo = [1]; % Do not use PDO.
B{3} = Model2.loadData('DUSP1SSAData50Expts.csv',{ 'rna','exp2_a3_Distorted' });
```

```
%> Sweep over Parameters to Plot Likelihood Function Landscape
fitErrorsB1 = B{1}.likelihoodSweep([2,4],linspace(.5,1.5,11),true);
title('Ideal data. Original FSP.')
fitErrorsB2 = B{2}.likelihoodSweep([2,4],linspace(.5,1.5,11),true);
title('Binomial data distortion. Original FSP.')
fitErrorsB3 = B{3}.likelihoodSweep([2,4],linspace(.5,1.5,11),true);
title('Binomial data distortion. FSP+PDO.')
```

Calculating Fisher Information for Single-Cell Experiments



The **Fisher Information Matrix (FIM)** quantifies the information that an observation (e.g., a smFISH measurement) is expected to have about combinations of model parameters:

$$I_{ij}(\theta) = \mathbb{E}_{\mathbf{D} \in \mathcal{D}} \left\{ \frac{d \log L(\mathbf{D})}{d\theta_i} \frac{d \log L(\mathbf{D})}{d\theta_j} \right\}$$

Using the likelihood functions from before, we can write these in terms of the sensitivity functions:

$$I_{ij}(\theta) = \mathbb{E}_{\mathbf{x} \in \mathcal{D}} \left\{ \frac{\mathbf{s}_{ix} \mathbf{s}_{jx}}{P_x P_x} \right\} = \sum_{\mathbf{x} \in \mathcal{D}} P_x \frac{\mathbf{s}_{ix}}{P_x} \frac{\mathbf{s}_{jx}}{P_x} = \sum_{\mathbf{x} \in \mathcal{D}} \frac{\mathbf{s}_{ix} \mathbf{s}_{jx}}{P_x}$$

Or, if there is distortion due to imaging/processing effects ($P_{\text{obs}} = \mathbf{CP}_{\text{true}}$), this becomes:

$$I_{ij}(\theta) = \sum_{\mathbf{x} \in \mathcal{D}_{\text{obs}}} \frac{[\mathbf{Cs}_i + (d\mathbf{C}/d\theta_i)\mathbf{P}]_x [\mathbf{Cs}_j + (d\mathbf{C}/d\theta_j)\mathbf{P}]_x}{[\mathbf{CP}]_x}$$



Estimating Expected MLE Uncertainty using Fisher Information



The FIM provides an asymptotic (multivariate Gaussian) estimate for an unbiased Maximum Likelihood Estimator.

Asymptotic normality of the MLE:

$$\sqrt{n}(\hat{\theta} - \theta^*) \xrightarrow{\text{dist}} \mathcal{N}(0, I(\theta^*)^{-1})$$

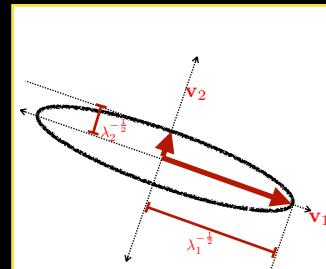
The FIM's eigenvalues $\{\lambda_i\}$ and its eigenvectors $\{v_i\}$ estimate the magnitudes and directions of uncertainty in MLE parameters (Cramer Rao Lower Bound).



Zach Fox



Huy Vo



Fox et al, PLoS Comp. Biol, 2019
Fox et al, Complexity, 2020

<https://github.com/MunskyGroup/SSIT>

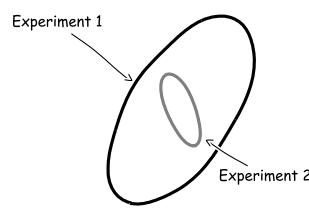


Using Fisher Information to Optimize the Design of Single-Cell Experiments



Different single-cell experiments reveal different amounts of information about different combinations of model parameters.

(Translation rate)



The FIM can estimate which experiments will provide tighter MLE results.

In this case, $|\mathcal{I}_2(\theta)| > |\mathcal{I}_1(\theta)|$



Zach Fox



Huy Vo

<https://github.com/MunskyGroup/SSIT>

Fox et al, PLoS Comp. Biol, 2019
Fox et al, Complexity, 2020

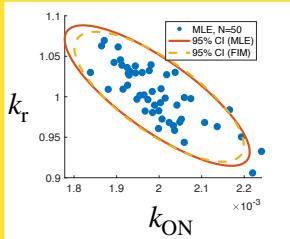


Example – FIM and MLE Variance for Bursting DUSP1 mRNA Expression

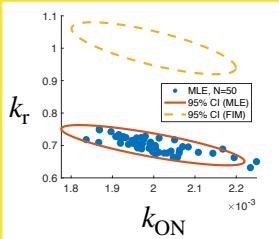


The FSP-FIM accurately predicts the MLE variance.

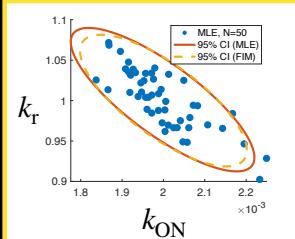
No Distortion in Data; Basic FSP



Binomial Distortion; Basic FSP



Binomial Distortion; FSP + PDO



When combined with the PDO, the FIM also correctly estimates the impact of data distortions.

MATLAB CODE

```
%> Find MLEs for Simulated Datasets (Takes several minutes/hours to complete)
for iExp = 1:Model12.seaOptions.NExp
for m = 1:3
switch m % Link appropriate data sets
case 1; b = B(1).loadData('DUSP1SADat200Expts.csv',{'rna',[ 'exp',num2str(iExp),'_s3' ]});
case 2; b = B(2).loadData('DUSP1SADat200Expts.csv',{'rna',[ 'exp',num2str(iExp),'_s3_Distorted' ]});
case 3; b = B(3).loadData('DUSP1SADat200Expts.csv',{'rna',[ 'exp',num2str(iExp),'_s3_Distorted' ]});
end
b.fittingOptions.modelVarsToFit = [2,3]; % Only fit two of the parameters
x0 = [b.parameters{b.fittingOptions.modelVarsToFit,2}]; % Initial parameter guess
[MLE(m,: ,iExp),fMLE(m,iExp)] = b.maximizeLikelihood(x0); % Run fitting code
end
end
```

MATLAB CODE

```
%> Compute FIM and Compare to MLE Spread
for m=1:3
fimResults(m) = B(m).computeFIM; % Compute FIM for individual times.
% Compute the FIM for full observations and no distortion.
FIM(m) = B(m).evaluateExperiment(fimResults(m),B(1).dataSet.nCells);
iVars = b.fittingOptions.modelVarsToFit; % Indices of free variables.
fimFreePars = FIM(m){1}(iVars,iVars); % Select FIM for parameters of interest
B(m).makeMLEFimPlot(squeeze(MLE_tmp(m,:,:)),fimFreePars) % Plot results
end
```

<https://github.com/MunskyGroup/SSIT>



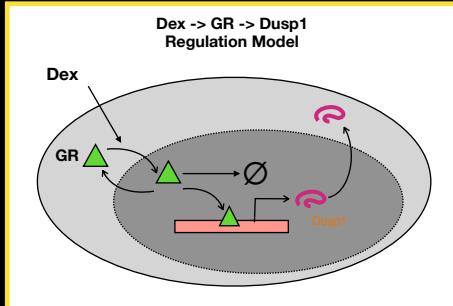
Example 2 – Parameter estimation and experiment design for real single-cell data.

<https://github.com/MunskyGroup/SSIT>

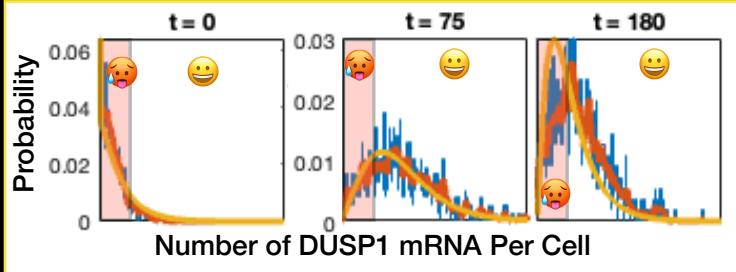
Quantifying Dexamethasone (Dex) stimulation of the Glucocorticoid Receptor (GR) and activation of DUSP1 transcription.



Model with GR transport and DUSP1 transcription dynamics.



By measuring the distribution of the cellular response over time, we quantify how each cell responds to Dex/GR.



Eric Ron



Luis Aguilera



Josh Cook



Low Dusp1
High Inflammation

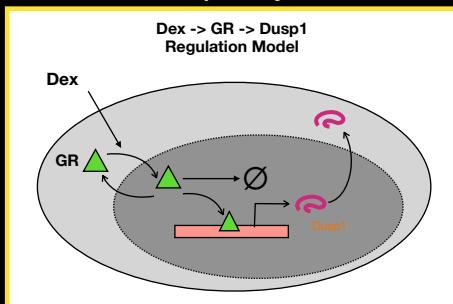


High Dusp1
Low Inflammation

Quantifying Dexamethasone (Dex) stimulation of the Glucocorticoid Receptor (GR) and activation of DUSP1 transcription.



Model with GR transport and DUSP1 transcription dynamics.



Load smFISH data into the previous model.

```
mReal = Model1; % Make copy of previous model
mReal = mReal.loadData('../ExampleData/Dusp1Data.csv',...
    {'rna', 'RNA_DUSP1_nuc'}, {'Dex_Conc', '100'}); % Load data into model
```

Specify Bayesian Prior and Fit Model to Data

```
% Specify Prior as log-normal distribution with wide uncertainty
mu_log10 = [-1,-2,0,-2,1,-1,-1]; % Prior log-mean
sig_log10 = 2*ones(1,7); % Prior log-standard deviation
mReal.fittingOptions.logPrior = @(x)-sum((log10(x)-mu_log10).^2./(2*sig_log10.^2));

mReal.fittingOptions.modelVarsToFit = [1:7]; % Choose parameters to search
DUSP1pars = [mReal.parameters(:,2)]; % Create first parameter guess
DUSP1pars = mReal.maximizeLikelihood(DUSP1pars); % Fit to maximize likelihood
mReal.parameters(:,2) = num2cell(DUSP1pars); % Update new parameters.
```

Plot Results

```
mReal.makeFitPlot. % Plot results.
```



Eric Ron



Luis Aguilera

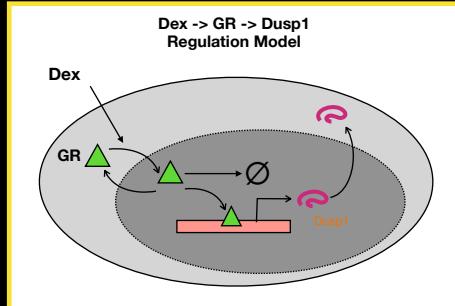


Josh Cook

Our simple model captures the full distributions of DUSP1 expression versus time.



Model with GR transport and DUSP1 transcription dynamics.



Eric Ron

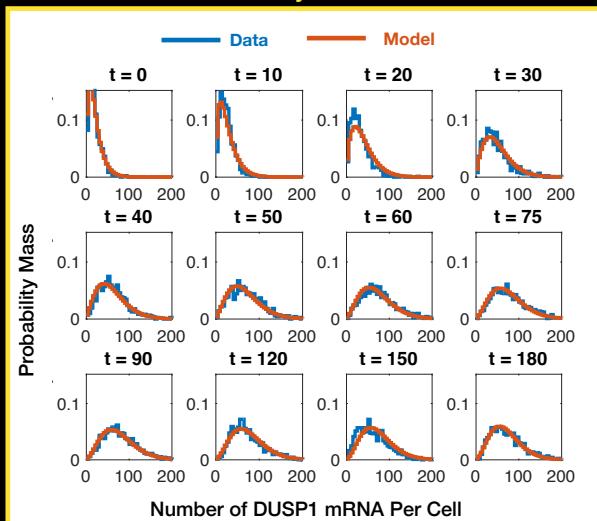


Luis Aguilera



Josh Cook

Fit of model to the DUSP1 activity.

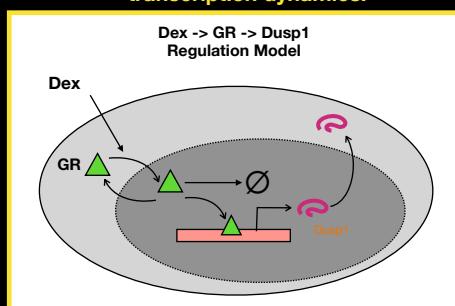


Ron, et al, In Preparation, 2024;
Cook, Ron, et al, Under Review, 2024

Our simple model captures the full distributions of DUSP1 expression versus time.



Model with GR transport and DUSP1 transcription dynamics.



Eric Ron

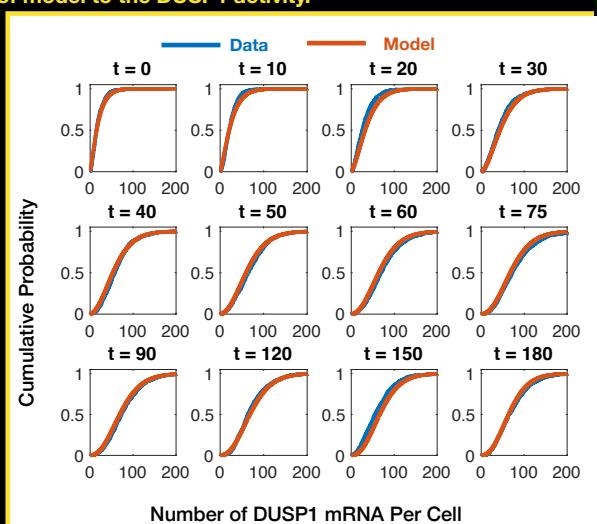


Luis Aguilera



Josh Cook

Fit of model to the DUSP1 activity.



Ron, et al, In Preparation, 2024;
Cook, Ron, et al, Under Review, 2024

Quantifying Dexamethasone (Dex) stimulation of the Glucocorticoid Receptor (GR) and activation of DUSP1 transcription.



Metropolis Hastings (MH) and FIM to Quantify Parameter Uncertainties

```
%> Compute FIM
fimResults = mReal.computeFIM([], 'log'); % Compute individual FIMs
fimTotal = mReal.evaluateExperiment(fimResults, mReal.dataSet.nCells, ...
    diag(sig_log10.^2)); % Compute total FIM including effect of prior.

%> Select FIM Components for DUSP1 Parameters only
mReal.fittingOptions.modelVarsToFit = [1:4]; % Parameters to search
FIMfree = fimTotal(1)([1:4],[1:4]); % Select free parameters.
COVfree = (1/2*(FIMfree+FIMfree'))^(-1); % Estimate Cov. using CRLB.

%> Define Metropolis Hastings Settings.
mReal.fittingOptions.logPrior = ...
    @(x)-sum((log10(x)-mu_log10([1:4])).^2./(2*sig_log10([1:4]).^2));
MHFitOptions = struct('proposalDistribution',@(x)mvrnd(x,COVfree),...
    'numberOfSamples',5000);
[DUSP1pars,~,MHResultsDusp1] = mReal.maximizeLikelihood(...
    [], MHFitOptions, 'MetropolisHastings'); % Run Metropolis Hastings
mReal.parameters([1:4],2) = num2cell(DUSP1pars);

%> Plot and compare FIM and Met. Hast. Results
mReal.plotMHResults(MHResultsDusp1,FIMfree,'log',[1]
```



Eric Ron

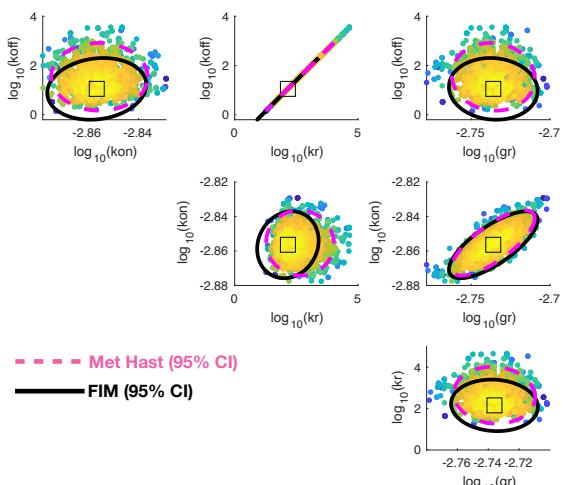


Luis Aguilera



Josh Cook

Comparison of Parameter Uncertainties using MCMC and FIM



Fisher Information can be used to choose more informative experiments



Use FIM to Find Optimal Design

```
nCellsOriginal = mReal.dataSet.nCells; % Original Expt Design
nCellsTotal = sum(nCellsOriginal); % Compute total number of cells

%> Optimize cell counts to minimize uncertainty in parameters 1-4.
nCellsOpt = mReal.optimizeCellCounts(fimResults,nCellsTotal,
    'Determinant',[, [], [], [], diag(sig_log10.^2));
```

Fit Data Only at Selected Time Points

```
mRealReduced = mReal; % Copy previous model
mRealReduced.fittingOptions.timesToFit = nCellsOpt>0; % Ignore timepoints.
parsRed = [mRealReduced.parameters(:,2)]; % Create first parameter guess
parsRed = mRealReduced.maximizeLikelihood(parsRed); % Maximize likelihood
mRealReduced.parameters(:,2) = num2cell(parsRed); % Update parameters.
```

Plot Fits and Predictions at All Time Points

```
mRealReduced.fittingOptions.timesToFit = ones(size(nCellsOpt),'logical');
mRealReduced.makeFitPlot % Plot fitting results
```



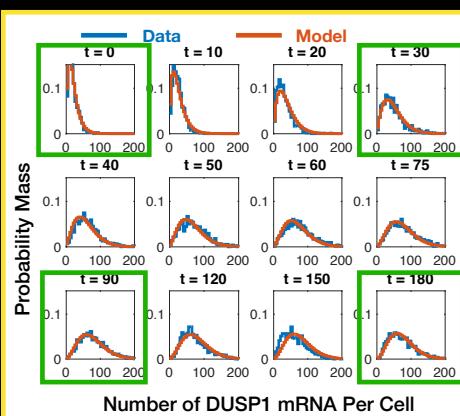
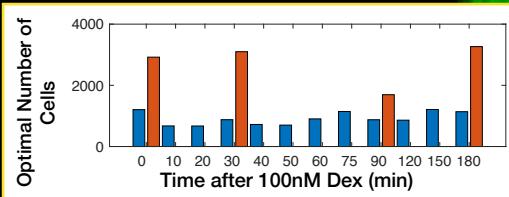
Eric Ron



Luis Aguilera



Josh Cook





Summary of SSIT Capabilities



- **Fast Model Creation** - add/remove species, reactions, time varying inputs
- **Flexible Solutions Schemes** for time-inhomogeneous discrete stochastic models with non-linear reaction propensities.
 - Stochastic Simulations
 - Finite State Projection
 - ODE and Moment Closure (only for polynomial propensity functions)
- **CME Sensitivity Analysis** (forward integration, finite difference)
- **Measurement Noise Correction** (probability distortion operator)
- **Extrinsic Noise Analysis** (parameter sampling, model extensions)
- **Data loading, fitting and plotting**
 - Maximum Likelihood Estimation (local and global optimizers)
 - Bayesian parameter estimation (Metropolis Hastings)
 - Joint model and PDO estimation
- **Fisher Information** to estimate expected MLE variance
- **FIM-Based Experiment design** (different inputs, different sample sizes, different sample times, different distortion operators, different observable species, different data binning strategies, etc...)

<https://github.com/MunskyGroup/SSIT>



Current Limitations and Ongoing Work



- **Model Size Limitations:**
 - FSP analyses are limited to a small number (e.g., <5) independent species (more if there are conservation relations or low-copy quantities).
 - Models with <1,000,000 high-probability states can be solved on laptop. HPC solutions are needed for larger models.
 - Models with more states can almost always be projected onto *much* lower dimension subspaces (e.g., lumped states, interpolation meshes, or slow eigen-modes), but finding the best subspace definitions for these projections is often not trivial.
- **Parameter Search Limitations:**
 - Parameter searches for larger models (e.g., >10 parameters) require substantial computational effort and experimental data:
 - The FIM can help to estimate data requirements and feasibility for different models with varying numbers of unknown parameters.
- **Measurement Distortion Limitations:**
 - Measurement distortions are typically unknown and need to be estimated empirically or fit simultaneously with the model:
 - FIM analysis can determine what distortions lead to non-identifiability or which are the most important to eliminate for a given model.
- **FIM Requirement for Prior Knowledge:**
 - To compute the FIM for experiment design, one must specify model and parameters.
 - Fortunately, FIM analyses can be sampled over a prior on expected mechanisms and parameters to find designs that work for all/most models.

<https://github.com/MunskyGroup/SSIT>

Thank you to all my collaborators and teammates!
current team

collaborators



Gregor Neuert

Hossein Jashnsaz



Timothy
Stasevich

Tatsuya
Morisaki

Ken Lyon Amanda Koch



Eric Ron

Luis
Aguilera



Will
Raymond

Alex
Popinga

Dmitri
Svetlov

alumni



Linda S. Forero-
Quintero



Huy Vo



Joshua
Cook



Zach Fox



funding



NIGMS



CAREER
Award



MONFORT
FAMILY FOUNDATION

**join these teams to do great
experiments.**

munsky@colostate.edu