

Using Stochastic Models to Design Maximally informative Single-Cell Experiments.

Brian Munsky

Chemical and Biological
Engineering
Colorado State University

ALL ANALYSES AND PLOTS IN THIS TUTORIAL
CAN BE REPRODUCED IN MATLAB:

BEFORE WE GET STARTED, PLEASE:

(1) Clone GitHub Repository:

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

(2) Download and add Tensor Toolbox to Matlab

path: <https://www.tensortoolbox.org/>

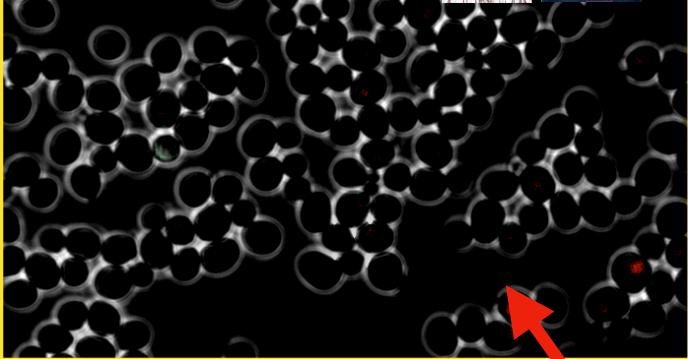
(3) Open script “**CommandLine/**

example_BPPB.m” in Matlab Editor



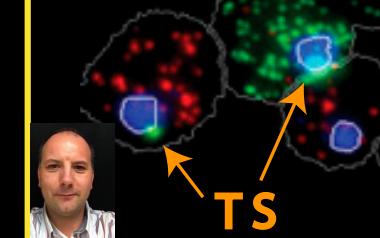
Measuring and Modeling the Central Dogma of Molecular Biology

Neuert, et al, *Science* 2013
Munsky, et al, *PNAS*, 2018
Jashnsaz, et al, *iScience*, 2020

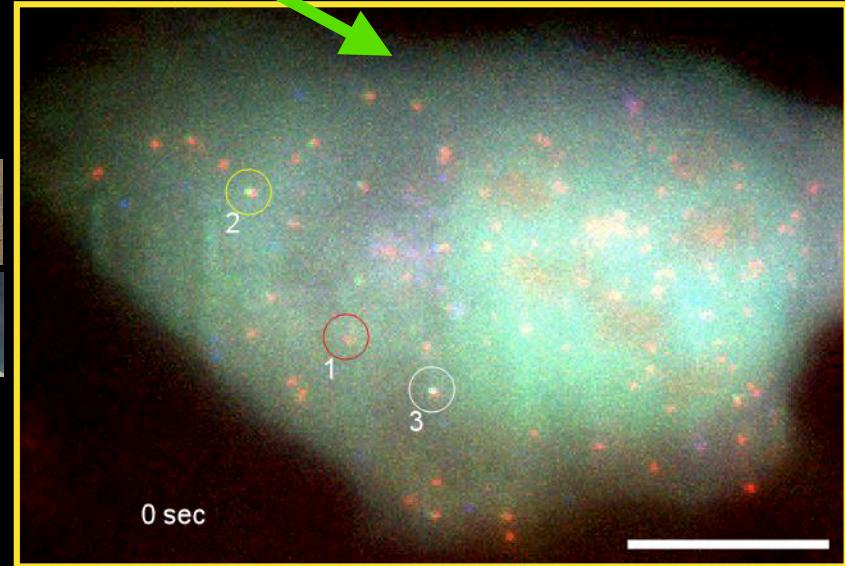
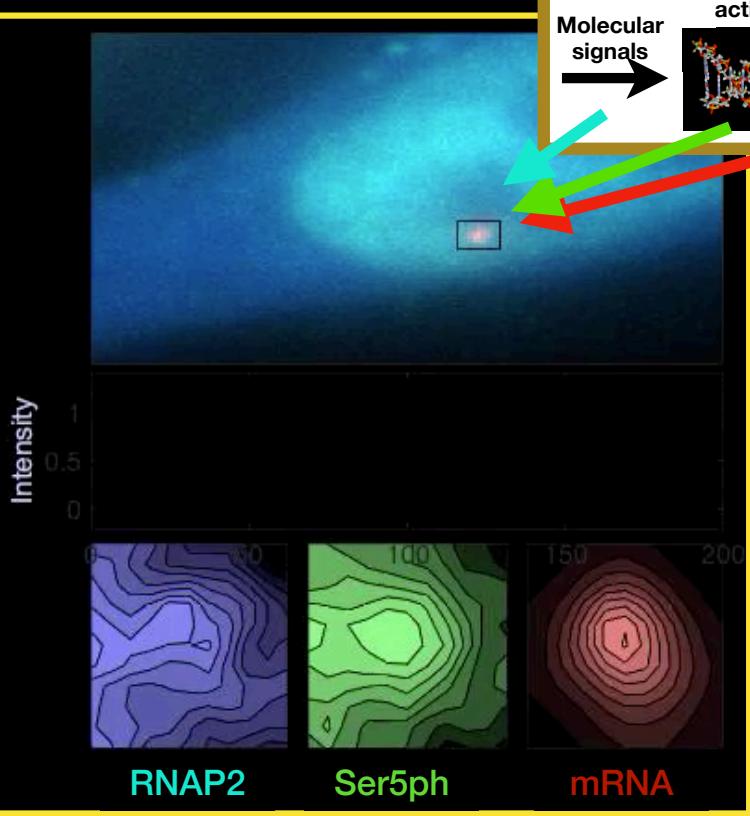
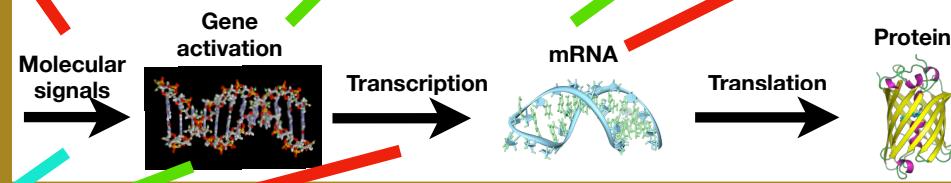
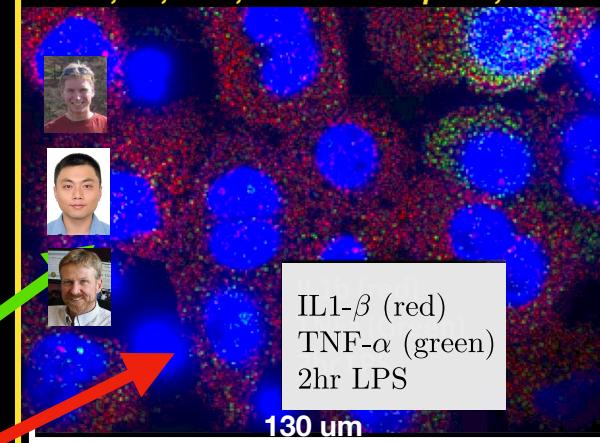


Neuert, et al, *Science* 2013
Munsky, et al, *PNAS*, 2018

10 min



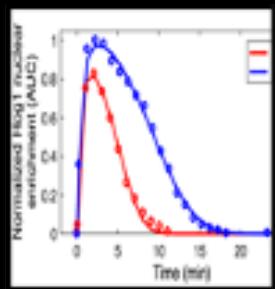
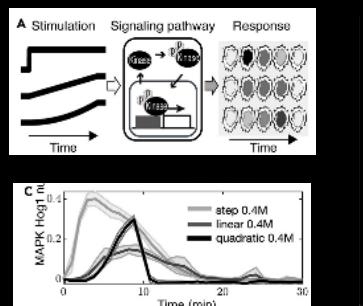
Kalb, Vo, et al, *Scientific Reports*, 2021



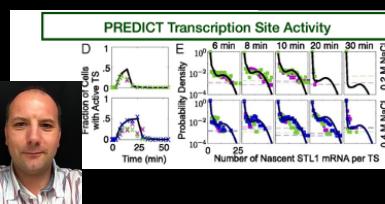
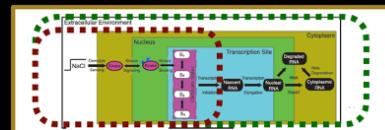
Lyon, Aguilera, et al, *Molecular Cell*, 2019
Aguilera, Raymond, et al, *PLoS Comp Biol*, 2019
Koch, Aguilera et al, *Nat. Struct, Mol. Biol.*, 2020

Measuring and Modeling the Central Dogma of Molecular Biology

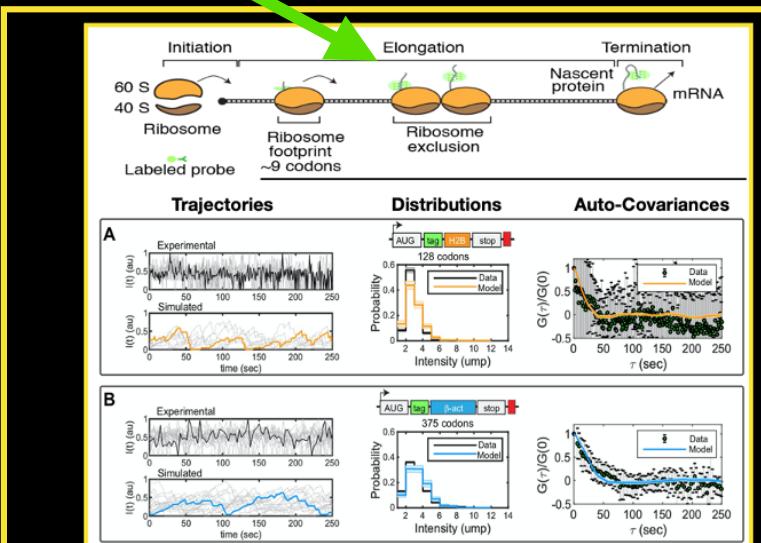
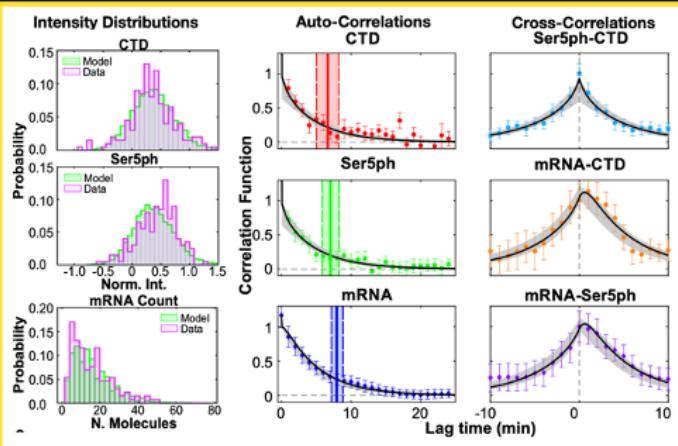
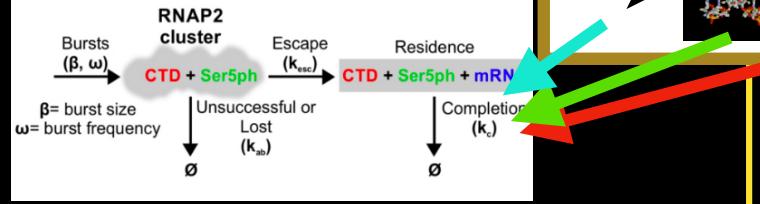
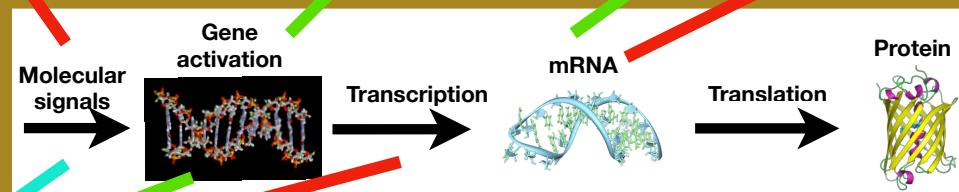
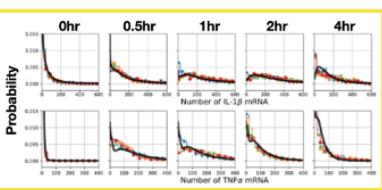
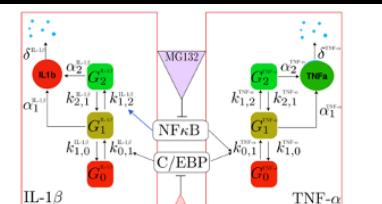
Neuert, et al, *Science* 2013
 Munsky, et al, *PNAS*, 2018
 Jashnsaz, et al, *iScience*, 2020



Neuert, et al, *Science* 2013
 Munsky, et al, *PNAS*, 2018

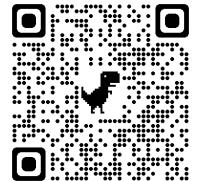


Kalb, Vo, et al, *Scientific Reports*, 2021



Forero, Raymond et al, *Nat. Comms.*, 2021

Lyon, Aguilera, et al, *Molecular Cell*, 2019
 Aguilera, Raymond, et al, *PLoS Comp Biol*, 2019
 Koch, Aguilera et al, *Nat. Struct, Mol. Biol.*, 2020



Observations From Measuring and Modeling Single-cell Processes



- * Transcription and Translation occur in bursts that can be measured in *real time*, at *single-molecule resolution*, and in *living cells*.
- * *Simple* discrete stochastic models are sufficient to quantitatively reproduce and often predict *every step* of these processes.
- * By *testing multiple models* in different stress or drug response conditions, it is possible to gain insight into which bursting mechanisms are affected under what experimental perturbations.



That's all 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

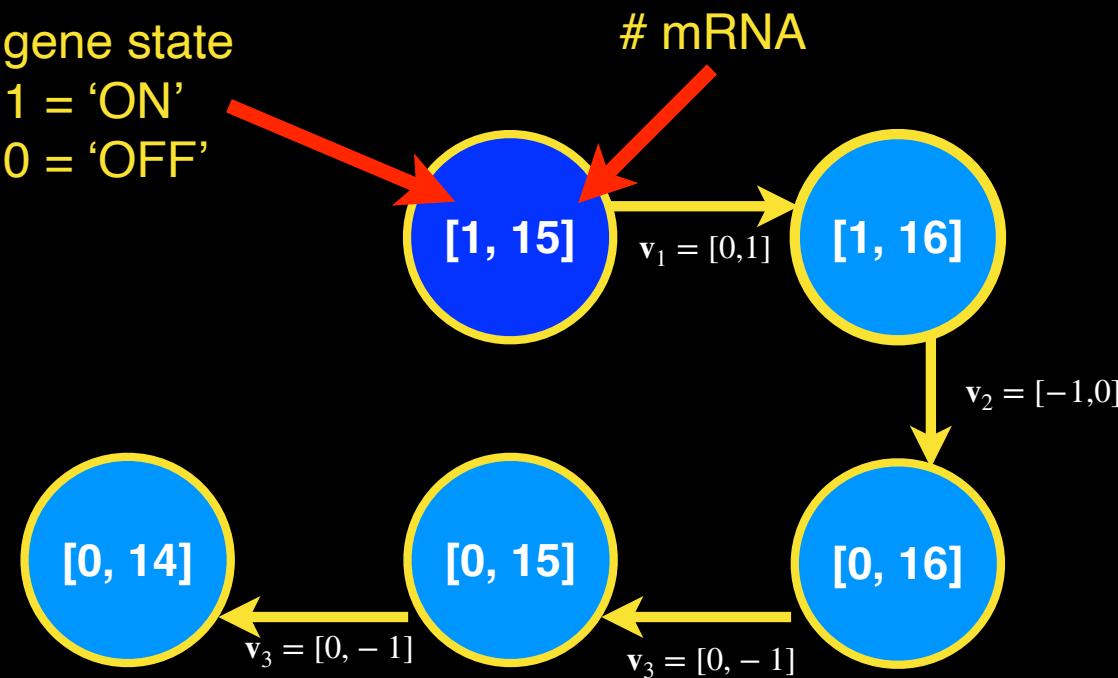
- * To squeeze **reliable information** out of an experiment, models and computation must account for **experimental uncertainties**.
- * We also need systematic tools to **choose** experiments can **minimize uncertainty** about mechanisms or parameters of interest.



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 $\{\mathbf{v}_\mu\}$.

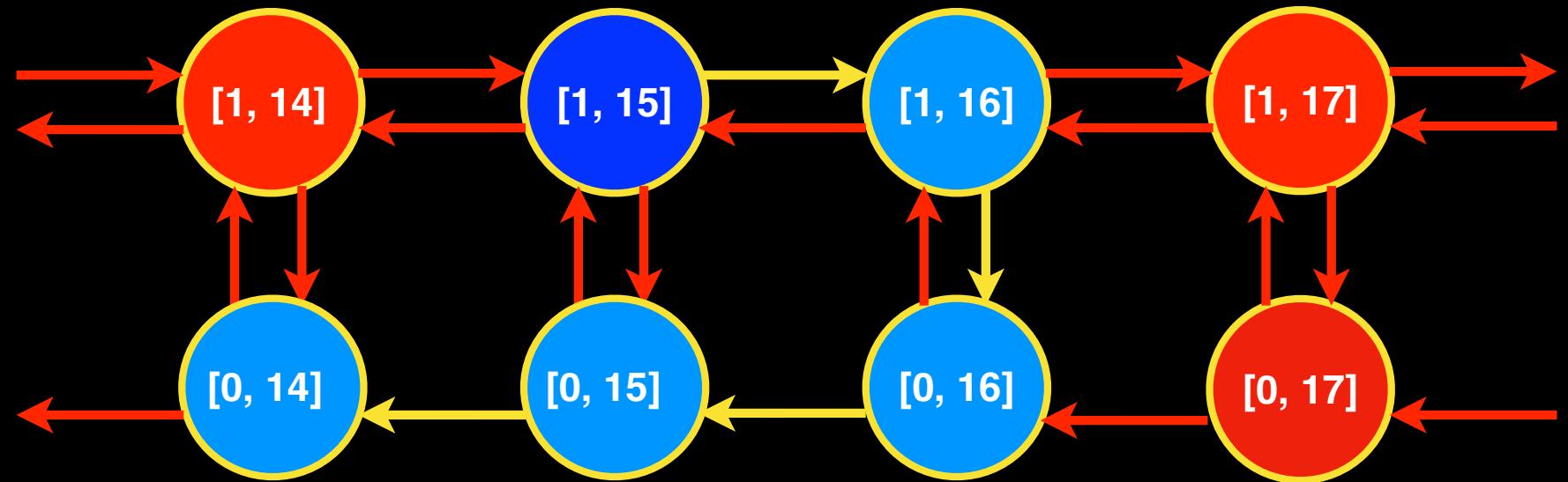




The Markov Model description of gene expression



- 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:

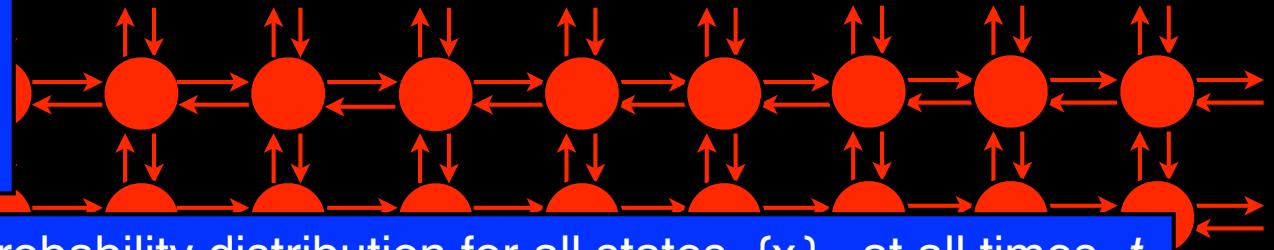




The Markov Model description of gene expression



Or others...



We wish to compute the probability distribution for all states, $\{x_i\}$, at all times, t .

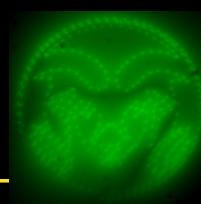
The probability mass for each state is described by the Linear ODE known as the Chemical Master Equation:

$$\frac{dP(\mathbf{x}_i)}{dt} = \sum_{\mu=1}^M -w_\mu(\mathbf{x}_i)P(\mathbf{x}_i) + w_\mu(\mathbf{x}_i - \mathbf{v}_\mu)P(\mathbf{x}_i - \mathbf{v}_\mu), \text{ for } i = 1, 2, \dots$$

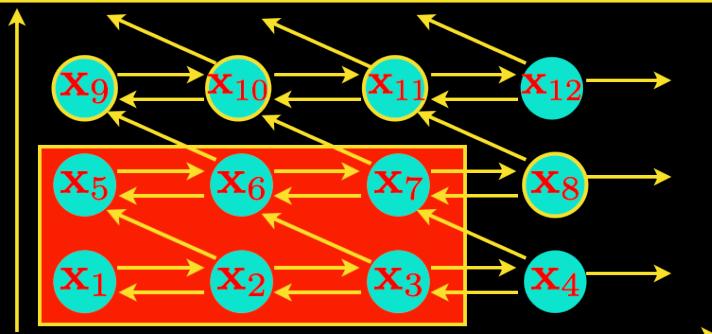
Or, in Matrix form:

$$\frac{d}{dt}\mathbf{P} = \mathbf{AP} \quad \text{where} \quad A_{ij} = \begin{cases} -\sum_{\mu=1}^M w_\mu(\mathbf{x}_j)P(\mathbf{x}_j) & \text{for } i = j \\ w_\mu(\mathbf{x}_j)P(\mathbf{x}_j) & \text{for } \mathbf{x}_i = \mathbf{x}_j + \mathbf{s}_\mu \\ 0 & \text{otherwise} \end{cases}$$

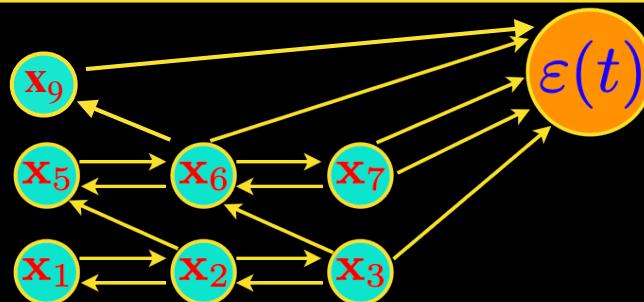
The Finite State Projection (FSP) approach to solve the CME



The Full System



The Projected System (FSP)



Full Master Equation

$$\begin{bmatrix} \dot{\mathbf{P}}_J \\ \dot{\mathbf{P}}_{J'} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_J & \mathbf{A}_{JJ'} \\ \mathbf{A}_{J'J} & \mathbf{A}_{J'} \end{bmatrix} \begin{bmatrix} \mathbf{P}_J(t) \\ \mathbf{P}_{J'}(t) \end{bmatrix}$$

Dimension = #(J) + #(J') = Infinite

FSP Master Equation

$$\begin{bmatrix} \dot{\mathbf{P}}_J^{FSP} \\ \dot{\varepsilon} \end{bmatrix} = \begin{bmatrix} -\mathbf{A}_J & \mathbf{0} \\ -1^T \mathbf{A}_J & 0 \end{bmatrix} \begin{bmatrix} \mathbf{P}_J^{FSP}(t) \\ \varepsilon(t) \end{bmatrix}$$

Dimension = #(J) + 1 = 8

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

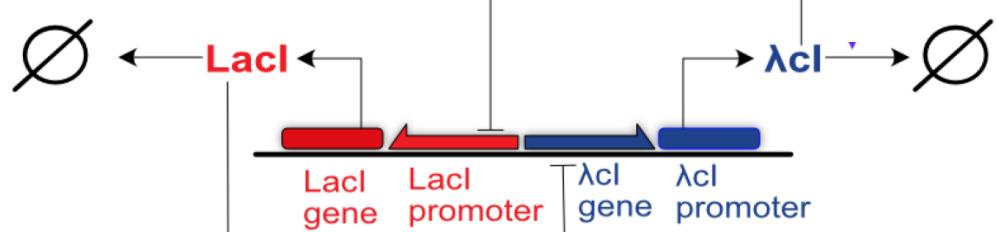
2) Exact error of solution: $\left\| \begin{bmatrix} \mathbf{P}_J(t) \\ \mathbf{P}_{J'} \end{bmatrix} - \begin{bmatrix} \mathbf{P}_J^{FSP}(t) \\ \mathbf{0} \end{bmatrix} \right\|_1 = \varepsilon_J(t)$

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

4) Convergence of Likelihood Function:

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

Example – FSP Analysis of Genetic Toggle Switch



$$w_1 = k_b + ka \frac{M^3}{M^3 + [\text{lacI}]^3}, \quad s_1 = [1, 0]^T$$

$$w_2 = g[\lambda cl], \quad s_2 = [-1, 0]^T$$

$$w_3 = k_b + ka \frac{M^3}{M^3 + [\lambda cl]^3}, \quad s_3 = [0, 1]^T$$

$$w_4 = g[\text{LacI}], \quad s_4 = [0, -1]^T$$

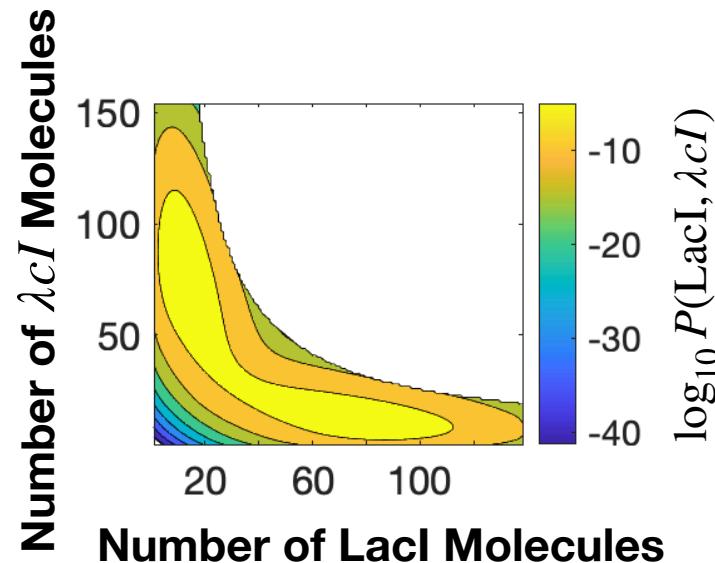
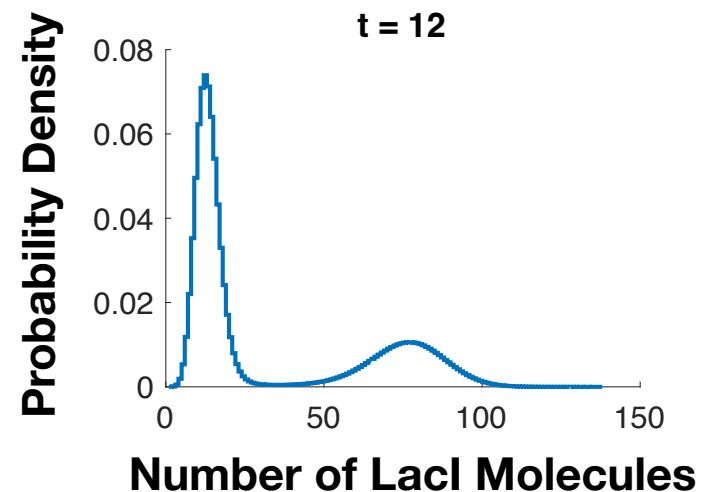
MATLAB CODE

```

%% Create SSIT Model
A = SSIT(); % Create SSIT instance using pre-selected model
A.species = {'x1'; 'x2'}; % Set species names.
A.parameters = {'kb', 10; 'ka', 80; 'M', 20; 'g', 1}; % Set parameter names and values
A.stoichiometry = [1, -1, 0, 0, 0, 1, -1]; % Set Stoichiometry matrix
A.propensityFunctions = {'kb+ka*M^3/(M^3+x2^3)'; 'g*x1'...
    'kb+ka*M^3/(M^3+x1^3)'; 'g*x2'}; % Set propensity functions
A.initialCondition = [0; 0]; % Set initial condition
A.customConstraintFuns = {'(x1-3).^2.*(x2-3).^2'}; % Set FSP constraint.
A.tSpan = [0:3:12]; % Set times at which to compute distributions

%% Solve using the FSP approach
A.solutionScheme = 'FSP'; % Set solutions scheme to FSP.
A.fspOptions.fspTol = 1e-4; % Set FSP error tolerance.
[FSPsoln,A.fspOptions.bounds] = A.solve; % Solve the FSP analysis
A.makePlot(FSPsoln, 'marginals', [2:5], false, [1, 2]) % Plot marginal distributions
A.makePlot(FSPsoln, 'joints', [2:5], false, [5]) % Plot joint distributions

```





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 chain 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:

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

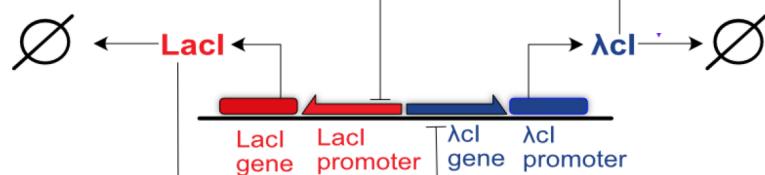


Example – FSP Sensitivity Analysis for Genetic Toggle Switch



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

Here we show only the sensitivity at the time $t=12\text{a.u}$ and for species LacI.

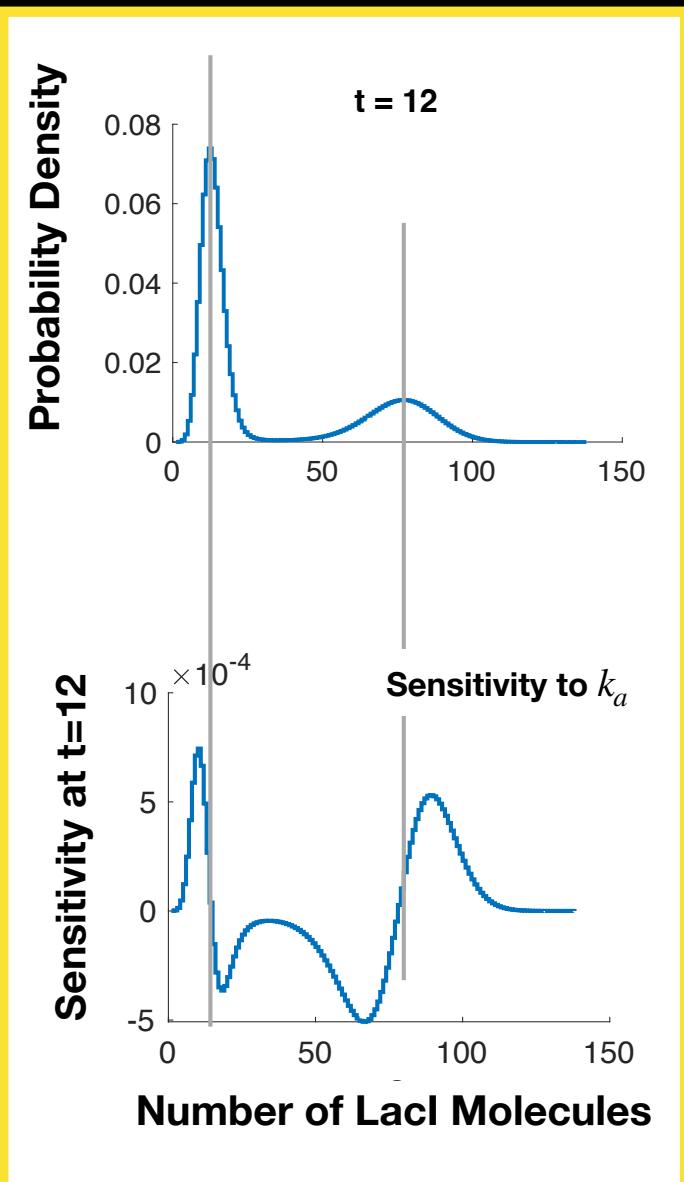


$$w_1 = k_b + ka \frac{M^3}{M^3 + [\text{lacI}]^3}, \quad w_2 = g[\lambda cI],$$

$$w_3 = k_b + ka \frac{M^3}{M^3 + [\lambda cI]^3}, \quad w_4 = g[\text{LacI}],$$

MATLAB CODE

```
%> Solve Sensitivity using FSP
A.solutionScheme = 'fspSens'; % Set solutions scheme to FSP Sensitivity
[sensSoln,bounds] = A.solve(FSPsoln.stateSpace); % Solve the sensitivity problem
A.makePlot(sensSoln,'marginals',[],false,[3,4]) % Plot marginal sensitivities
```





Extending FSP Models to Account for Image Distortion Effects

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

All measurements are noisy.

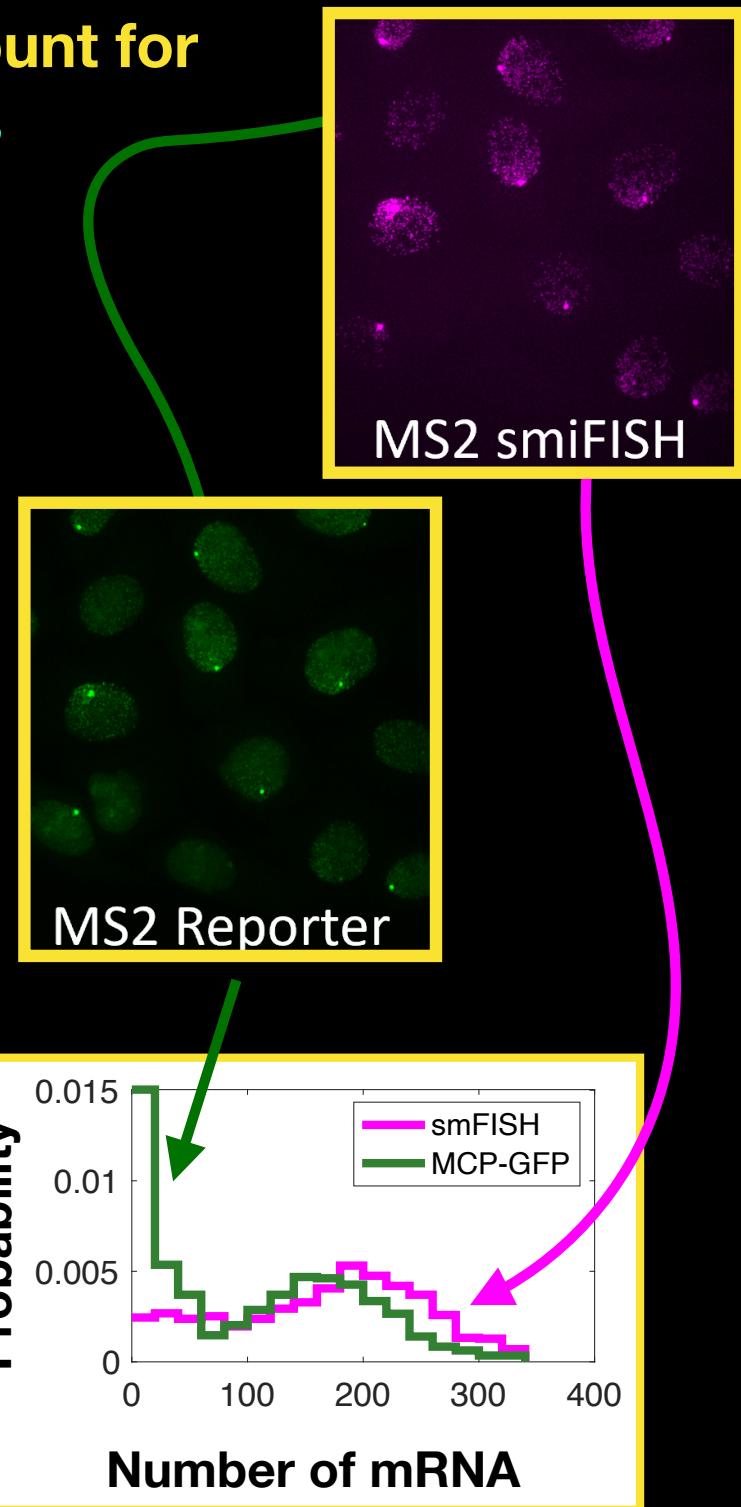
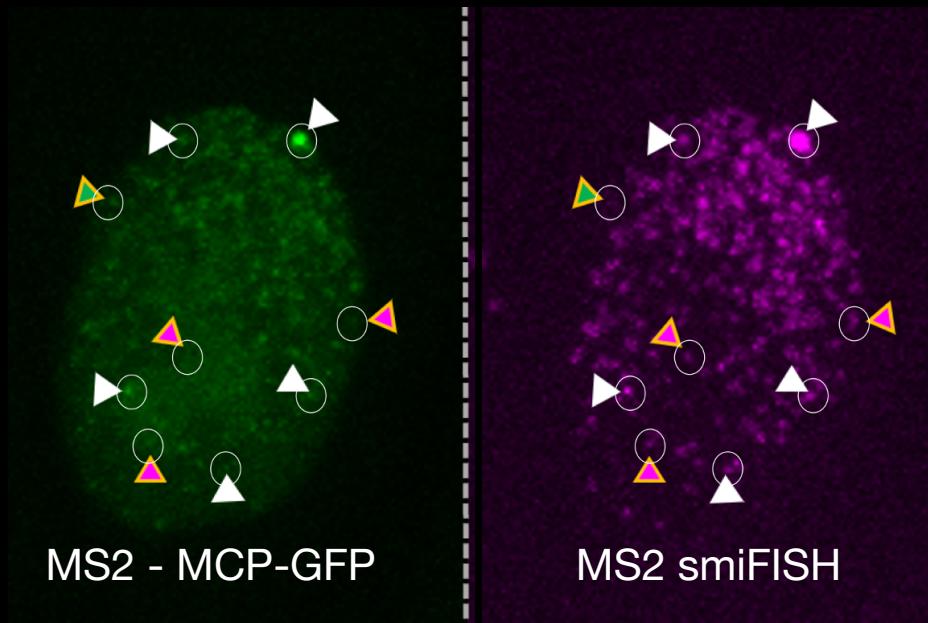
All data is subject to distortion.



Linda S. Forero-Quintero



Huy Vo





Extending FSP Models to Account for Image Distortion Effects

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

All measurements are noisy.

All data is subject to distortion.

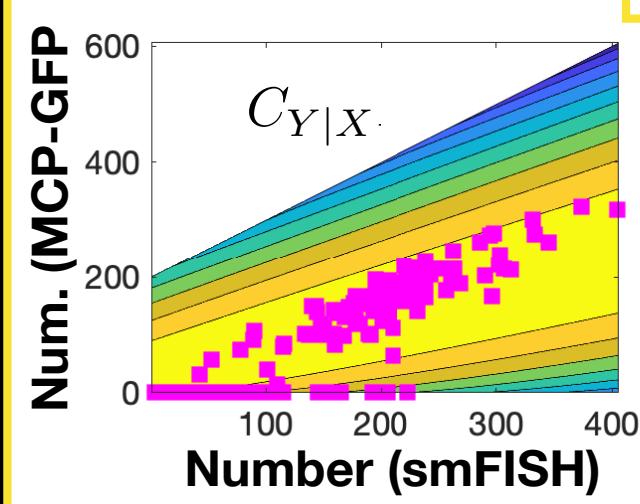
Probabilistic Distortion of data is described by a **Markov Kernel**.



Linda S. Forero-Quintero



Huy Vo

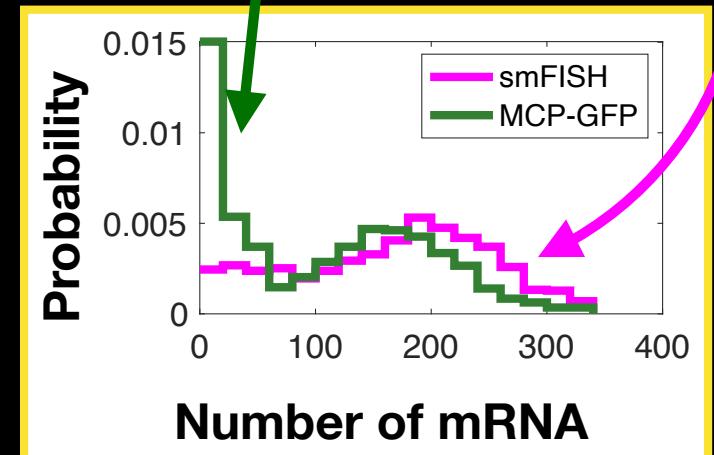


Experimental distortions match well to a combination of binomial spot loss and Poisson false spot generation.

$$P^Y = C_{Y|X} P^X$$

↑
‘observed dist.’ ‘true dist.’
probabilistic distortion kernel,
(derived or empirical)

$$P^Y = P^X$$
$$C_{Y|X} = I$$



MS2 smiFISH



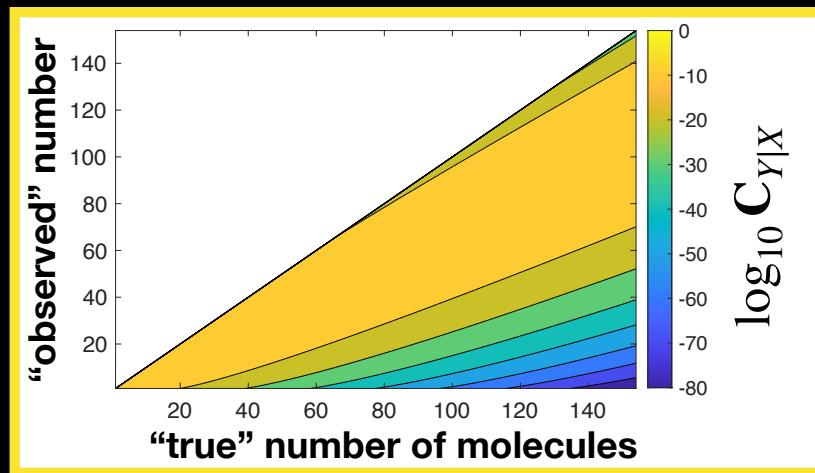
Example – Distortion Effects for Genetic Toggle Switch



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

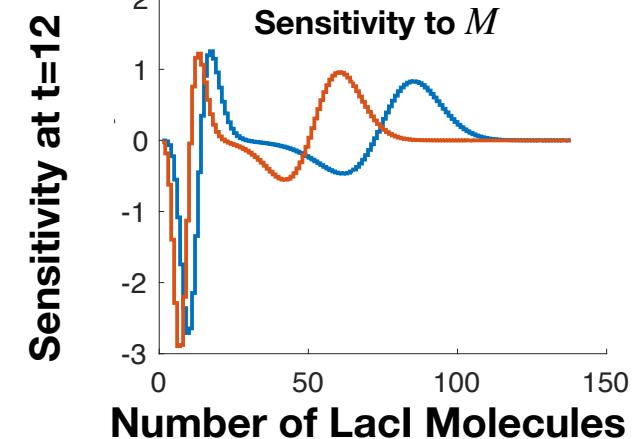
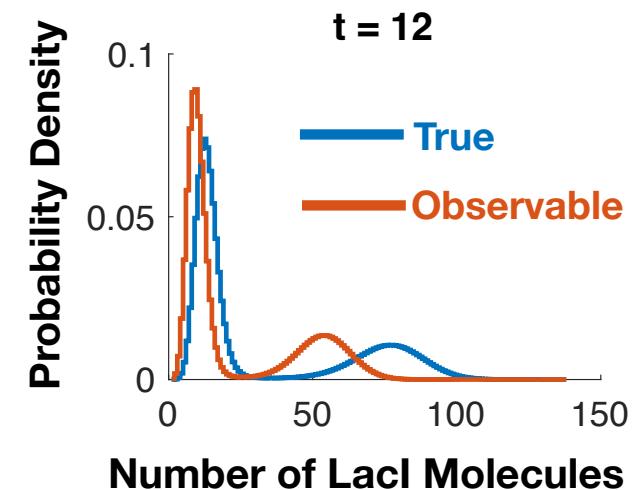
Here we model this loss using a binomial distortion with a 70% detection rate.

PDO



MATLAB CODE

```
%% Define a Binomial PDO
A.pdoOptions.type = 'Binomial';
A.pdoOptions.props.CaptureProbabilityS1 = 0.7; % Distortion for S1
A.pdoOptions.props.CaptureProbabilityS2 = 0.7; % Distortion for S2
A.pdoOptions.PDO = A.generatePDO(A.pdoOptions,[0.7,0.7],FSPsoln.fsp,true);
%% Plot the PDO
figure(20); contourf(log10(A.pdoOptions.PDO.conditionalPmfs{1})); colorbar
xlabel('true number of mRNA'); ylabel('observed number of mRNA');
%% Calculate and plot FSP and Sensitivity
A.solutionScheme = 'FSP'; % Set solution scheme to FSP.
A.makePlot(FSPsoln,'marginals',[2:5],true,[1,2]) % Plot Distorted Marginals
A.solutionScheme = 'fspSens'; % Set solution scheme to Sensitivity
A.makePlot(sensSoln,'marginals',[],true,[3,4]) % Plot Distorted Sensitivities
```





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.



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.



Example – Likelihood Function for Genetic Toggle Switch



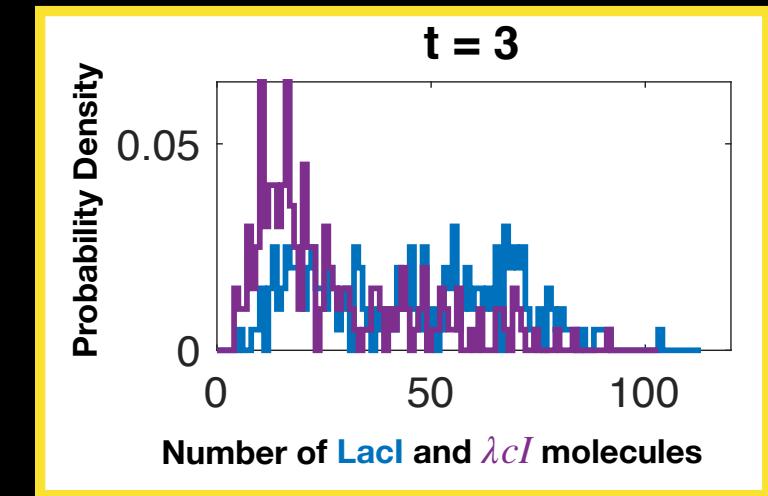
We simulate 150 sets of data, each with 200 independent cells at 5 time points.

MATLAB CODE

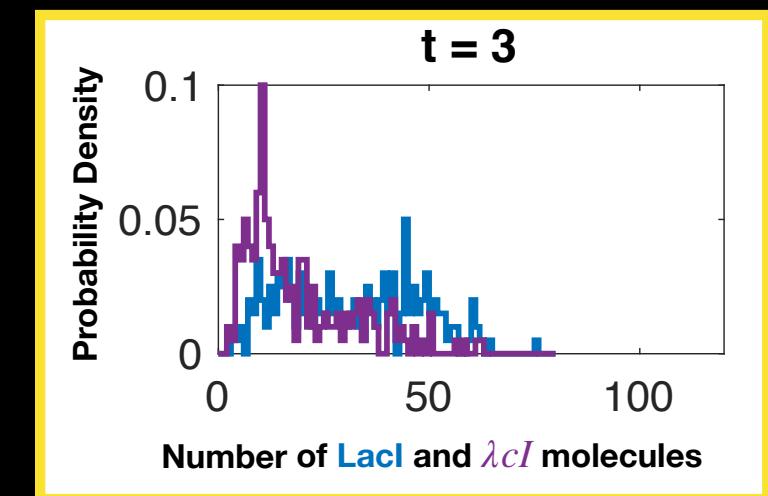
```
% Simulate Data for Subsequent Fitting
A.solutionScheme = 'SSA'; % Set solution scheme to SSA.
A.ssaOptions.Nexp = 150; A.ssaOptions.nSimsPerExpt = 200;
A.ssaOptions.applyPDO = true; % Include distortion in the SSA.
A.solve([], 'ToggleSSAData50Expts.csv');
```

The SSA automatically saves SSA results with and without the distortion.

No Distortion in Data



Binomial Distortion



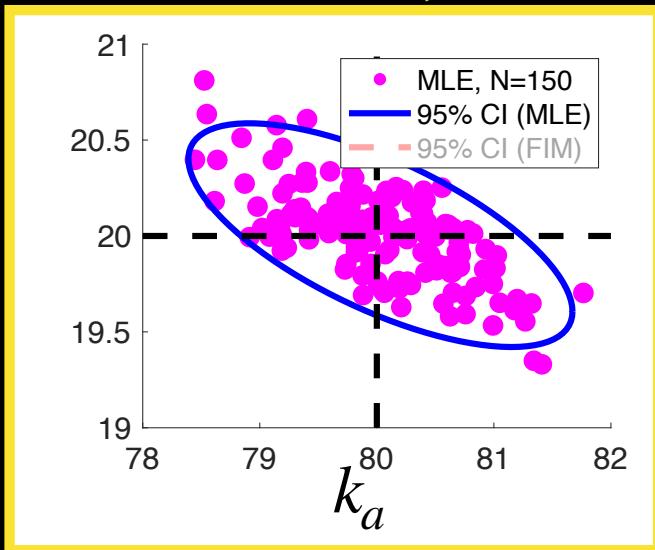


Example – FIM and MLE Variance for Genetic Toggle Switch

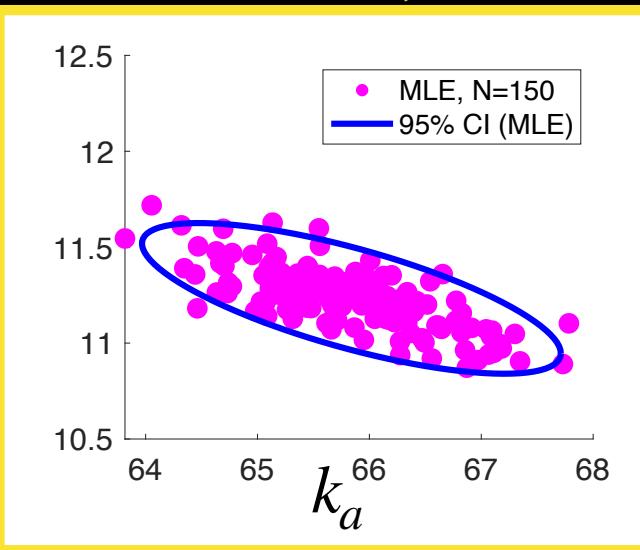


We then fit each dataset to find the maximum likelihood estimates.

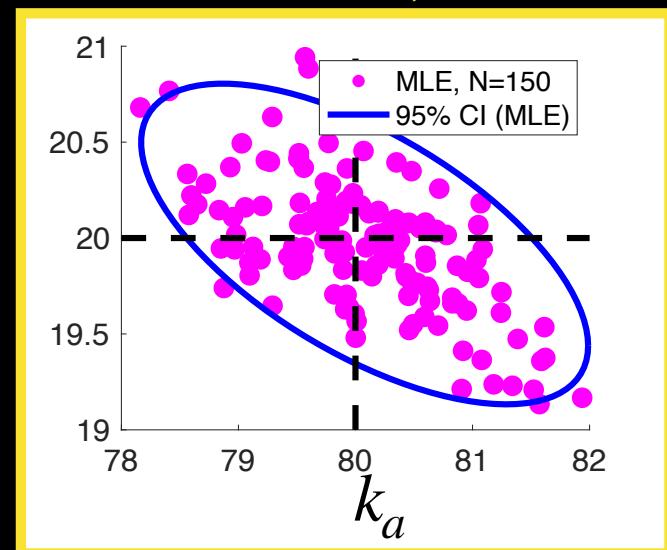
No Distortion in Data; Basic FSP



Binomial Distortion; Basic FSP



Binomial Distortion; FSP + PDO



MATLAB CODE

```
%% Find MLE for each simulated data set.
MLE = zeros(3,2,A.ssaOptions.Nexp);
fMLE = inf(3,2,A.ssaOptions.Nexp);
for iExp = 1:A.ssaOptions.Nexp
    B{3} = B{3}.loadData('ToggleSSAData50Expts.csv',{'x1',[ 'exp',num2str(iExp), '_s1_Distorted' ],...
                           'x2',[ 'exp',num2str(iExp), '_s2_Distorted' ]}); % Link data to model.
    B{3}.fittingOptions.modelVarsToFit = [2,3];
    x0 = [B{3}.parameters{B{3}.fittingOptions.modelVarsToFit,2}]';
    [MLE(3,:,iExp),fMLE(m,:,iExp)] = B{3}.maximizeLikelihood(x0);
end
```



Huy Vo



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:

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

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

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



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:

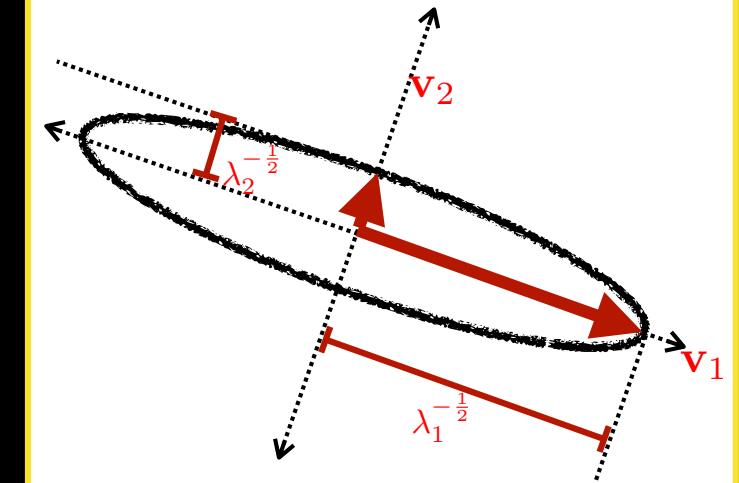
$$\sqrt{n}(\hat{\theta} - \theta^*) \xrightarrow{dist} \mathcal{N}(0, I(\theta^*)^{-1})$$

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



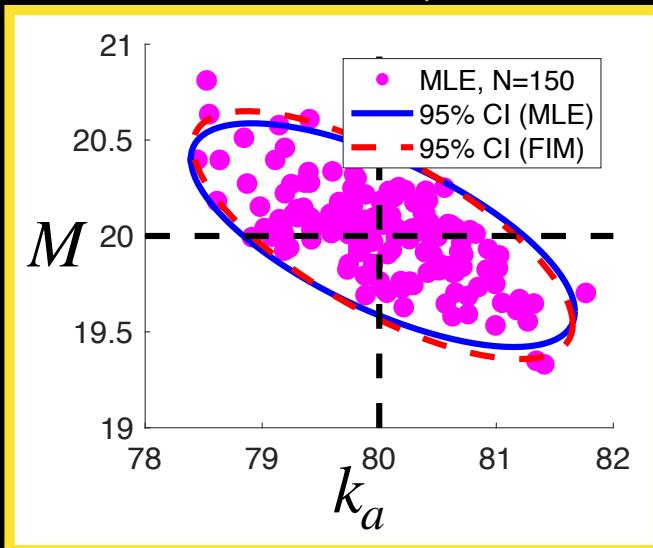


Example – FIM and MLE Variance for Genetic Toggle Switch

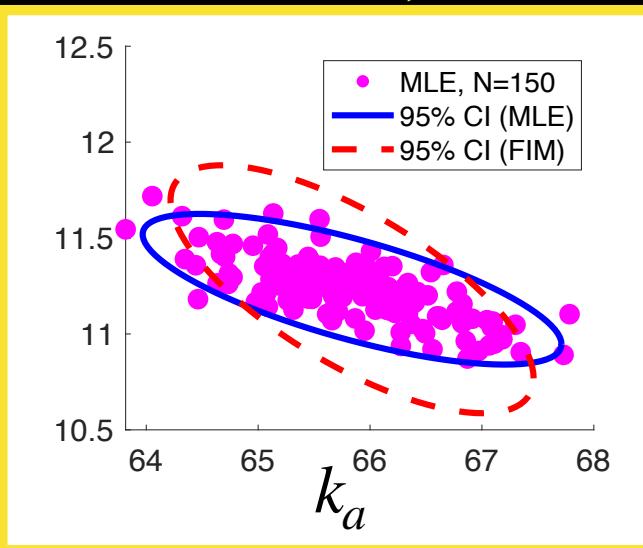


And we show that the spread of the MLE is well predicted by the FSP-FIM.

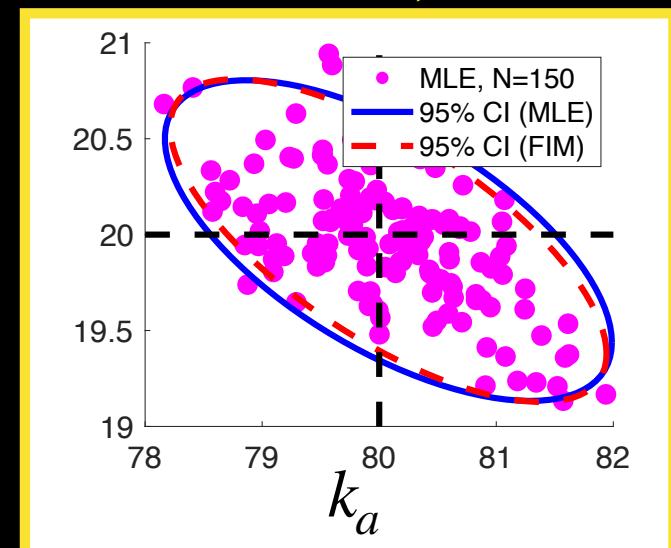
No Distortion in Data; Basic FSP



Binomial Distortion; Basic FSP



Binomial Distortion; FSP + PDO



The FSP-FIM accurately predicts the MLE variance.

The FIM also correctly estimates the impact of data distortions.

MATLAB CODE

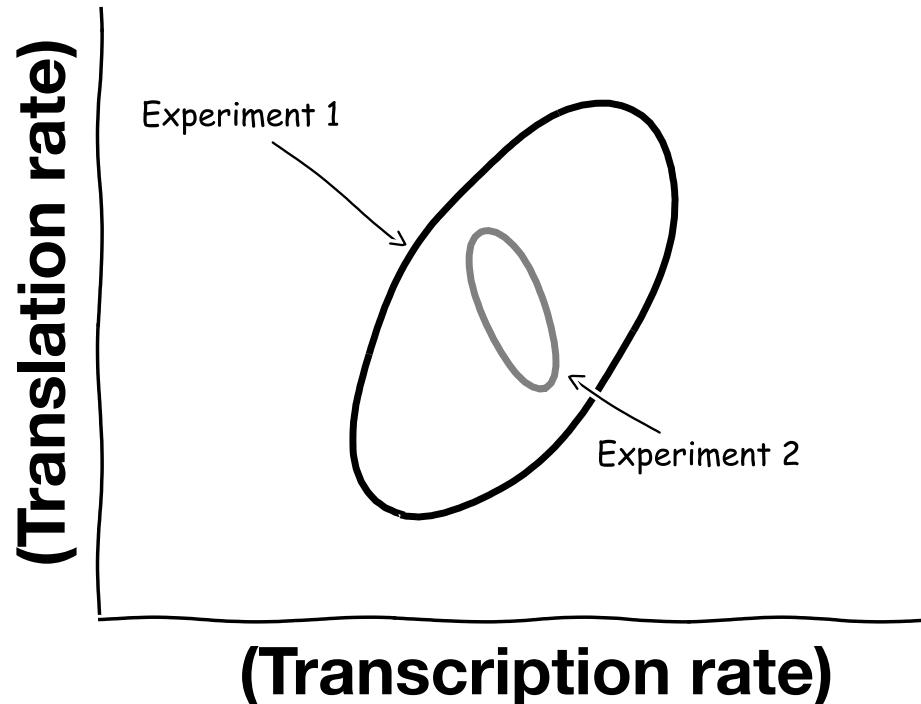
```
%% Compute FIM
cellCounts = A.ssaOptions.nSimsPerExpt*ones(size(A.tSpan)); % Number of cells in each experiment.
for m=1:3
    fimResults = B{m}.computeFIM(sensSoln.sens);
    % Compute the FIM for full observations and no distortion.
    [FIM{m},sFIMcov{m},fimMetrics{m}] = B{m}.evaluateExperiment(fimResults,cellCounts);
    fimFreePars = FIM{m}(A.fittingOptions.modelVarsToFit,A.fittingOptions.modelVarsToFit);
    B{m}.makeMLEFIMPlot(squeeze(MLE(m,:,:)),fimFreePars,0.95)
end
```



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

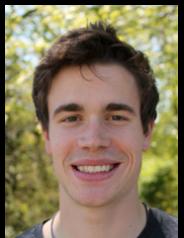


Different single-cell experiments reveal different amounts of information about different combinations of model parameters.



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

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



Example – Designing Optimal Experiment Times for Genetic Toggle Switch



We now suppose that we can measure up to 10,000 cells split among the possible time points: $t = \{0,1,2,\dots,99\}$.

Here we find the optimal set of times to use for the measurement.

MATLAB CODE

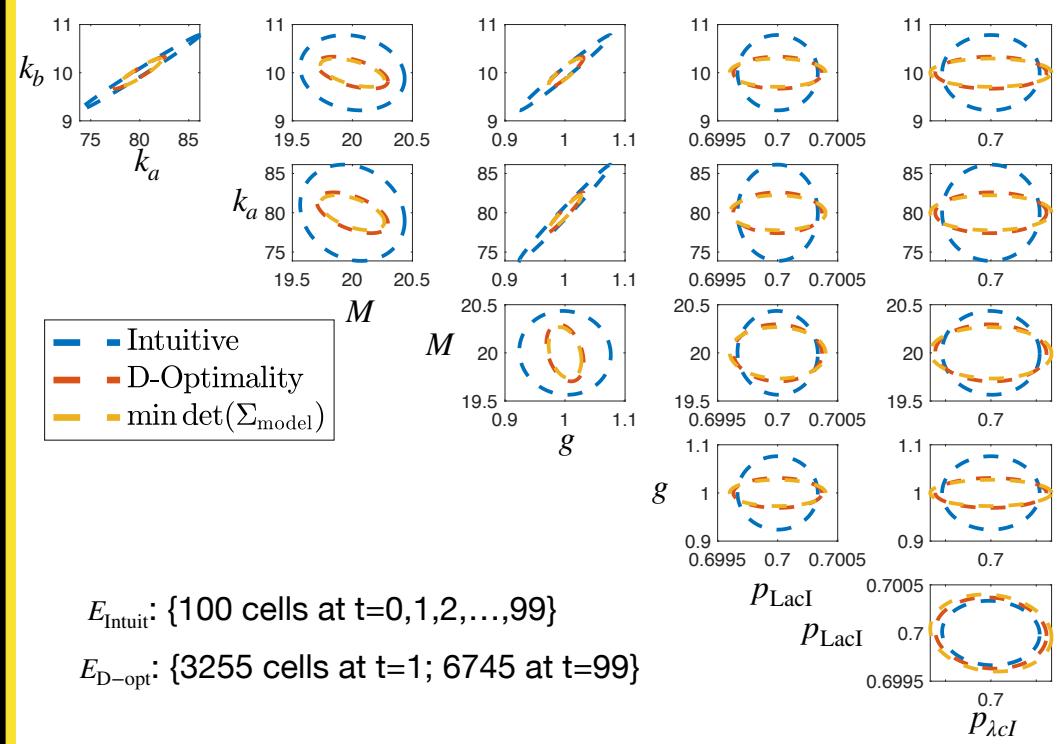
```
% Define Potential Experiments
A.tSpan = [0:1:99]; % Set times at which to compute distributions
A.solutionScheme = 'fspSens'; % Set scheme to FSP Sensitivity
[sensSoln,bounds] = A.solve; % Solve the sensitivity problem

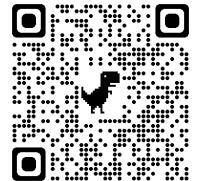
% Compute FIM with all parameters free to vary
A.fittingOptions.pdoVarsToFit = 'all';
fimResults = A.computeFIM(sensSoln.sens);

% Optimize cell counts using FIM
cellCounts = 100*ones(size(A.tSpan));
nCellsTotal = sum(cellCounts);
nCOpt=A.optimizeCellCounts(fimResults,nCellsTotal,'Determinant');

% compute total FIM
FIMintuit = A.evaluateExperiment(fimResults,cellCounts);
FIMoptDet= A.evaluateExperiment(fimResults,nCellsOptDet);

% Plot results
pars = {[A.parameters{:,2}],[0.7,0.7]}' ;
for i=1:5;
    for j = i+1:6
        subplot(5,5,(i-1)*5+j-1)
        A.makeMleFimPlot([],FIMintuit,[j,i],0.95,1,pars([j,i]));
        A.makeMleFimPlot([],FIMoptDet,[j,i],0.95,1,pars([j,i]))
    end;
end
```





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...)

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 right subspace is 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.

Acknowledgments and Collaborators (2019-2022)

Hog Signaling/Transcription Activation

Gregor Neuert, Vanderbilt
Guoliang Li, Vanderbilt
Hossein Jahnsaz, Vanderbilt
Jason Hughes, Vanderbilt
Douglas Shepherd, CU Denver

Single-mRNA Translation

Tim Stasevich, CSU
Tatsuya Morisaki, CSU
Kenneth Lyon, CSU
Amanda Koch, CSU
Linda Forero Quintero, CSU

Cell Heterogeneity in Algal Biofuels:

Babette Marrone, LANL
Claire Sanders, LANL
Elaheh Alizadeh, CSU

Cell-free Single-BioMolecule Kinetics:

Sabrina Leslie, McGill University
Shane Scott, McGill University

MultiScale Bayesian FSP tools:

Ania Baetica, CalTech
Thomas Catanach, Sandia

Soil Microbiome Machine Learning

John Dunbar Lab, LANL

Inflammation driven transcription

James Werner, LANL
Dan Kalb, LANL

q-bio Training and Textbook:

William S Hlavacek, LANL
Lev S. Tsimring, UCSD
Marek Kimmel, Rice University
Former students of the 2010-2017
q-bio Summer Schools



Graduate Students:

Zachary Fox **Joshua Cook**
Lisa Weber Eric Ron
Mohammad Tanhaemami
Michael May
Jaron Thompson
Will Raymond

Postdocs:

Luis Aguilera
Linda Forero Quintero
Huy Vo

Undergrads:

Elliot Djokic
Charis Ellis
Torin Moore
Zach Haigh
Charlotte Mitchell
Rachel Keating

Stuart McKnight
Katie Davis

Funding:

WMKF

NIH/NIGMS

DOE/BER

NSF CAREER



munsky@colostate.edu

The 2023 UQ-bio Summer School (featuring on-campus projects and online events)

The Third Annual Undergraduate Quantitative Biology (UQ-Bio) Summer School will be held May 30 – June 14, 2023, at the Colorado State University (Fort Collins CO, USA).

Applications Open Dec. 1, 2022 via the program website (<https://q-bio.org>)

Applications are due February 1, 2023. In-person participation will be limited to 30 students based on past coursework and statements of interests. All events are **free** to all participants.

School Overview: The UQ-bio Summer School is an annual event intended to help **undergraduate** and **first year graduate students** acquire essential skills to advance predictive modeling of cellular regulatory systems. The 2023 program will emphasize experimental and computational techniques useful to understand single-cell gene regulation.

The program will feature **daily live events** (Monday through Saturday) including research seminars from top scientists (5hrs/week), mentored project sessions (4hrs/week), hands-on software tutorials (10hrs/week), career discussion forums (3hrs/week), student presentations and hackathons (2hrs/week), laboratory tours/demonstrations, and more. The summer school is designed for undergraduate students and early-stage graduate students, or anyone with a quantitative background who is new to modeling cellular regulatory systems/networks.

The 5 modules of the 2022 UQ-bio summer school will be:

- Bootcamp Basics to get Started with Scientific Computing in Python (May, Online)
- Single-Cell Optical Microscopy Experiments and Image Processing (May 31 – June 3)
- Multivariable Statistics and Machine Learning for Single-Cell Data (June 5 - 7)
- Stochastic Simulations of Single-Cell Gene Regulatory Processes (June 8 - 10)
- Master Equation Analyses of Single-Cell Gene Regulatory Processes (June 12 - 14)
- (Optional) Quantitative Cell and Molecular Biology Symposium (June 15-16)

Learn more about these topics and watch some past lectures at: <https://q-bio.org/wp/home/>



Interested to be a mentor or speaker at the UQ-bio program? Please contact us directly!

For inquiries about the summer school, please contact:

Dr. Brian Munsky: qbio_summer_school@colostate.edu

For more information, please visit the school website at: <http://q-bio.org>

**UQ-BIO 2023,
May 30 - June 14, Fort Collins, CO**

An introduction to quantitative modeling using single-cell optical microscopy experiments.

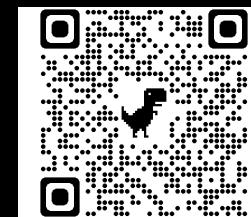
2022 Topics include:

- *Microscopy,
- *Image Processing,
- *Statistical Analyses,
- *Stochastic Processes,
- *Model Inference.

All lessons include hands-on Python programming.

Learn more and apply early at:

<https://q-bio.org>



Example UQ-Bio
CoLab Notebooks