

# User Guide: ISAP \*

Weilong Hu<sup>†</sup>  
Yannis Pantazis<sup>‡</sup>  
Markos A. Katsoulakis<sup>§</sup>

November 27, 2016

---

\*Version 1, 2015/01/18

<sup>†</sup>Email: [weilong@math.umass.edu](mailto:weilong@math.umass.edu)

<sup>‡</sup>Email: [pantazis@math.umass.edu](mailto:pantazis@math.umass.edu)

<sup>§</sup>Email: [markos@math.umass.edu](mailto:markos@math.umass.edu)

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>		
<b>2</b>	<b>Mathematical background</b>	<b>4</b>		
2.1	Stochastic chemical kinetics simulation . . . . .	4		
2.2	Stochastic sensitivity analysis . .	5		
<b>3</b>	<b>Installation</b>	<b>6</b>		
3.1	Requirements . . . . .	6		
<b>4</b>	<b>Defining the stochastic network</b>	<b>7</b>		
4.1	Create examples . . . . .	7		
4.2	Create basic data structures . . .	8		
4.3	Example . . . . .	8		
<b>5</b>	<b>Stochastic simulation</b>	<b>11</b>		
5.1	Stochastic Simulation Algorithm (SSA) [2] . . . . .	11	5.3	Mean Field Approximation (MFA) / Ordinary Differential Equation (ODE) [2] . . . . . 11
5.2	Chemical Langevin Equation (CLE) [2] . . . . .	11	5.4	Interpolating simulated time series 12
			<b>6</b>	<b>Stochastic sensitivity analysis</b> <b>13</b>
			6.1	Compute both RER and PFIM . . 13
			6.2	Compute RER . . . . . 13
			6.3	Compute PFIM . . . . . 13
			<b>7</b>	<b>Example problems</b> <b>15</b>
			7.1	Crypt Compartmental Model . . 15
			<b>8</b>	<b>Summary</b> <b>17</b>
			8.1	Structure of the package . . . . . 17
			8.2	Remarks . . . . . 18
				<b>Bibliography</b> <b>19</b>

## Introduction

Stochastic simulation and modeling play an important role to elucidate the fundamental mechanisms in complex biochemical networks. The parameter sensitivity analysis of reaction networks becomes a powerful mathematical and computational tool, yielding information regarding the robustness and the identifiability of model parameters. However, due to overwhelming computational cost, many existing sensitivity analysis approaches are not applicable to models with a high-dimensional parameter space.

Here we present an (I)nfo-theoretic (S)ensitivity (A)nalysis in (P)ath-space Matlab package, ISAP, that simulates stochastic process with various algorithms and implements a gradient-free approach to quantify the parameter sensitivities of stochastic chemical systems in true dynamics using pathwise Fisher Information Matrix (PFIM) which is suitable for high dimensional system due to sparsity. This computational tool provides a significant advantage when dealing with complex stochastic systems with a large number of parameters and species, which are typical in the modeling and simulation of complex biochemical phenomena and it is often regarded as a fast sensitivity screening tool for all existing software for sensitivity analysis of stochastic system.

## Mathematical background

In this section we briefly introduce the mathematical background of stochastic chemical kinetics simulation and stochastic sensitivity analysis.

### 2.1 Stochastic chemical kinetics simulation

ISAP considers problems modeled in the framework of stochastic chemical kinetics. Spatially uniform mixture of  $N$  chemical species which can interact through  $M$  specified chemical reaction channels is placed in a fixed volume container.

The reaction networks can be decomposed to coupled system of single directional chemical reactions of the form:



with the corresponding propensity function,  $a(\mathbf{X})$ , where  $X$ 's are the population of chemical species. The following example is a system of 2 chemical species ( $X_1, X_2$ ) with 3 reactions:

- $X_1 \rightarrow 2X_1$ ,
- $X_1 + X_2 \rightarrow 2X_2$
- $X_2 \rightarrow NULL$

with the 3 propensity functions  $a_j(\mathbf{X})$  where  $j = 1, 2, 3$ :

- $a_1(\mathbf{X}) = k_1 \times X_1$
- $a_2(\mathbf{X}) = k_2 \times X_1 \times X_2$
- $a_3(\mathbf{X}) = k_3 \times X_2$

According to the coupled biochemical reaction system, the package extracts a stoichiometry matrix which consists of the coefficients of the chemical species in the reactions and propensity functions which determine the reaction rates. With the stoichiometry matrix and the propensity functions, realization of all reactions taken place in the given time period can be generated according to any simulation algorithms by users' choice. In our package, there are three different algorithms available,

- Stochastic Simulation Algorithm (SSA) [2]: generates exact, discrete, single event, stochastic process.
- Chemical Langevin Equation (CLE) [2]: generates approximated, continuous-time stochastic process governed, stochastic differential equations.
- Mean Field Approximation / Ordinary Differential Equation (ODE) [2]: generates approximated, continuous-time, deterministic process.

Although the SSA method provides exact numerical realizations on the time evolution of a well-stirred chemically reacting system, it requires a great amount of simulation time. Many multi-scale approximations of the original SSA have been developed to accelerated simulation algorithms. The CLE method approximates the SSA method by assuming that all reactions fire simultaneously on a fixed time interval and the MFA method can be regarded as the CLE method without the noise term [1].

## 2.2 Stochastic sensitivity analysis

The parameter sensitivity can be estimated via calculating Pathwise Relative Entropy [1] which quantifies the loss of information of assuming a perturbed distribution instead of employing the true distribution. In the stationary regime, Relative Entropy increases linearly in time. Let  $Q_{[0,T]}^\theta$  denote the stationary path distribution of the process in the interval  $[0, T]$ . The pathwise Relative Entropy Rate (RER) [1],  $H(Q^\theta | Q^{\theta+\epsilon})$ , which is the time average of the Relative Entropy, is an appropriate time-independent measure of sensitivity since it measures the loss of information per unit time in path space due to an arbitrary  $\epsilon$ -perturbation of parameter combinations in any direction in the parameter space. A Taylor series expansion of RER in terms of  $\epsilon$  reveals that RER is locally a quadratic function of the parameter vector  $\epsilon \in \mathbf{R}^K$ .

$$H(Q^\theta | Q^{\theta+\epsilon}) = \frac{1}{2} \epsilon^T F_H(Q^\theta) \epsilon + O(|\epsilon|^3) \quad (2.2)$$

The pathwise Fisher Information Matrix (PFIM) [1],  $F_H(Q^\theta)$ , contains up to third order accuracy all the sensitivity information for the path distribution  $Q^\theta$  with parameter vector  $\theta$  for any perturbation direction  $\epsilon$ . Therefore, the computation of the PFIM is sufficient up to third order for the evaluation of all the local sensitivities of the path distribution around the parameter vector  $\theta$ . Unlike the traditional FIM which is computed only based on the stationary distribution, the pathwise FIM gathers more information by taking into consideration the dynamical aspects of the process as well. Moreover, an explicit formula for the pathwise FIM is given by

$$F_H(Q^\theta) = E_{\mu^\theta} \left[ \sum_j a_j^\theta(x) \nabla_\theta \log a_j^\theta(x) \nabla_\theta \log a_j^\theta(x)^T \right] \quad (2.3)$$

This explicit formula reveals that for a typical reaction network the PFIM has block-diagonal structure upon rearrangement of the parameter vector and PFIM is gradient-free as it only depends on the propensity functions  $a_j^\theta(x)$  and the current populations  $x$  which updates via the stoichiometry matrix. Since the stationary distribution  $\mu^\theta$  is usually unknown, an unbiased estimator for the PFIM [1] is computed numerically as an ergodic average. RER and the corresponding PFIM become computationally feasible as they are directly computable from the propensity functions. For biochemical reaction networks where model parameters differ by orders of magnitude, it is reasonable to carry out sensitivity analysis by adding perturbations which are proportional to the parameter magnitude. By perturbing the logarithm of the model parameters and use the chain rule, we obtain the logarithmically-scaled PFIM [1],  $F_H(Q^{\log \theta})$

$$\left( F_H(Q^{\log \theta}) \right)_{k,l} = \theta_k \theta_l \left( F_H(Q^\theta) \right)_{k,l} \quad \text{where } k, l = 1, \dots, K \quad (2.4)$$

---

## Installation

To install, download the ISAP compressed file [here](#).

### 3.1 Requirements

**Platform** This code was developed and tested on Mac OS X 10.6.8 and Windows 7. Other Mac OS X and Windows distributions should be okay.

**Matlab** A recent version of Matlab (2013a or later) with Symbolic Math and SimBiology Toolbox should be installed along with a supported compiler for that version (see [http://www.mathworks.com/support/sysreq/previous\\_releases.html](http://www.mathworks.com/support/sysreq/previous_releases.html) for a list of supported compilers for your release on your platform).

## Defining the stochastic network

### 4.1 Create examples

In order to build an example, user need to create three .txt files and store them in the '\examples\example\_name\input\_files' folder. The three input files include the following information of the reaction network:

- Coupled one direction reaction functions
- Initial populations of biochemical species
- Reaction rate constants for all reactions

The followings are three ways to create input files:

- Several examples(i.e. BIO520, EGFR, Lotka Volterra Equations, Michaelis Menten kinetics, p53 reaction network, Schlogl's model, and Yeast model) are provided in this software.
- User can also download examples from online databases, such as [BioModels Database](#). Our software provides a parser function, sbml2txt.m, to convert the downloaded .xml file to the our input files as the following steps:
  1. Go to [BioModels Database](#) and select the desired BioModel ID (e.g. BIOMD0000000520);
  2. Select the curated data (e.g. 'SBML L2 V4 (curated)' from 'download SBML') and download the xml file.
  3. Use sbml2txt.m function to create three input files.

Note: Current version of ISAP package is able to handle the SBML file examples with the following features, otherwise user have to create examples manually:

- No 'Rules': assignment rule and rate rule
- No 'Functions': abstract propensity function
- No 'Events'
- Names of the parameter are distinct to each other
- Only single compartment size is allowed
- Parameter value must be constant
- User can also follow the formats in section 4.3 to create examples(three input .txt files: reaction functions, initial populations and reaction rate constants) manually.

## Relevant functions

- `sbml2txt(file, outputfile)`

**Description** Remove the 'listOfModifiers' tag from xml file and rewrite if necessary. Load xml file into matlab and create three input files: reactions mechanism, reaction constants and initial populations.

**Inputs**

- 'file': Complete path of the location of .xml file
- 'outputfile': Name of the output folder

**Outputs**

- 'reactions mechanism': Coupled one direction reaction functions
- 'reaction constants': Reaction rate constants for all reactions
- 'initial populations': Initial populations of biochemical species

## 4.2 Create basic data structures

In order to simplify the computation, the package takes the three input files and further generate three .txt files as below.

- 'connectivity matrix': associates each species with an index and represents the reaction network as a matrix.
- 'propensity functions': extracts propensity functions from the reaction mechanism.
- 'stoichiometry matrix': consists of the coefficients of the chemical species in the reactions.

## 4.3 Example

Consider a reaction mechanism involving 3 reactants ( $X_1$ ,  $X_2$  and  $X_3$ ) and 3 reaction constants ( $k_1$ ,  $k_2$  and  $k_3$ ).

- $X_1 + X_2 \rightarrow X_3$  with propensity function  $k_1 \times X_1 \times X_2$
- $X_3 \rightarrow X_1 + X_2$  with propensity function  $k_2 \times X_3$
- $X_2 \rightarrow NULL$  with propensity function  $k_3 \times X_2$

Input files:

1. `reactants.txt`: It lists the reactions with the propensity functions, i.e.,

- $X_1 + X_2 = X_3$     ' $k_1 * X_1 * X_2$ '
- $X_3 = X_1 + X_2$     ' $k_2 * X_3$ '
- $X_2 = NULL$     ' $k_3 * X_2$ '

2. `initial_population.txt`: It assigns numbers to each reactant, i.e.,

- $NULL = 0$
- $X_1 = 100$
- $X_2 = 100$
- $X_3 = 100$

3. `reaction_constants`: It assigns numbers to each reactant constant, i.e.,



- $k_1 = 62849$
- $k_2 = 0.70599$
- $k_3 = 1$

Output files:

1. stoichiometry.txt: The stoichiometry of participating species in reactions is generated in the following format:

Stoich 1	Stoich 2	Stoich 3	Stoich 4
-1	-1	1	0
-1	-0	1	1
-1	-0	0	0

2. connectivity.txt: The connectivity is generated in the following format and shows which species are involved in each reaction:

Connectivity 1	Connectivity 2	Connectivity 3	Connectivity 4
2	3	4	0
4	0	2	3
3	0	1	0

3. propensities.txt: The reaction propensity functions are generated in the following format:

- $k_1 * X_1 * X_2$
- $k_2 * X_3$
- $k_3 * X_2$

## Relevant functions

- rxn2csm(fname)

**Description** Read three files with the reactions mechanism, reaction constants and initial populations to creates the connectivity matrix, stoichiometric matrix and the propensity functions.

### Inputs

- 'fname': Directory with file name.

### Outputs

- 'stoichiomatrix.txt': Stoichiometry of participating species in reactions is generated.
- 'connectivity.txt': Connectivity is generated.
- 'propensities.txt': Reaction propensity functions are generated.

- initialization\_lin(rname)

**Description** Initialization for simulation and sensitivity analysis. Reactions, species, reaction rates, connectivity and initial configuration are specified. Linear-scale sensitivity analysis is considered.

### Inputs

- 'rname': File name that describes the model.

### Outputs

- 'data.M': Number of reactions.
- 'data.N': Number of species.

- 'data.K': Number of parameters.
  - 'data.X': Initial population of the system.
  - 'data.pX': Chemical species pattern X(i).
  - 'data.pC': System parameters pattern con(i).
  - 'data.theta': Name of system parameters.
  - 'data.prop': Cell vector with the propensity functions as matlab handles.
  - 'data.props': Cell vector with the propensity functions as strings.
  - 'data.vprops': Cell vector with the propensity functions as strings with numeric coefficients.
  - 'data.m\_glpf': Matrix of the gradient of log of propensity functions.
  - 'data.cv\_glpf': Vector of the non-zero elements of the symbolic PFIM.
  - 'data.prop\_index': Index of the non-zero elements of the symbolic PFIM.
  - 'data.con': Value of reaction constants.
  - 'data.sname': Name of chemical species.
  - 'data.stoichio': Stoichiometric matrix.
  - 'data.cm\_rea': Connectivity matrix for the reactions.
  - 'data.cm\_spe': Connectivity matrix for the species.
  - 'get\_rate.m' file: Compute reaction rate for the given reaction.
  - 'get\_PFIM.m' file: Compute element of PFIM value for the given reaction.
- initialization\_log(rname)

**Description** Initialization for simulation and sensitivity analysis. Reactions, species, reaction rates, connectivity and initial configuration are specified. Logarithmic-scale sensitivity analysis is considered.

**Inputs**

- 'rname': File name that describes the model.

**Outputs**

- 'data': Data structure array (same as the linear case in section 4.3).

## Stochastic simulation

Our package provides three different ways to conduct stochastic simulation.

### 5.1 Stochastic Simulation Algorithm (SSA) [2]

Stochastic Simulation Algorithm generates exact, discrete, single event, stochastic process.

#### Relevant functions

- `reaction_ssa(data)`

**Description** Simulation for well-mixed reaction systems with Stochastic simulation algorithm.

#### Inputs

- 'data': Data structure array (defined by initialization in section 4.3).

#### Outputs

- 'output.t': Time vector.
- 'output.species\_ts': The concentrations of each species at each time step.

### 5.2 Chemical Langevin Equation (CLE) [2]

Chemical Langevin Equation generates approximated, continuous, stochastic differential trajectories.

#### Relevant functions

- `reaction_cle(data)`

**Description** Simulation for well-mixed reaction systems with Chemical Langevin Equation.

#### Inputs

- 'data': Data structure array (defined by initialization in section 4.3).

#### Outputs

- 'output.t': Time vector.
- 'output.species\_ts': The concentrations of each species at each time step.

### 5.3 Mean Field Approximation (MFA) / Ordinary Differential Equation (ODE) [2]

Mean Field Approximation / Ordinary Differential Equation generates approximated, continuous, deterministic differential trajectories.

## Relevant functions

- `reaction_mfa(data)`

**Description** Simulation for well-mixed reaction systems with Ordinary Differential Equation.

### Inputs

- `'data'`: Data structure array (defined by initialization in section 4.3).

### Outputs

- `'output.t'`: Time vector.
- `'output.species_ts'`: The concentrations of each species at each time step.

## 5.4 Interpolating simulated time series

Different simulation algorithms provide sample points from different sampling time instants. The package can homogenize the time interval into equal partitions by interpolation and re-sampling.

## Relevant functions

- `sim_rxn(data, dname, method)`

**Description** Simulate the reaction network. Each time-series is interpolated at given time instants and stored in `.mat` files.

### Inputs

- `'data'`: Data structure array (defined by initialization in section 4.3).
- `'dname'`: Directory for interpolated simulation.
- `'method'`: Simulation algorithm (SSA, CLE or MFA)

### Outputs

- `'mat'` file: Interpolated simulation data file.

## Stochastic sensitivity analysis

ISAP makes stochastic sensitivity analysis flexible by providing functions to compute the RER only, PFIM only or both.

### 6.1 Compute both RER and PFIM

With input time series, ISAP computes both RER and PFIM for all parameters assuming some perturbations are added to the original parameter values.

#### Relevant functions

- `compute_PFIM_RER(data, dname)`

**Description** Compute both PFIM and RER from the stored time-series.

#### Inputs

- 'data': Data structure array (defined by initialization in section 4.3).
- 'dname': directory name of population data.

#### Outputs

- 'RER': Vector of RERs corresponds to the perturbation matrix.
- 'PFIM': Pathwise Fisher Information Matrix.

### 6.2 Compute RER

With input time series, ISAP computes the relative entropy rates for all parameters assuming some perturbations are added to the original parameter values.

#### Relevant functions

- `comp_RER(X, data)`

**Description** Compute a vector of RERs with input time series(species population only, no time) and perturbation matrix (each row represents a perturbation vector to all parameters).

#### Inputs

- 'X': Time series of the populations of chemical species.
- 'data': Data structure array (defined by initialization in section 4.3).

#### Outputs

- 'RER': Vector of RERs corresponds to the perturbation matrix.

### 6.3 Compute PFIM

With input time series, ISAP computes the Fisher information matrix for all parameters assuming some perturbations are added to the original parameter values.

## Relevant functions

- `comp_PFIM(X, data)`

**Description** Compute PFIM with input time series(species population only, no time).

### Inputs

- 'X': Time series of the populations of chemical species.
- 'data': Data structure array (defined by initialization in section 4.3).

### Outputs

- 'PFIM': Pathwise Fisher Information Matrix.

## Example problems

Several examples are included in the examples directory:

**BIO520** Model of the colon crypt(3 species, 7 reactions, 11 parameters).

**EGFR** Epidermal growth factor receptor model(23 species, 47 reactions, 50 parameters).

**Lotka Volterra** The Lotka Volterra predator-prey model(2 species, 3 reactions, 3 parameters).

**Michaelis Menten** Michaelis Menten kinetics(2 species, 2 reactions, 2 parameters).

**p53** Tumor protein p53 model(3 species, 5 reactions, 7 parameters).

**Schlogl** Schloegl model(1 species, 4 reactions, 4 parameters).

**yeast** Yeast model organism(7 species, 8 reactions, 8 parameters).

### 7.1 Crypt Compartmental Model

Here we will take Crypt Compartmental Model, the model of the colon crypt, as an example to explain how to download and transform examples from the BioModels Database and then introduce the test functions in ISAP. In general, it is easy to modify the test functions to accommodate user defined reaction network.

1. Create example folders in following directory: '`\examples\BIO520\input_files`'
2. Go to <http://www.ebi.ac.uk/biomodels-main/BIOMD0000000520>;
3. Select 'SBML S2 V4 (curated)' from 'download SBML' and download the xml file to '`\xml_models`'.
4. Set file='`xml_models/BIOMD0000000520.xml`'.
5. Set outputfile='`examples/BIO520/input_files/BIO520`'.
6. Use `sbml2txt(file, outputfile)` function to create three input files.
7. Run `example_methods_BIO520` and `example_isap_BIO520` to do simulation and perform sensitivity analysis respectively.

#### Relevant functions

- `example_methods_BIO520`

**Description** Simulate the BIO520 reaction network with SSA, CLE and MFA. Interpolate the simulated time series.

#### Outputs

- Plot the simulation time series of all species from different algorithms side by side.

- `example_isap_BIO520`

**Description** Simulate the BIO520 reaction network with SSA, CLE or MFA and interpolate the simulated time series. Add 10 percents perturbation on the parameter values and conduct either linear-scale or log-scale parameter sensitivity analysis by computing RER and PFIM.

**Outputs**

- Plot 3D barplot for the elements of PFIM.
- Plot the RER computed directly and utilizing PFIM side by side with confidence intervals.



## Summary

### 8.1 Structure of the package

Names of the m-functions and their hierarchy:

1. `example_methods_{BIO520|EGFR|LotkaVolterra|MichaelisMenten|p53|Schlogl|yeast}.m`

- `rxn2csm.m`
- `initialization_{lin|log}.m`
  - `create_getrate.m.m`
  - `create_getPFIM.m.m`
- `sim_rxn.m`
  - `reaction_ssa.m`
    - \* `reaction_rate.m`
      - `get_rate.m`
    - \* `update_rates.m`
  - `reaction_cle.m`
  - `reaction_mfa.m`
- `plot_species.m`

2. `example_isap_{BIO520|EGFR|LotkaVolterra|MichaelisMenten|p53|Schlogl|yeast}.m`

- `rxn2csm.m`
- `initialization_{lin|log}.m`
  - `create_getrate.m.m`
  - `create_getPFIM.m.m`
- `sim_rxn.m`
  - `reaction_ssa.m`
    - \* `reaction_rate.m`
      - `get_rate.m`
    - \* `update_rates.m`
  - `reaction_cle.m`
  - `reaction_mfa.m`

- `compute_PFIM_RER.m`
  - `comp_PFIM.m`
    - \* `update_PFIM.m`
      - `get_PFIM.m`
  - `comp_RER.m`
    - \* `update_RER_per.m`
      - `rate_eps.m`
- `plot_PFIM.m`
- `plot_RER.m`

## 8.2 Remarks

1. Do not use the words 'X' and 'con' as a variable name of a reaction species or a reaction constant. They are used as keywords.
2. Variable names must start with a letter. Arithmetic symbols are not allowed to name variables.
3. All the elements in the three input files should be separated by spaces.
4. At most 2 reactants and 2 outputs. Single reactant and output should be put in the first and third position respectively. If no reactants or outputs, use NULL instead.
5. Use equal sign instead of right arrow in the `reactants.txt` file.
6. The propensity functions in the `reactants.txt` file should be quoted by single quotation marks.
7. Unit of all population must be the same.
8. The first species in the `initial_population.txt` file is always NULL.
9. When switching to a different reaction model, it is advisable to delete the function `get_PFIM.m` and `get_rate.m`.

---

## Bibliography

- [1] Pantazis Y, Katsoulakis M, Vlachos D *Parametric Sensitivity Analysis for Biochemical Reaction Networks based on Pathwise Information Theory* Bioinformatics, 14: 331-350, 2013
- [2] Gillespie D *Stochastic Simulation of Chemical Kinetics* Annu Rev Phys Chem 58: 35-55, 2007.
- [3] Salis H, Kaznessis Y *Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions* J Chem Phys 122(5): 54103-54116, 2005.
- [4] Slepoy A, Thompson A, and Plimpton S *A constant-time kinetic Monte Carlo algorithm for simulation of large biochemical reaction networks* J Chem Phys 128: 205101-205108, 2008.