



## ISAP-MATLAB Package for Sensitivity Analysis of High-Dimensional Stochastic Chemical Networks

**Weilong Hu**  
University of  
Massachusetts-Amherst

**Yannis Pantazis**  
University of  
Massachusetts-Amherst

**Markos A. Katsoulakis**  
University of  
Massachusetts-Amherst

---

### Abstract

Stochastic simulation and modeling play an important role to elucidate the fundamental mechanisms in complex biochemical networks. The parametric 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, parametric sensitivity analysis is an extremely challenging problem for stochastic models with a high-dimensional parameter space and for which existing approaches are very slow. Here we present an information-theoretic sensitivity analysis in path-space (**ISAP**) MATLAB package that simulates stochastic processes with various algorithms and most importantly implements a gradient-free approach to quantify the parameter sensitivities of stochastic chemical reaction network dynamics using the pathwise Fisher information matrix (PFIM; [Pantazis, Katsoulakis, and Vlachos 2013](#)). The sparse, block-diagonal structure of the PFIM makes its computational complexity scale linearly with the number of model parameters. As a result of the gradient-free and the sparse nature of the PFIM, it is highly suitable for the sensitivity analysis of stochastic reaction networks with a very large number of model parameters, which are typical in the modeling and simulation of complex biochemical phenomena. Finally, the PFIM provides a fast sensitivity screening method ([Arampatzis, Katsoulakis, and Pantazis 2015](#)) which allows it to be combined with any existing sensitivity analysis software.

*Keywords:* stochastic biochemical networks, parametric sensitivity analysis, high-dimensional parameter space, pathwise Fisher information matrix, fast sensitivity screening.

---

## 1. Introduction

Complex reaction networks modeled by stochastic population dynamics defined on a graph representing interactions (reactions) between species, are ubiquitous across many scientific disciplines, e.g., chemistry and chemical engineering, systems biology, bioinformatics, epi-

Event	Reaction	Propensity function
1	$A + 2X \rightarrow 3X$	$k_1 \times A \times X \times (X - 1)/2$
2	$3X \rightarrow A + 2X$	$k_2 \times X \times (X - 1) \times (X - 2)/6$
3	$B \rightarrow X$	$k_3 \times B$
4	$X \rightarrow B$	$k_4 \times X$

Table 1: Reactions and propensity functions for the Schlögl model.

demology, game theory and time series analysis (Goutsias and Jenkinson 2013). Stochastic simulation and modeling provide an intrinsic understanding of the interplay between complexity and robustness of biochemical reaction networks and have attracted an increasing amount of attention from the modeling, computational and statistical communities. We provide a mathematical description of the exact stochastic simulation algorithm and its connections with approximate models of reaction networks in Section 2. Next we use a very simple example to introduce stochastic simulation and parametric sensitivity analysis. The robustness of a system is a property that allows a system to maintain its functions against internal and external perturbations. Due to its bimodality behavior, the Schlögl model (Vellela and Qian 2009) is often considered as a good example that motivates the need for stochastic simulations. The Schlögl model (Vellela and Qian 2009; Pantazis and Katsoulakis 2013) describes a well-mixed chemical reaction network among three chemical species  $A$ ,  $B$  and  $X$  interacting through 4 reactions governed by the corresponding propensity functions (Table 1). The concentrations  $A$ ,  $B$  are kept constant and the reaction rates  $k_1, \dots, k_4$  are the parameters of the model.

Sensitivity analysis provides a mathematical tool to describe the robustness of a system to perturbations by determining the (usually unknown) most/least sensitive directions in the potentially high-dimensional parameter space. The eigenvalues of the proposed pathwise Fisher information matrix (PFIM) are an approximation of the relative entropy rates (RER), which serve as a measure of the system parameter sensitivities. As shown in Figure 1, the most sensitive direction corresponds to the eigenvector with the eigenvalue 396 while the least sensitive direction corresponds to the eigenvector with eigenvalue 30. To visualize the impacts of the perturbation on different sensitive-leveled parameters towards the reaction system, we compared the stationary distributions of the population of chemical species  $X$  for the following cases: the unperturbed process, 5% of perturbation to the most sensitive parameter  $k_1$  and 5% of perturbation to least sensitive parameter  $k_3$ . Figure 2 indicates that the reaction network is more robust to perturbation on the less sensitive parameters as the stationary distribution departs much more from the unperturbed distribution when perturbing the most sensitive parameter comparing to the least sensitive parameter. As a result, the parameter sensitivity analysis of complex networks can be considered as an essential mathematical and computational tool to measure the robustness and the identifiability of model parameters (Pantazis *et al.* 2013).

In recent years there has been a significant development in sensitivity analysis tools and software packages for low-dimensional stochastic processes. Existing modeling software packages like **COPASI** (Hoops *et al.* 2006), **PottersWheel** (Maiwald and Timmer 2008), **SensSB** (Rodriguez-Fernandez and Banga 2010) and **StochSS** (StochSS Org. 2013) analyze parameter sensitivity with deterministic modeling of dynamical systems. Packages **COPASI** and **SensSB** perform sensitivity analysis using finite-difference methods. Package **PottersWheel** introduced mean optimal transformations (Hengl, Kreutz, Timmer, and Maiwald 2007), a non-

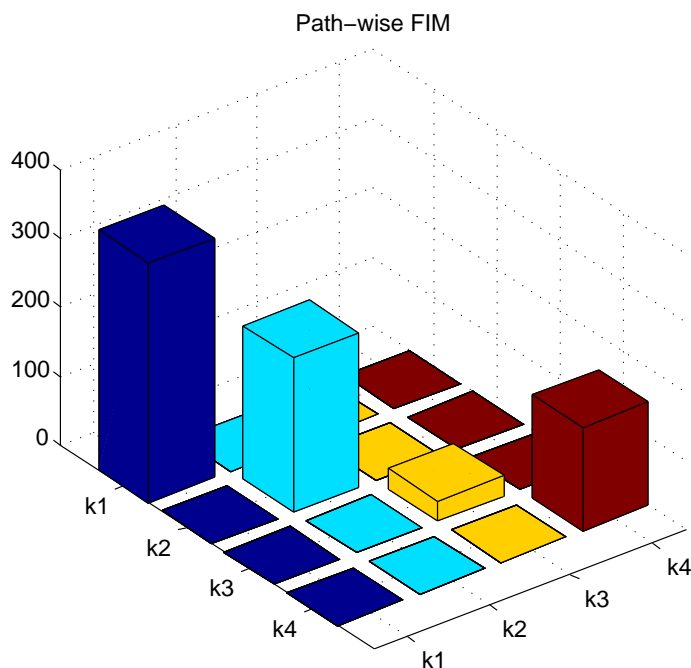


Figure 1: The proposed pathwise FIM (PFIM) from the Schlögl model. A large value of the diagonal element indicates high sensitivity of the corresponding parameter.

parametric bootstrap-based algorithm, and the profile likelihood (Raue *et al.* 2009) method for sensitivity analysis. Package **StochSS** implements forward sensitivity analysis (Maly and Petzold 1996; Feehery, Tolsma, and Barton 1997) for the deterministic (ODE-based) solver based on the **SUNDIAL**'s **CVODES** solver (Hindmarsh *et al.* 2005). Later several stochastic sensitivity analysis packages have been designed for stochastic systems. For example, package **SPSens** (Sheppard, Rathinam, and Khammash 2013) carries out stochastic parameter sensitivity analysis with several finite-difference estimators including the independent random number method, the common random numbers method (Sheppard, Rathinam, and Khammash 2012), the common reaction path (Rathinam, Sheppard, and Khammash 2010) method and the coupled finite difference (Anderson 2012) method along with the Girsanov likelihood ratio (Plyasunov and Arkin 2007) and the regularized pathwise derivative (Sheppard *et al.* 2012) method. Another software package, **StochSens** (Komorowski, Žurauskiene, and Stumpf 2012), is developed for stochastic sensitivity analysis using the linear noise approximation (Komorowski, Costa, Rand, and Stumpf 2011) and Fisher information matrix.

However, for stochastic models with a high-dimensional parameter space as it is typically the case in biochemical reaction networks, the existing sensitivity analysis approaches are often inapplicable due to the overwhelming computational cost since they either rely primarily on finite-difference or employ various approximation in the dynamics that are not suitable for small population sizes. Traditional finite-difference methods break down for high-dimensional systems due to the complexity of the systems, i.e., too many directional derivatives combined with the high variance of each finite difference estimator (McGill, Ogunnaike, and Vlachos 2012). On the other hand, recently there has been introduced a sensitivity method (Pantazis *et al.* 2013) which, firstly, is suitable for high-dimensional systems and, secondly, is able to deal with the exact stochastic dynamics instead of using approximations. The pathwise

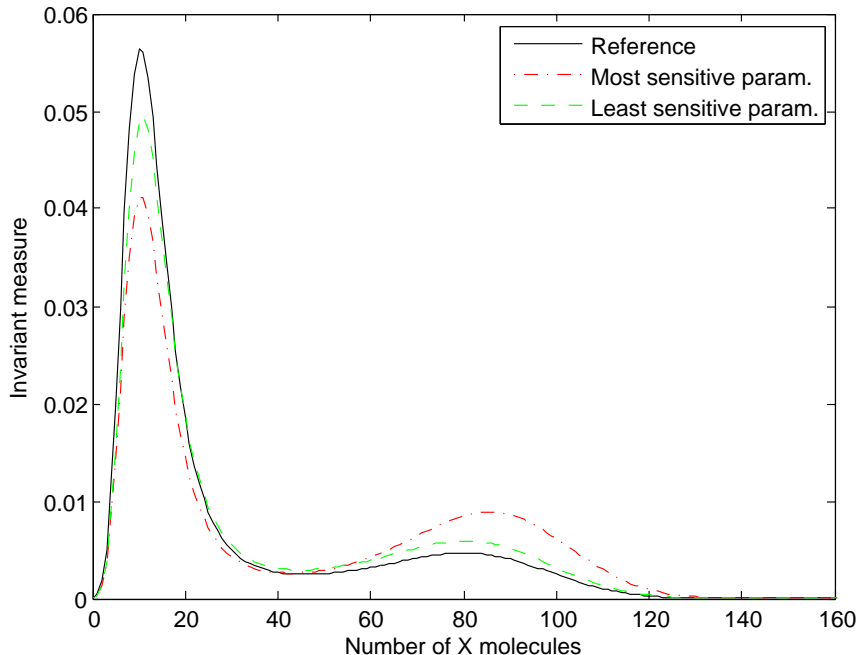


Figure 2: The stationary distributions of the population of chemical species  $X$  for the unperturbed process (black solid line), process after perturbed on the most sensitive parameter  $k_1$  (red dotted line) and process after perturbed on the least sensitive parameter  $k_3$  (green dashed line).

sensitivity analysis method in [Pantazis and Katsoulakis \(2013\)](#) quantifies the loss of information caused by parameter perturbations between time series distributions. The proposed method is realized by employing the relative entropy rate (RER) which provides a measure of the information loss per unit time in path space after introducing an arbitrary perturbation in the parameter vector. An associated pathwise Fisher information matrix (PFIM) which is derived as an approximation of the RER provides a gradient-free approach (i.e., a sensitivity analysis methodology which does not depend on the perturbation) to quantify parameter sensitivities. The block-diagonal structure of the PFIM in reaction networks ([Pantazis et al. 2013](#)) indicates its sparsity which in turn reduces the computational complexity of the stochastic analysis using the PFIM for a  $K$ -parameter model from  $\mathcal{O}(K^2)$  to  $\mathcal{O}(K)$ . Being a gradient-free sensitivity analysis method, this pathwise approach shows great advantage in computation speed especially when dealing with complex stochastic systems with high-dimensional parameter space. Due to the property that both RER and PFIM rely only on information for local dynamics rather than the equilibrium PDFs, the pathwise sensitivity method is suitable for non-equilibrium steady state systems (i.e., systems for which we do not know explicitly the equilibrium distribution), which is the case in most reaction networks, driven, or externally coupled systems. Furthermore, the pathwise sensitivity method can be also used as a computational efficient way to screen out insensitive parameters. Indeed, based on a new sensitivity bound (11) which incorporates the variance of the quantity of interest and the PFIM an accelerated sensitivity analysis strategy has been proposed ([Arampatzis et al. 2015](#)).

Here we present an information-theoretic sensitivity analysis in path-space (**ISAP**) MATLAB (The MathWorks Inc. 2017) package which simulates deterministic and stochastic reaction networks with three algorithms and primarily focuses on implementing symbolic and numerical procedures to access stochastic sensitivity analysis by calculating both RER and PFIM for chemical kinetics models as proposed in Pantazis *et al.* (2013). In particular, this package delivers numerical realizations of time series using an array of options (e.g., mean field approximation/ordinary differential equation, chemical Langevin approximation and exact stochastic simulation algorithm; Gillespie 2007) depending on the desired accuracy, available resources and simulation regimes.

Compared to other existing software for sensitivity analysis, our package can provide efficient sensitivity analysis on complex stochastic systems especially for those with high-dimensional parameter space due to the sparsity and block-diagonal structure of PFIM which can improve scalability by reducing the computation complexity from quadratic,  $\mathcal{O}(K^2)$ , down to linear,  $\mathcal{O}(K)$ , with  $K$  being the number of model parameters (Arampatzis *et al.* 2015). Furthermore, our package is compatible to user-input time series data from other sources and can work with any other sensitivity packages in a complementary manner as a sensitivity screening tool that is cheap and efficient to implement as a first step in the sensitivity analysis using any of the existing methods and software (Arampatzis *et al.* 2015). Several reaction network models including p53, EGFR, Lotka-Volterra and yeast growth model are provided in the software to demonstrate the applicability of the package and more examples can be found in online databases such as the BioModels Database (Le Novère *et al.* 2006, <http://www.ebi.ac.uk/biomodels-main/publmodels>).

## 2. 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

In the same mathematical context of continuous time Markov chains we can also consider well-mixed chemical reaction networks. With these models, however, one ignores the spatial locations of individual molecules and it is only the molecular populations for individual species that are needed to define the dynamical model.

We consider a well-mixed chemical reaction network with  $N$  chemical species,  $\mathbf{C} = \{C_1, \dots, C_N\}$ , which can interact through  $M$  specified chemical reaction channels,  $\mathbf{R} = \{R_1, \dots, R_M\}$ , governed by  $K$  model parameters,  $\theta = \{\theta_1, \dots, \theta_K\}$ . The state of the system at any time  $t \geq 0$  is denoted by a  $N$ -dimensional vector  $\mathbf{X}_t = [X_{t,1}, \dots, X_{t,N}]^\top$  where  $X_{t,i}$  is the population of chemical species  $C_i$  at time  $t$ . Let  $\nu$  be the  $N \times M$ -dimensional stoichiometry matrix consisting of  $\nu_{i,j}$ , the stoichiometric coefficient of species  $C_i$  in reaction  $R_j$ . For each reaction  $j$ , we have a corresponding  $N$ -dimensional stoichiometry vector  $\nu_j$  that identifies the numbers of molecules of every species that are produced and consumed in each instance of a reaction of type  $j$ . Provided that the reaction network at time  $t$  is in state  $\mathbf{X}_t = \mathbf{x}$ , a propensity function,  $a_j^\theta(\mathbf{x})$ , is defined to identify the approximate probability  $a_j^\theta(\mathbf{x})dt$  that the  $j$ th reaction occurs in the

time interval  $[t, t + dt]$ . For example in mass action kinetics the propensities have the form,

$$a_j(\mathbf{x}) = \theta_j \times g_j(\mathbf{x}), \quad j = 1, \dots, M, \quad (1)$$

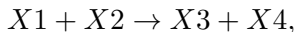
where  $g$  is a polynomial. Other types of propensities include the Michaelis-Menten kinetics with propensity function

$$a_k^\theta(\mathbf{x}) = \frac{\theta_k \times \mathbf{x}_A}{\theta_{k'} + \mathbf{x}_A}. \quad (2)$$

Here we consider its simplest form, where it gives the rate for what is called a single-substrate reaction without intermediate. It identifies the rate at which molecules of species  $A$  transform to species  $B$  (i.e., the reaction  $A \rightarrow B$ ).

Mathematically,  $\{\mathbf{X}_t\}_{t \in \mathbb{R}_+}$  is a countable state space continuous time Markov chain (CTMC) with transition rates,  $a_j^\theta(\cdot)$ ,  $j = 1, \dots, M$ . The clock of the jumps from a current state  $\mathbf{x}$  to a new state  $\mathbf{x}' = \mathbf{x} + \nu_j$  is determined by the transition rates with transition probabilities  $\frac{a_j^\theta(\mathbf{x})}{a_0^\theta(\mathbf{x})}$ , where  $a_0^\theta(\mathbf{x}) := \sum_{j=1}^J a_j^\theta(\mathbf{x})$  is the total rate. To fully understand the reaction network, the initial population of all chemical species should also be given.

**ISAP** considers problems modeled in the framework of stochastic chemical kinetics where the reaction networks can be decomposed to coupled systems of single directional chemical reactions of the form:



with the corresponding propensity function,  $a(\mathbf{x})$ , where  $\mathbf{x}$ '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  $j = 1, 2, 3$ :

- $a_1(\mathbf{x}) = k1 \times x_1$ ;
- $a_2(\mathbf{x}) = k2 \times x_1 \times x_2$ ;
- $a_3(\mathbf{x}) = k3 \times x_2$ ;

which in this example (not necessary in general) follow mass-action kinetics law. 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. Given 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 chosen by the user. In our package, there are three different available options:

- Stochastic simulation algorithm (SSA; Gillespie 2007): generates the exact, discrete-event stochastic process.

- Chemical Langevin equation (CLE; Gillespie 2007): generates an approximated, continuous time stochastic process governed by stochastic differential equations.
- Mean field approximation (MFA)/ordinary differential equation (ODE; Gillespie 2007): generates an approximated, continuous time, deterministic process governed by ODEs.

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 especially when multiscale reaction networks are studied. Many multiscale approximations of the original SSA have been developed to accelerate the simulation algorithms. The CLE method approximates the stochastic process by assuming that all reactions fire simultaneously on a fixed time interval while the MFA method can be regarded as the CLE method without the noise term (Pantazis *et al.* 2013).

Mean-field models for reaction networks are commonplace due to their computational efficiency. Here the stochastic process is written as

$$\mathbf{X}(t) = x(t) + \eta \times \xi(t), \quad (3)$$

where  $x(t)$  is the mean of the process,  $\xi(t)$  is the stochastic zero-mean part while  $\eta$  is the amplitude of the stochastic term which is proportional to the inverse square root of the reactant populations. As a consequence, the fluctuations of the time-evolving species populations become vanishingly small compared to the deterministic contributions for large populations and the deterministic term whose dynamics are governed by the ordinary differential equation (ODE) system (Equation 4) dominates the process.

$$\dot{x}_i(t) = \sum_{j=1}^M \nu_{j,i} \times a_j^\theta(x(t)), \quad i = 1, \dots, N. \quad (4)$$

The mean field approximation (Equation 4), as well as a higher order stochastic differential equation (SDE) approximation called chemical Langevin can be readily obtained from the stochastic dynamics.

## 2.2. Pathwise sensitivity analysis

The parameter sensitivities are estimated via calculating the relative entropy between time series (or path) distributions (Pantazis and Katsoulakis 2013) which quantifies the loss of information when assuming a perturbed distribution instead of employing the true distribution. In the stationary regime, pathwise relative entropy increases linearly in time. We refer to Pantazis and Katsoulakis (2013) and Pantazis *et al.* (2013) for definitions, derivations and properties of the information quantities discussed here. Let  $Q^\theta$  denote the path distribution of the process with parameter vector  $\theta \in \mathbb{R}^K$ . The relative entropy rate (RER; Pantazis and Katsoulakis 2013),  $H(Q^\theta|Q^{\theta+\epsilon})$ , which is the time average of the relative entropy between path distributions, is a 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 perturbation vector  $\epsilon \in \mathbb{R}^K$ :

$$H(Q^\theta|Q^{\theta+\epsilon}) = \frac{1}{2} \epsilon^\top F_H(Q^\theta) \epsilon + O(|\epsilon|^3). \quad (5)$$



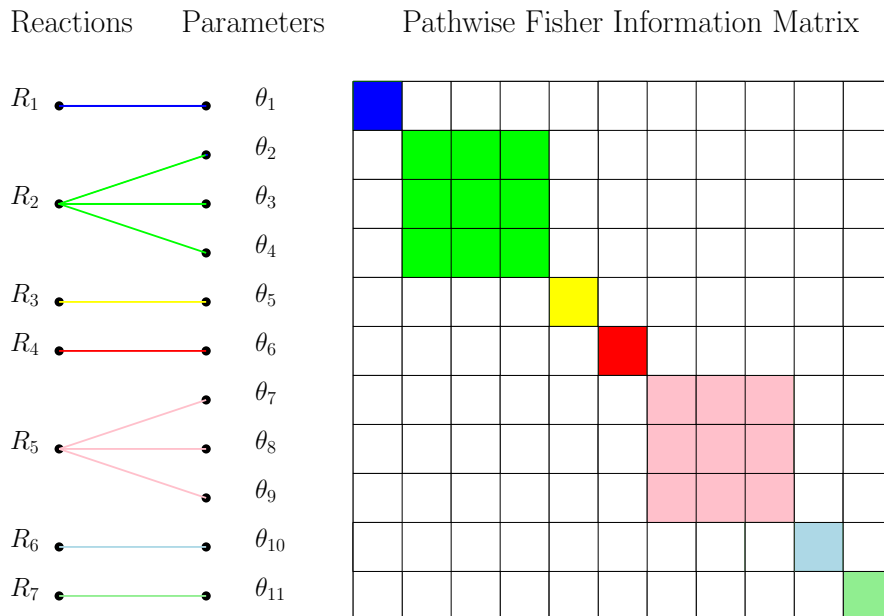


Figure 3: The graph representation (left panel) of the dependencies between the reactions (left column) and the model parameters (right column) as well as the corresponding block-diagonal structure of the PFIM (right panel) of the crypt compartmental model (Smallbone and Corfe 2014). In chemical reaction networks, reactions typically depend only on a small subset of the parameter vector. This parametric dependence of the propensities can be directly reflected on the PFIM. By grouping the reactions into subsets in such a way that each subset contains the minimum number of reactions having common parameters, the PFIM becomes a block-diagonal matrix upon rearrangement of the parameter vector (Pantazis and Katsoulakis 2013). In this example, the grouping of the 11 parameters is carried out by noting the connected parts of the graph and that the largest dimension of the blocks is 3.

The PFIM (Pantazis and Katsoulakis 2013),  $F_H(Q^\theta)$ , 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 PFIM gathers additional information by taking into consideration the dynamical aspects of the process as well. Most importantly, an explicit formula for the PFIM is given by

$$F_H(Q^\theta) = E_{\mu^\theta} \left[ \sum_j a_j^\theta(\mathbf{x}) \nabla_\theta \log a_j^\theta(\mathbf{x}) \nabla_\theta \log a_j^\theta(\mathbf{x})^\top \right], \quad j = 1, \dots, M. \quad (6)$$

This explicit formula reveals that for a typical reaction network the PFIM has block-diagonal structure upon rearrangement of the parameter vector. Moreover, PFIM is a steady state average which only depends on the propensity functions  $a_j^\theta(x)$ . Figure 3 describes the block-diagonal structure for the dynamics of the crypt compartmental model (Smallbone and Corfe 2014). Concentrating on the PFIM estimation, since the stationary distribution,  $\mu^\theta$ , is usually



unknown, an unbiased estimator for the PFIM (Pantazis and Katsoulakis 2013) 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 may 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 using the chain rule, we obtain the logarithmically-scaled PFIM (Pantazis and Katsoulakis 2013),  $F_H(Q^{\log \theta})$

$$\left(F_H(Q^{\log \theta})\right)_{k,l} = \theta_k \theta_l \left(F_H(Q^\theta)\right)_{k,l}, \quad k, l = 1, \dots, K. \quad (7)$$

Since the stationary distribution,  $\mu^\theta$ , is often unknown, an unbiased estimator for FIM is given as ergodic averages (Pantazis and Katsoulakis 2013),

$$\bar{F}_H^{(n)} = \frac{1}{T} \sum_i \Delta t_i \sum_j a_j^\theta(\mathbf{x}_i) \nabla_\theta \log a_j^\theta(\mathbf{x}_i) \nabla_\theta \log a_j^\theta(\mathbf{x}_i)^\top, \quad i = 0, \dots, n-1, \quad j = 1, \dots, M, \quad (8)$$

where  $\Delta t_i$  is an exponential random variable with parameter given by the total rate,  $a_0^\theta(\mathbf{x}_i) = \sum_{j=1}^M a_j^\theta(\mathbf{x}_i)$ , with  $T = \sum_{i=1}^n \Delta t_i$  is the total simulation time. The sequence  $\{\mathbf{x}_i\}_{i=0}^n$  forms a Markov chain with transition probabilities from state  $\mathbf{x}_i$  to state  $\mathbf{x}_{i+1}$  determined by the ratio  $\frac{a_j^\theta(\mathbf{x}_i)}{a_0^\theta(\mathbf{x}_i)}$ .

Finally, we would like to address how the PFIM can be utilized for fast parameter screening. We refer to Arampatzis *et al.* (2015) for a detailed discussion. Proceeding, let us denote by  $\mathbf{G}(\cdot) = [G_1(\cdot), \dots, G_L(\cdot)]^\top$  the vector with  $L$  state-dependent observable functions,  $G_l : \mathbf{X} \rightarrow \mathbb{R}$ ,  $l = 1, \dots, L$ . One typical option for the observable function is the time-average of a function which is defined in a general setting as

$$G_l(\{\mathbf{X}_t\}) = \frac{1}{T} \int_t g_l(\mathbf{X}_t) dt, \quad (9)$$

where the most common observable is the population of the  $l$ th species, i.e., the projection of the state vector to the  $l$ th direction ( $g_l(\mathbf{x}) = x_l$ ). The sensitivity matrix of the observable functions,  $S \in \mathbb{R}^{K \times L}$ , is defined by

$$S_{k,l} = \frac{\partial}{\partial \theta_k} E_{Q^\theta} [G_l(\{\mathbf{X}_t\})], \quad k = 1, \dots, K, \quad l = 1, \dots, L. \quad (10)$$

The element,  $S_{k,l}$ , is the sensitivity index (SI) of the  $l$ th observable with respect to the  $k$ th parameter. For the stationary regime and time-averaged observables, a sensitivity bound for the absolute value of the SIs can be derived either from rearranging the Cramer-Rao inequality (Arampatzis *et al.* 2015) or from variational formulas (Dupuis, Katsoulakis, Pantazis, and Plecháč 2016). Particularly, the sensitivity bound reads

$$|S_{k,l}| \leq C_l \sqrt{F_H(Q^\theta)_{k,k}}, \quad k = 1, \dots, K, \quad l = 1, \dots, L, \quad (11)$$

where  $F_H(Q^\theta)$  is the PFIM while  $C_l$  is a constant related only to the  $l$ th observable function,  $G_l$ . Apparently, this inequality can be utilized as a screening tool to discard the most insensitive SIs since diagonal PFIM elements with small values imply low SIs (Arampatzis *et al.* 2015).

Classical sensitivity analysis approaches scale poorly in the presence of high dimensions and many parameters. Take the mean field approximations (Equation 4) for example, where we consider a system of  $N$  ordinary differential equations,

$$\dot{y} = f(y; \theta), \quad y(0) = y_0 \in \mathbb{R}^N, \quad (12)$$

when performing a local sensitivity analysis on the model parameters  $\theta \in \mathbb{R}^K$ , we define sensitivity indices measuring the impact of small parameter perturbations on the model by

$$s_k = \frac{\partial y}{\partial \theta_k}. \quad (13)$$

Then, a new system of ordinary differential equations is derived and augmented with the original system (Equation 12):

$$\dot{s}_k = \frac{\partial f}{\partial y} s_k + \frac{\partial f}{\partial \theta_k}, \quad k = 1, \dots, K. \quad (14)$$

Therefore, in order to study the local sensitivity of the system of ordinary differential equations we need to solve  $K \times N$  additional equations, where both  $K, N \gg 1$ . As a result, to estimate parameter sensitivity, the classical sensitivity analysis approaches have to solve an adjunct system of  $(K + 1) \times N$  differential equations (i.e., the ordinary differential equation system (Equation 12) that governs the dynamics of the reaction network augmented by the ordinary differential equations for the derivatives with respect to the parameters (Equation 14)). In comparison, the proposed pathwise sensitivity analysis approach can estimate parameter sensitivity in a more efficient manner as it only needs to solve  $N$  differential equations (Equation 12) and calculate the unbiased estimator of PFIM (Equation 8).

### 3. Structure of the package

The **ISAP** package is implemented as a set of MATLAB functions. By taking the user-defined biochemical reaction system, the package extracts a stoichiometry matrix which consists of the coefficients of the chemical species in the reactions and the propensity functions which determine the reaction rates. With the stoichiometry matrix and the propensity functions, realizations of time series in the given time period can be generated according to different simulation algorithms (SSA, CLE and MFA) by users' choice.

This package provides both, simulators for various processes as well as tools for sensitivity, robustness and identifiability of model parameters (see the schematic in Figure 4). Our package using time-series either locally generated or externally provided by another simulator is able to efficiently compute the unbiased estimator of both RER and PFIM (Pantazis and Katsoulakis 2013). Furthermore, this package offers functions to extract sensitivity coefficients from the spectral analysis of the PFIM and parameter sensitivities can be ranked accordingly. In our package, simultaneous perturbations can be added to any subset of model parameters to test the robustness of the system and the analysis of eigenvalues of the PFIM provides information on the identifiability of model parameters (Pantazis *et al.* 2013).



2. Select the curated data (e.g., ‘SBML L2 V4 (curated)’ from ‘download SBML’) and download the .xml file.
3. Use the `sbml2txt.m` function to create the three input files.

Note: The current version of the **ISAP** package is able to handle the SBML file examples with the following features, otherwise users 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.
- Users 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 the .xml file and rewrite if necessary. Load the .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 generates 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 ( $X1$ ,  $X2$  and  $X3$ ) and 3 reaction constants ( $k1$ ,  $k2$  and  $k3$ ).

- $X1 + X2 \rightarrow X3$  with propensity function  $k1 \times X1 \times X2$ ;
- $X3 \rightarrow X1 + X2$  with propensity function  $k2 \times X3$ ;
- $X2 \rightarrow NULL$  with propensity function  $k3 \times X2$ .

Input files:

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

- $X1 + X2 = X3$  ' $k1 * X1 * X2$ ';
- $X3 = X1 + X2$  ' $k2 * X3$ ';
- $X2 = NULL$  ' $k3 * X2$ '.

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

- $NULL = 0$ ;
- $X1 = 100$ ;
- $X2 = 100$ ;
- $X3 = 100$ .

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

- $k1 = 62849$ ;
- $k2 = 0.70599$ ;
- $k3 = 1$ .

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 this section).

## 5. Stochastic simulation

Our package provides three different ways to conduct stochastic simulation.

### 5.1. Stochastic simulation algorithm

A stochastic simulation algorithm generates an exact, discrete, single event, stochastic process.

#### *Relevant functions*

- reaction\_ssa(data)

**Description:** Simulation for well-mixed reaction systems with a 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

A chemical Langevin equation generates an approximated, continuous, stochastic differential trajectories.



*Relevant functions*

- `reaction_cle(data)`

**Description:** Simulation for well-mixed reaction systems with a 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**

A 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 an 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.

## 6. Pathwise sensitivity analysis

Package **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 an input time series, package **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 an input time series, package **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 an input time series, package **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.

## 7. Example problems

Several examples are included in the examples directory:

**BIO520:** Crypt compartmental model (3 species, 7 reactions, 11 parameters).

**BIO427:** Model of epidermal growth factor receptor and type 1 insulin-like growth factor pathways in non-small cell lung cancer (NSCLC) (21 species, 22 reactions, 54 parameters).

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

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

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

**p53:** Tumor protein p53 gene network (3 species, 5 reactions, 7 parameters).

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

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

We shall illustrate the usefulness and flexibility of the **ISAP** package by analyzing two biochemical reaction networks. All the input files of built-in examples are provided in the example folder and .xml folder for replication. Moreover, we demonstrate the procedures to download and transform examples from the BioModels Database in the first example.

### 7.1. Crypt compartmental model

It is widely believed that colonic crypts, millions of invaginations in the lining of the colon, are the places where colorectal cancer first appears. The crypt compartmental approach which accounts for populations of stem cells, differentiated cells, and transit cells helps us to model

the development and early progression of colorectal cancer. Here we study a reaction network model for the dynamics of the crypt compartmental model (Smallbone and Corfe 2014) which consists of 3 cell species with interactions through 7 reactions with 11 parameters. First we will take the 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 package **ISAP**. In general, it is easy to modify the test functions to accommodate a user defined reaction network.

1. Create an example folder in the following directory: '`\examples\BI0520\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/BI0520/input_files/BI0520'`.
6. Use the `sbml2txt(file, outputfile)` function to create three input files.
7. Run `example_methods_BI0520` and `example_isap_BI0520` to do simulation and perform sensitivity analysis respectively.

The main components of the test functions are listed as below:

**Step 1:** Set the directory of the reaction network.

```
>> dname = 'examples/BI0520/';
>> fname = [dname 'input_files/BI0520'];
```

**Step 2:** Create auxiliary files, if necessary.

```
>> exCM = dir([fname '_connectivity.txt']);
>> if isempty(exCM)
    rxn2csm([fname '.txt']);
end
```

**Step 3:** Create the data structure.

```
>> data = initialization_log(fname); % for log-scale sensitivity analysis
>> data.T = 100; % final time
>> data.NoJ = 1000000; % maximum number of jumps
>> data.NoS = 5; % number of repeated simulations
>> dt = 0.1; % sampling frequency in time
>> data.ti = dt:dt:data.T; % sampling time instants
>> data.tau = 0.05; % time-step for CLE
```

**Step 4:** Simulate the reaction network and save the species time series.

```
>> sim_rxn(data, [dname 'data/'], 'SSA');
>> sim_rxn(data, [dname 'data/'], 'MFA');
>> sim_rxn(data, [dname 'data/'], 'CLE');
```

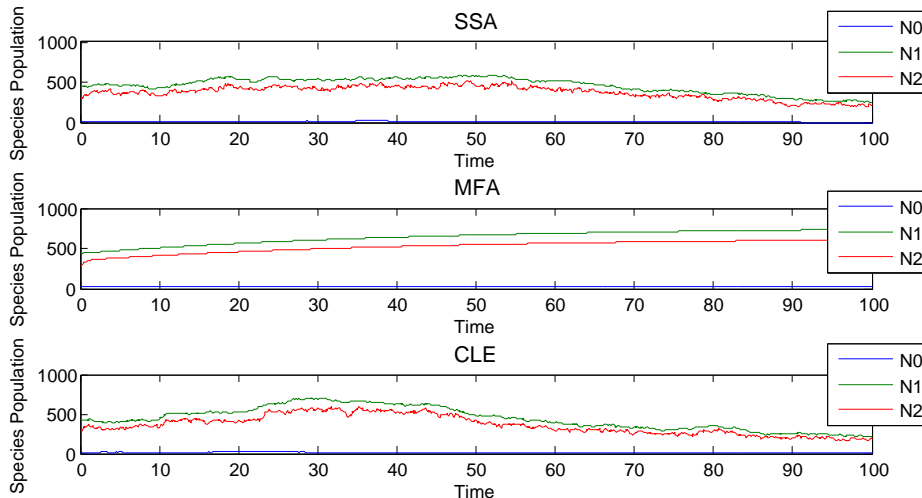


Figure 5: Time series from colon crypt cycle model version 0 (<http://www.ebi.ac.uk/biomodels-main/BIOMD0000000520>) generated from SSA, MFA and CLE. In this example, stochasticity plays a very important role when low values of the species occur as the MFA of the stochastic system can be valid only when there are large numbers of molecules of each species (Higham 2008).

Algorithm	SSA	MFA	CLE
Time (s)	71.24	0.22	3.57

Table 2: Table of computation time with SSA, MFA and CLE.

**Step 5:** Plot the species of the reaction network.

```
>> plot_species(data, [dname 'data/X_SSA_1.mat'], [3, 1, 1], 6);
>> title('SSA');
>> plot_species(data, [dname 'data/X_MFA_1.mat'], [3, 1, 2], 6);
>> title('MFA');
>> plot_species(data, [dname 'data/X_CLE_1.mat'], [3, 1, 3], 6);
>> title('CLE');
```

The time series of the reaction network, see Figure 5, can be realized with SSA, MFA or CLE. A comparison of the simulation time with different algorithms is listed in Table 2. The SSA method requires a great amount of simulation time (71.24 seconds) to provide exact numerical realizations of the system. Assuming that all reactions fire simultaneously on a fixed time interval, the CLE method approximates the SSA method and accelerates the simulation time to 3.57 seconds. Being regarded as the CLE method without the noise terms, the MFA method provides the fastest but least accurate results in 0.22 seconds.

For the purposes of pathwise RER and PFIM calculations, we use the simulation data from the CLE method.

**Step 6:** Create 10-percent perturbation vectors.

```
>> eps0 = 0.1;
```

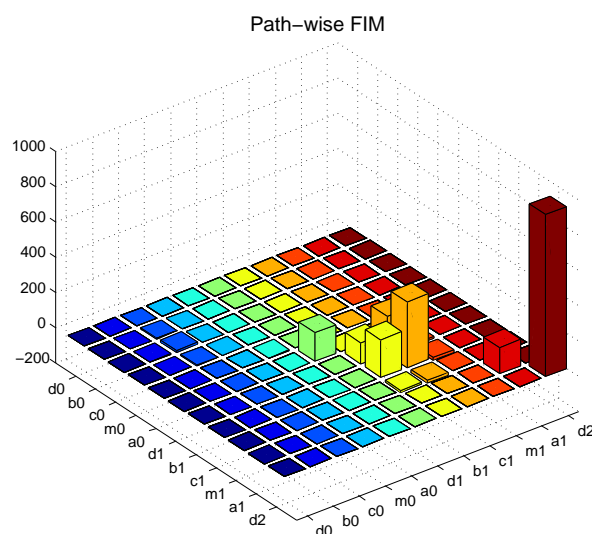


Figure 6: The proposed pathwise FIM (PFIM) from colon crypt cycle model version 0 (<http://www.ebi.ac.uk/biomodels-main/BIOMD0000000520>).

```
>> pert_vec = diag(eps0 * data.con);
>> pert_vec = [pert_vec; -pert_vec];
>> data.eps0 = eps0;
>> data.pert_vec = pert_vec;
```

**Step 7:** Compute PFIM and plot 3D PFIM.

```
>> PFIM = compute_PFIM_RER(data, dname);
>> figure;
>> plot_PFIM(PFIM, data);
```

The 3D barplot shows a block-diagonal structure of elements of the PFIM with the blocks denoting the reaction subsets with the same parametric dependence (Pantazis *et al.* 2013). The height of the bars corresponds to the values for the elements of PFIM. By visually inspecting Figure 4, the sensitive parameters are  $d_1$ ,  $b_1$ ,  $c_1$ ,  $a_1$  and  $d_2$ . In particular,  $d_2$  is the most sensitive parameter. The plot demonstrates that perturbations may affect the dynamics to those sensitive model parameters and the rest of the parameters are robust to variations.

### Relevant functions

- `example_methods_BI0520`

**Description:** Simulate the crypt compartmental model 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_BI0520`

**Description:** Simulate the crypt compartmental model reaction network with SSA, CLE or MFA and interpolate the simulated time series. Add 10 percent 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.

## 7.2. NSCLC model

The second example comes from a Systems Biology approach which aims to understand the molecular biology of the epidermal growth factor receptor (EGFR, also known as ErbB1/HER1) and type 1 insulin-like growth factor (IGF1R) pathways in non-small cell lung cancer (NSCLC) (Bianconi, Baldelli, Ludovini, Crinò, Flacco, and Valigi 2012). This reaction network consists of 21 species which can interact through 22 reactions with 54 parameters.

**Step 1:** Set the directory of the reaction network.

```
dname = 'examples/BIO427/';
fname = [dname 'input_files/BIO427'];
```

**Step 2:** Create auxiliary files, if necessary.

```
>> exCM = dir([fname '_connectivity.txt']);
>> if isempty(exCM)
    rxn2csm([fname '.txt']);
end
```

**Step 3:** Create the data structure.

```
>> data = initialization_log(fname); % for log-scale sensitivity analysis
>> data.T = 100; % final time
>> data.NoJ = 1000000; % maximum number of jumps
>> data.NoS = 5; % number of repeated simulations
>> dt = 0.1; % sampling frequency in time
>> data.ti = dt:dt:data.T; % sampling time instants
>> data.tau = 0.05; % time-step for CLE
```

We realize the processes with the mean field approximation.

**Step 4:** Simulate the reaction network and save the species time series.

```
>> sim_rxn(data, [dname 'data/'], 'MFA');
```

Solving the derived system of ODEs with MATLAB routine `ode15s`, we compute the PFIM at the steady state regime. The associated PFIM has dimension  $54 \times 54$ . The block diagonal structure of the PFIM (Figures 7 and 8) indicates sparsity which can be used to reduce the computational costs.



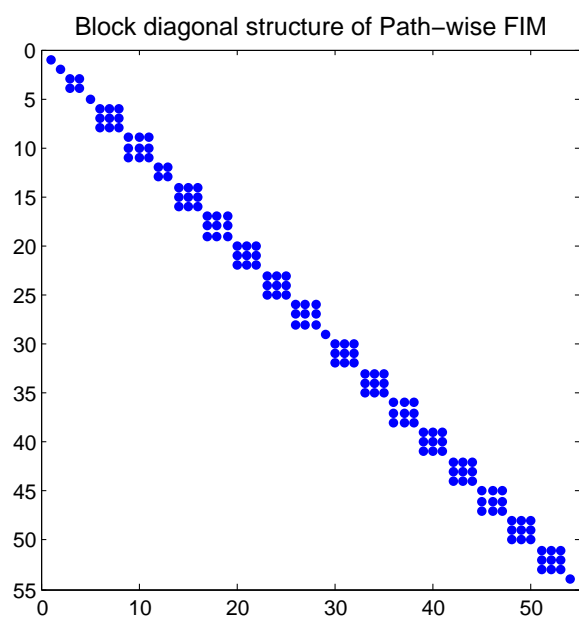


Figure 7: Block diagonal structure of PFIM of the NSCLC model (<http://www.ebi.ac.uk/biomodels-main/BIOMD000000427>). White space represents the zero elements of the PFIM which indicates sparsity in PFIM. Only 5 percent of the elements of PFIM are nonzero.

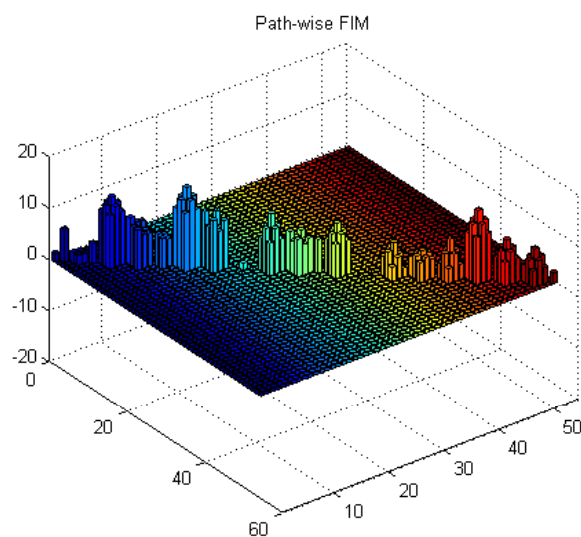


Figure 8: Block diagonal structure of PFIM in log scale of the NSCLC model. Parameters with higher values are more sensitive to perturbations. About 4.5 percent of elements of the PFIM are greater than zero in log scale.

**Step 5:** Create 10-percent perturbation vectors.

```
>> eps0 = 0.1;
>> pert_vec = diag(eps0 * data.con);
>> pert_vec = [pert_vec; -pert_vec];
>> data.eps0 = eps0;
>> data.pert_vec = pert_vec;
```

**Step 6:** Compute PFIM and plot 2D PFIM.

```
>> PFIM = compute_PFIM_RER(data, dname);
>> figure;
>> spy(PFIM(:,:,1));
```

**Step 7:** Compute PFIM and plot 3D log PFIM.

```
>> PFIM = compute_PFIM_RER(data, dname);
>> lPFIM = PFIM;
>> idx = PFIM ~= 0;
>> lPFIM(idx) = log(abs(PFIM(idx)));
>> figure;
>> plot_PFIM(lPFIM, data);
```

## 8. Summary of functions

We summarize the implemented functions showing their hierarchical dependencies.

### 8.1. Names of the m-functions and their hierarchy

1. `example_methods_{BI0520|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_{BI0520|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`

## 9. Conclusion

Package **ISAP** is a novel and extendable MATLAB package for simulation and computationally-efficient sensitivity analysis of complex stochastic reaction networks. The wide range of capabilities and the detailed documentation make package **ISAP** ideal for helping the modeler to create user-defined models and generate simulation data. By employing the RER, the implemented sensitivity analysis approach quantifies the information loss per unit time along different parameter perturbations between time series distributions. An associated PFIM addresses a gradient-free approach for parametric sensitivity analysis. Compared to other sensitivity software packages, our package provides a significant advantage in reducing the computational cost especially for complex stochastic networks with a high-dimensional parameter space and can be used as a screening tool for insensitive parameters.

## Acknowledgment

We would like to thank Georgios Arampatzis for his assistance in developing an early version of the SSA code and Jie Wang for testing the software and providing feedback.

The work was supported in part by the Office of Advanced Scientific Computing Research, U.S. Department of Energy under Contract No. DE-SC0010723. Also, the research of MAK was partially supported by the Defense Advanced Research Projects Agency (DARPA) EQUiPS program under the grant W911NF1520122.

## References

- Anderson DF (2012). “An Efficient Finite Difference Method for Parameter Sensitivities of Continuous Time Markov Chains.” *SIAM Journal on Numerical Analysis*, **50**(5), 2237–2258. doi:10.1137/110849079.
- Arampatzis G, Katsoulakis MA, Pantazis Y (2015). “Accelerated Sensitivity Analysis in High-Dimensional Stochastic Reaction Networks.” *PloS ONE*, **10**(7), e0130825. doi:10.1371/journal.pone.0130825.
- Bianconi F, Baldelli E, Ludovini V, Crinò L, Flacco A, Valigi P (2012). “Computational Model of EGFR and IGF1R Pathways in Lung Cancer: A Systems Biology Approach for Translational Oncology.” *Biotechnology Advances*, **30**(1), 142–153. doi:10.1016/j.biotechadv.2011.05.010.
- Dupuis P, Katsoulakis MA, Pantazis Y, Plecháč P (2016). “Path-Space Information Bounds for Uncertainty Quantification and Sensitivity Analysis of Stochastic Dynamics.” *SIAM/ASA Journal on Uncertainty Quantification*, **4**(1), 80–111. doi:10.1137/15m1025645.
- Feehery WF, Tolsma JE, Barton PI (1997). “Efficient Sensitivity Analysis of Large-Scale Differential-Algebraic Systems.” *Applied Numerical Mathematics*, **25**(1), 41–54. doi:10.1016/s0168-9274(97)00050-0.
- Gillespie DT (2007). “Stochastic Simulation of Chemical Kinetics.” *Annual Review of Physical Chemistry*, **58**, 35–55. doi:10.1146/annurev.physchem.58.032806.104637.
- Goutsias J, Jenkinson G (2013). “Markovian Dynamics on Complex Reaction Networks.” *Physics Reports*, **529**(2), 199–264. doi:10.1016/j.physrep.2013.03.004.
- Hengl S, Kreutz C, Timmer J, Maiwald T (2007). “Data-Based Identifiability Analysis of Non-Linear Dynamical Models.” *Bioinformatics*, **23**(19), 2612–2618. doi:10.1093/bioinformatics/btm382.
- Higham DJ (2008). “Modeling and Simulating Chemical Reactions.” *SIAM Review*, **50**(2), 347–368. doi:10.1137/060666457.
- Hindmarsh AC, Brown PN, Grant KE, Lee SL, Serban R, Shumaker DE, Woodward CS (2005). “SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers.” *ACM Transactions on Mathematical Software*, **31**(3), 363–396. doi:10.1145/1089014.1089020.
- Hoops S, Sahle S, Gauges R, Lee C, Pahle J, Simus N, Singhal M, Xu L, Mendes P, Kummer U (2006). “COPASI – A COmplex PATHway Simulator.” *Bioinformatics*, **22**(24), 3067–3074. doi:10.1093/bioinformatics/btl485.

- Komorowski M, Costa MJ, Rand DA, Stumpf MPH (2011). “Sensitivity, Robustness, and Identifiability in Stochastic Chemical Kinetics Models.” *Proceedings of the National Academy of Sciences of the United States of America*, **108**(21), 8645–8650. doi:[10.1073/pnas.1015814108](https://doi.org/10.1073/pnas.1015814108).
- Komorowski M, Žurauskiene J, Stumpf MPH (2012). “**StochSens** – MATLAB Package for Sensitivity Analysis of Stochastic Chemical Systems.” *Bioinformatics*, **28**(5), 731–733. doi:[10.1093/bioinformatics/btr714](https://doi.org/10.1093/bioinformatics/btr714).
- Le Novère N, Bornstein B, Broicher A, Courtot M, Donizelli M, Dharuri H, Li L, Sauro H, Schilstra M, Shapiro B, others (2006). “BioModels Database: A Free, Centralized Database of Curated, Published, Quantitative Kinetic Models of Biochemical and Cellular Systems.” *Nucleic acids research*, **34**(Suppl 1), D689–D691. doi:[10.1093/nar/gkj092](https://doi.org/10.1093/nar/gkj092).
- Maiwald T, Timmer J (2008). “Dynamical Modeling and Multi-Experiment Fitting with Pot-  
tersWheel.” *Bioinformatics*, **24**(18), 2037–2043. doi:[10.1093/bioinformatics/btn350](https://doi.org/10.1093/bioinformatics/btn350).
- Maly T, Petzold LR (1996). “Numerical Methods and Software for Sensitivity Analysis of Differential-Algebraic Systems.” *Applied Numerical Mathematics*, **20**(1–2), 57–79. doi:[10.1016/0168-9274\(95\)00117-4](https://doi.org/10.1016/0168-9274(95)00117-4).
- McGill JA, Ogunnaike BA, Vlachos DG (2012). “Efficient Gradient Estimation Using Finite Differencing and Likelihood Ratios for Kinetic Monte Carlo Simulations.” *Journal of Computational Physics*, **231**(21), 7170–7186. doi:[10.1016/j.jcp.2012.06.037](https://doi.org/10.1016/j.jcp.2012.06.037).
- Pantazis Y, Katsoulakis MA (2013). “A Relative Entropy Rate Method for Path Space Sensitivity Analysis of Stationary Complex Stochastic Dynamics.” *The Journal of Chemical Physics*, **138**(5), 054115. doi:[10.1063/1.4789612](https://doi.org/10.1063/1.4789612).
- Pantazis Y, Katsoulakis MA, Vlachos DG (2013). “Parametric Sensitivity Analysis for Biochemical Reaction Networks Based on Pathwise Information Theory.” *BMC Bioinformatics*, **14**(1), 311. doi:[10.1186/1471-2105-14-311](https://doi.org/10.1186/1471-2105-14-311).
- Plyasunov S, Arkin AP (2007). “Efficient Stochastic Sensitivity Analysis of Discrete Event Systems.” *Journal of Computational Physics*, **221**(2), 724–738. doi:[10.1016/j.jcp.2006.06.047](https://doi.org/10.1016/j.jcp.2006.06.047).
- Rathinam M, Sheppard PW, Khammash M (2010). “Efficient Computation of Parameter Sensitivities of Discrete Stochastic Chemical Reaction Networks.” *The Journal of Chemical Physics*, **132**(3), 034103. doi:[10.1063/1.3280166](https://doi.org/10.1063/1.3280166).
- Raue A, Kreutz C, Maiwald T, Bachmann J, Schilling M, Klingmüller U, Timmer J (2009). “Structural and Practical Identifiability Analysis of Partially Observed Dynamical Models by Exploiting the Profile Likelihood.” *Bioinformatics*, **25**(15), 1923–1929. doi:[10.1093/bioinformatics/btp358](https://doi.org/10.1093/bioinformatics/btp358).
- Rodriguez-Fernandez M, Banga JR (2010). “**SensSB**: A Software Toolbox for the Development and Sensitivity Analysis of Systems Biology Models.” *Bioinformatics*, **26**(13), 1675–1676. doi:[10.1093/bioinformatics/btq242](https://doi.org/10.1093/bioinformatics/btq242).

- Sheppard PW, Rathinam M, Khammash M (2012). “A Pathwise Derivative Approach to the Computation of Parameter Sensitivities in Discrete Stochastic Chemical Systems.” *The Journal of Chemical Physics*, **136**(3), 034115. doi:10.1063/1.3677230.
- Sheppard PW, Rathinam M, Khammash M (2013). “SPSens: A Software Package for Stochastic Parameter Sensitivity Analysis of Biochemical Reaction Networks.” *Bioinformatics*, **29**(1), 140–142. doi:10.1093/bioinformatics/bts642.
- Smallbone K, Corfe BM (2014). “A Mathematical Model of the Colon Crypt Capturing Compositional Dynamic Interactions between Cell Types.” *International Journal of Experimental Pathology*, **95**(1), 1–7. doi:10.1111/iep.12062.
- StochSS Org (2013). *StochSS: Stochastic Simulation Service*. URL <http://www.stochss.org/>.
- The MathWorks Inc (2017). *MATLAB – The Language of Technical Computing, Version R2017b*. Natick. URL <http://www.mathworks.com/products/matlab/>.
- Vellela M, Qian H (2009). “Stochastic Dynamics and Non-Equilibrium Thermodynamics of a Bistable Chemical System: The Schlögl Model Revisited.” *Journal of the Royal Society Interface*, **6**(39), 925–940. doi:10.1098/rsif.2008.0476.

### Affiliation:

Weilong Hu, Markos A. Katsoulakis  
 Department of Mathematics and Statistics  
 University of Massachusetts-Amherst  
 MA, United States of America  
 E-mail: [weilong@math.umass.edu](mailto:weilong@math.umass.edu), [markos@math.umass.edu](mailto:markos@math.umass.edu)  
 URL: <https://www.math.umass.edu/directory/graduate-students/weilong-hu>  
<https://people.math.umass.edu/~markos/>

Yannis Pantazis  
 Department of Mathematics and Statistics  
 University of Massachusetts-Amherst  
 MA, United States of America  
*Current affiliation:*  
 Institute of Applied and Computational Mathematics  
 Foundation for Research and Technology  
 Hellas, Heraklion, Greece  
 E-mail: [pantazis@iacm.forth.gr](mailto:pantazis@iacm.forth.gr)

---

*Journal of Statistical Software*  
 published by the Foundation for Open Access Statistics  
 June 2018, Volume 85, Issue 3  
 doi:10.18637/jss.v085.i03

<http://www.jstatsoft.org/>  
<http://www.foastat.org/>  
 Submitted: 2015-04-03  
 Accepted: 2017-06-22

---