Proceedings of the 2012 IEEE 12th International Conference on Bioinformatics & Bioengineering (BIBE), Larnaca, Cyprus, 11-13 November 2012

A distributed software package for global sensitivity analysis of biological models

Quang Tung Thai, Seokjong Yu and Yongseong Cho
Department of Supercomputing
Korea Institute of Science and Technology Information (KISTI)
Daejeon, Korea South
Email: {tqtung, codegen, yscho}@kisti.re.kr

Abstract—**Mathematical modeling is a vital tool for studying biological systems. Due to the system complexity and technical challenges in molecular level measurement, it is commonly the case that a large number of model parameters have uncertain values. Analyzing model dynamics from a single estimated parameter set is insufficient and liable for misleading results. Global sensitivity analysis (GSA) has been recommended as a must-have step in the process of developing reliable models. However, the technique comes at high computation costs as it is based on Monte Carlo simulation which requires a large number of model evaluations and manipulating on massive data for sensitivity estimation. In this work, we develop a software package for global sensitivity estimation of biological models. The software is deployed on KISTI high performance computing (HPC) cluster environment to provide web-service to system biology modelers.**

Keywords-component; system biology; mathematical modeling; sensitivity analysis; parallel computing

I. INTRODUCTION

Systems biology is an interdisciplinary research field that focuses on understanding how various biological components interact to give rise to biological phenomenal [1]. In order to comprehend biological system working mechanism, mathematical modeling is a vital tool. Not only providing the systematically understandings, mathematical models can help perform experiments which have been conventionally time consuming, expensive or even infeasible. Consequently, new hypothesis can be formulated through such *in silico* experiments.

Due to the nature of biological complexity, usually we have knowledge of the qualitative 'structure' of a model but little knowledge of model parameters is available. Parameter estimation is still a major challenge and remains the bottleneck in biological modeling as the number of parameters is usually large while the available experiment data is limited. In general, it is possible to fit a model to any experimental data, i.e. to find a set of parameter values whose model simulation outcome can resembles such data as expected. Models with globally estimated parameters although can provide biologically plausible predictions, they are usually vulnerable to instrumental or unethical use. Therefore, it has been recommended that model calibration and analysis must be performed in a rigorous systematic framework where uncertainty in model input must be taken into account by applying sensitivity analysis.

Sensitivity analysis (SA) is the assessment of how a model output is dependent of its parameter variation. Not until recently, it is often defined as the local measure of the effect of a given input on a given output, i.e. system derivatives such as $S_i = {}^{\partial Y}\!/_{\partial X_i}$, where Y is the output of interest and X_i is an input factor. This local approach has the attraction of being very efficient in computation time, yet it is only informative at the nominal point where the model is fitted. In the presence of uncertainty inputs which is often the case of biological models, the assessment therefore should make use of methods based on exploring the input space, considering that results obtained comprehensively from a handful of points thrown into the space is much more informative and robust than the derivative estimated at a single point. This approach, where the variation in model input is performed in a global fashion, i.e. all input factors are perturbed for whole input space exploration, is called global sensitivity analysis (GSA). Although the goals of GSA is to measure the influence of variance in the model input to the model output, its ultimate usefulness lies on the ability to identify parameters, components that capture essential characteristics of a model. Such analysis is particularly useful for complex biological networks whose output behavior is dictated by a large number of input parameters. Its application includes, to name a few, model reduction, experiment design and model validation [2]. The technique has been an essential tool which is applied regularly in the process of developing complex biological models [3][4][5][6].

In general, GSA is based on Monte Carlo methodology which requires repeated batch jobs of a large number of simulations to obtain statistically valid results. This may not be a problem for relatively simple biological models, but as the complexity of the system model increases this will increase the computational demand. The need for high performance computing becomes essential.

Despite increasing research utilizing GSA techniques, there are few current available software tools for sensitivity analysis of biological models. Simlab [7] is a general sensitivity analysis package that implements most recent GSA techniques but it is lacking of a simulation engine for biological models. SBML-SAT [8] is developed specifically for biological system modeling as it provides interface to model simulation as well as SBML language, a de-factor standard for model representation in system biology. System biology toolbox [9], a package for mathematical modeling of

978-1-4673-4358-9/12/$31.00 ©2012 IEEE

biological systems, also provides the GSA features. However those software packages are developed for desktop-based systems without the utilization of multi-cores power. They are suitable for models with small number of parameters. In recent applications of GSA on large models [10], intensive model evaluations are distributed on cluster nodes then massive simulation results are collected to estimate parameter sensitivity. The whole process is manually handled and computationally inefficient. In this work, we aim to develop a software package that can utilize high performance computing systems for the task of parameter sensitivity estimation and to deploy the package in our HPC system to provide web-service to system biologists.

Remaining of the article is organized as follows: Section 2 provides background of mathematical modeling and global sensitivity analysis. In section 3, the architecture of the software and its parallelization are described. Computational performance evaluation and software deployment is explained in section 4.

II. Background

A. Mathematical models of biological systems

Computational models use mathematical rules to describe underlying dynamics of interactions among system components. In this work we target models that use ordinary differential equations (ODE) due to their popularity in system biology field. Set of ODEs that governs the temporal evolution of molecular entities is depicted as follows:

$$\frac{dX}{dt} = f(X,\theta) \qquad (1)$$
$$X(t=0) = X_0$$

Here, X is a set of species such as proteins, enzymes; X_0 is their concentrations at initial state; θ is a vector of kinetic parameters that represent rates of reactions in the model. Given parameter values and initial conditions, the equations can be solved with numerical methods to produce time course evolution of species.

B. Global sensitivity analysis

The term sensitivity analysis has a variety of meanings in different disciplines. A general definition by Saltalli [2] states that “sensitivity analysis is the study of how the variation in the output of a model can be apportioned, qualitatively or quantitatively, to difference sources of variation, and how the given model depends upon the information fed to it”. To apply this definition to biological models, it is necessary to identify model input and output.

In the scope of system biology, set of parameters $P=(\theta, X_0)$ is considered as input factors which are sources of variation. The output is a singular variable which is usually a feature from time course data of an under-investigated species. In biological signaling pathways, this species is usually a downstream transcription factor protein which has important roles in controlling cellular processes. Let X_i be the interested species then the output is defined by:

$$Y = g(X_i(t)) \qquad (2)$$

where $X_i(t)$ represent the time course evolution of the species and g is a feature extraction function. Multiple features can be used such as values at specific time points, steady state of the signal, signaling duration. For simplicity, we consider output is a single variable. By combining (1) and (2) the relationship between output and inputs can be represented as a function:

$$Y = h(P) \qquad (3)$$

This function can be evaluated by solving the ODE system in (1) then evaluating the right-hand side of (2).

Towards this end, the model we are interested in is represented in (3). The goal of sensitivity analysis is to quantify the influence of uncertainty in input factors P to the uncertainty in output Y.

When input factors are relatively certain, partial derivative of the output function with respect to the input factors can be used as sensitivity measures. They can be computed numerically by performing multiple simulations varying input factors around a nominal value. This technique is called local sensitivity analysis as it measures the local impact of input factors on model output. For biological networks, input factors of their models will often be uncertain, as aforementioned, thus results from local SA might be not sufficient. It is necessary to assess model dynamics under global perturbation of inputs and GSA must be used.

In following sections, we introduce three main GSA methods that have been successfully utilized in system biology research field. After brief explanations, we describe their general underlying procedure to compute sensitivity measures.

1) Variance based methods

The main idea of the variance-based methods is to quantify the amount of variance that each input factor P_i contributes to the total variance of the output $V(Y)$. Assuming that the factor P_i is fixed at its true value p_i^*, the conditional variance of output given $P_i = p_i^*$ is $V(Y \mid P_i)$ obtained by taking the variance over all factors but P_i. However in most cases, the true value p_i^* is unknown, thus the average of this conditional variance for all possible p_i^* is used, i.e. $E[V(Y \mid P_i)]$. Following property of the variance, the total output variance can be decomposed as:

$$V(Y) = V(E[Y \mid P_i]) + E[V(Y \mid P_i)] \qquad (4)$$

The conditional variance $V_i = V(E[Y \mid P_i])$ is called the “main effect” and is used as an indicator of the importance of P_i on the variance of Y. Normalizing the main effect V_i by the total variance $V(Y)$ we obtain:

$$S_i = \frac{V_i}{V(Y)} \qquad (5)$$

This ratio S_i is named first order sensitivity index of factor P_i on output Y by Sobol [11].

Similarly, the joint effect of the pair (P_i, P_j) on Y is conditional variance $V(E[Y \mid P_i P_j])$ and is known as second-order effect. Subtracting the first order effect of each factor, the remaining is effect of interaction between the two factors:

$$V_{ij} = V(E[Y \mid P_i P_j]) - \\ V(E[Y \mid P_i]) - V(E[Y \mid P_j]) \tag{6}$$

The second-order sensitivity index of interaction between P_i and P_j on Y is defined as:

$$S_{ij} = \frac{V_{ij}}{V(Y)} \tag{7}$$

High-order effects can be defined in a similar fashion, for example the variance of the third order effect among three factors P_i, P_j, P_l would be:

$$V_{ijl} = V(E[Y \mid P_i P_j P_l]) - \\ V_{ij} - V_{il} - V_{jl} - V_{ijl} \tag{8}$$

$$S_{ijl} = \frac{V_{ijl}}{V(Y)}$$

The sum of all the order effects that a factor accounts for is called the 'total' effect. So for an input P_i, the total sensitivity index ST_i is defined as the sum of all indices relating to P_i (first and higher orders). For example, in a model with 3 input factors (k=3), the total sensitivity index for input factor P_1 would then be:

$$ST_1 = S_1 + S_{12} + S_{13} + S_{123} \tag{9}$$

In general, comparing to the first-order sensitivity index S_i the total sensitivity index ST_i is a more accurate measure for the effect of a factor on model output variance as it takes into account all interaction effects involving that factor.

In order to estimate the total sensitivity indices, all variances must be computed which in fact is to calculate integrals of function $Y=h(P)$ over multi-dimensional areas. These integrals can be computed by Monte Carlo simulation method. However, a 'brute force' approach to obtain the 'total' effect would needs as many as O(2^k) model evaluations which is not advisable for models with a relatively large number of input factors. Therefore, much of research has been to find approximation formulas of the variances and their efficient computation methods. In this work, we implement three methods to obtain sensitivity indices, these are Sobol's [11], Jansen's [12] and Saltalli's [13] methods. We refer to [2] for detail information of formula derivation and sampling techniques.

In general, variance-based methods work well with all models without any assumptions. However their main concern is the number of model evaluations which is usually large. The Sobol method would requires $N_0(2k+1)$ model evaluations, the Jensen's method needs kN_0 while the Saltalli method would requires $k(N_0+1)$ where k is the number of input factors and N_0 is a base sampling size. The value N_0 should be large enough for reliably computing one variance term, such as $V(Y)$. As a default rule in global sensitivity analysis, there is no such a standard sampling size, ones should increasing their sampling size and observe the sensitivity ranking of input factors. The sample size is considered large enough if the ranking order does not change for top sensitive factors.

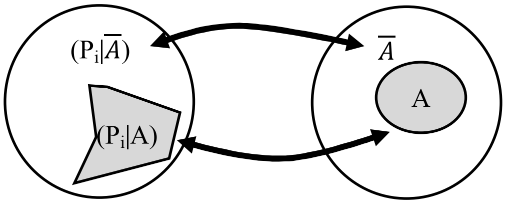

Figure 1 Mapping the well-behaved group and misbehaving group between input space and output space

2) Regional based method

Although variance based method can provide fine detailed information about which factors are important and how much they contribute to the variance of the output, they are extremely computational expensive. For some cases, the variance of output is not major interests but a particular portion of the distribution of the output is what we concern. For example, we are often interested in Y being above or below a given threshold. In this kind of setting, the question is which factors are mostly responsible for producing realizations of Y in the region of interest? To answer this, we tend to divided the realizations of output Y into two groups: 'good' or 'bad' and want to rank the input factors in term of their influence on this division. These lead to the regional based method which in fact demands less number of model evaluations.

In this approach, ones run a Monte Carlo experiment producing realizations of the output corresponding to different sampled points in the input factor space. Having done this, realizations of output Y is then classified into two groups: 'acceptable group' A, and 'unacceptable group' $\overline{\mathrm{A}}$. Realizations in group A is considered well-behaved comparing to an expected outcome of the model while the ones in group' $\overline{\mathrm{A}}$ is considered misbehaving. The classification can be as simple as comparing them to a threshold value.

Once the two groups are classified, the sensitivity of each input factor P_i on the output Y is determined as following:

- The classification in the output space is mapped to the input space, as shown on Figure 1. For each realization in A, its corresponding input sample is classified to group (P_i|A). Similarly, for each realization in $\overline{\mathrm{A}}$, its corresponding input sample is classified to group ($P_i|\overline{A}$). As a result the N input samples are classified into two groups: (P_i|A) and ($P_i|\overline{A}$)
- Empirical cumulative distributions $\mathrm{F}_1(P_i)$ and $\mathrm{F}_2(P_i)$ are estimated for the two sample groups (P_i|A) and ($P_i|\overline{A}$) respectively.
- A statistical test is performed to measure if the difference of the two distributions is significant or not. The Smirnov test can be used and it is defined as the greatest vertical distance between the two distributions

$$smirov(Y, P_i) = \\ Max_{P_i} \mid F_2(P_i) - F_2(P_i) \mid \tag{10}$$

The Smirnov score is considered as the sensitivity measure of factor P_i on the output.

3) Screening method

When dealing with complex models that have a large number of uncertain inputs and model evaluation is expensive, it is not feasible to use the variance methods and even the regional method is still computationally demanding. One may want to use only a few number of model evaluations to identify set of factors that have little contribution to the output uncertainty so that they can be neglected in consequent analysis, i.e. we can fix them at any given value over their range of uncertainty without reducing significantly the output variance. In that case, screening method is a well-suited tool.

General idea of screening method, proposed by Morris [14], is averaging local sensitivity measures throughout the input space. Instead of varying multiple input factors at the same time, only one factor is changed in small variations to measure its so-called "elementary effect" on output variance. For a given input factor, elementary effects are measured at multiple points in the input space then data analysis on their distribution would explain if the input factor is important.

In Morris' design, the input factor space is discretized and the possible input factor values will be restricted to be inside a regular k-dimensional q-level grid, where q is the number of "levels" of the design. The elementary effect of a given factor P_i at point $p=(p_1, p_2 .. p_k)$ of the grid is defined as a finite difference derivative approximation:

$$ee_i(p) = [h(p_1, p_2, .., p_{i-1}, p_i + \Delta_i, \\ p_{i+1}, ..., p_k) - h(p)] / \Delta_i \qquad (11)$$

where Δ_i is a predetermined multiple of $1/q$ which is the grid interval on dimension i^{th} (assuming $P_i \in [0,1]$). For each input factor, elementary effects are estimated at several random selected points in the input space. Let μ and σ are mean and standard deviation of the elementary effects. Intuitively, if all elementary effects are zero, then P_i does not have any effect on the output Y, also μ and σ are both zero. If all elementary effects have the same value, i.e σ is zero, then y is a linear function of P_i. In general, a high μ indicates a factor with an important overall influence on model output; a high σ indicates either a factor interacting with other factors or a factor whose effects are non-linear. These two quantities are considered as sensitivity measures for the input factors. It is worthy to note that they are only qualitative, i.e. the input factors are ranked in order of importance but they are not quantified on how much a given factor is more importance than others.

To estimate one elementary effect, as shown in equation (11), two model evaluations are needed. Assuming r elementary effects are required to produce reliable statistics μ and σ then the method will need $2rk$ model evaluations to estimate all sensitivity measures for k factors. However, Morris came up with a sampling design that can significantly reduce that number. In his design, a path consisting of $k+1$ points on the input grid is randomly generated in such a way that any two consecutive points are different in only one dimension, thus using k+1 model evaluations on that path can estimate k elementary effects for k factors. In the end, generating such r paths is enough to estimate rk elementary effects. The number of model evaluations needed is therefore only $r(k+1)$.

4) General framework for GSA

Although there are various techniques to estimate global sensitivity, they are based on a general framework of Monte Carlo simulation which following these steps:

- Step 1: For the defined model $Y = h(P)$ with k input factors $P = P_1, P_2, ..P_k$ and output Y, ranges and distribution are selected for each input factor. An expert in the system biological modeling must define these ranges and distributions. Usually when no prior knowledge is available, uniform distribution can be used.

- Step 2: Generates random values for the input factors based on their ranges and distribution. Output of this step is a matrix **P** of $N \times k$ size where the value of N is determined by users and it is dependent on model size and GSA technique used. Each row of the matrix represents a set of input factor values of the model.
- Step 3: The model is evaluated for each input vector to estimate the value of output Y. At the end of this step, a column vector **Y** of size N is estimated. For ODE models, model evaluation includes two steps. First, the set of ODEs (1) must be solved using numerical integration methods. Then value of output Y is computed with feature extraction function in equation (2).
- Step 4: The input matrix **P** is mapped to output vector **Y** using a regression model or a statistical model regarding to the used GSA technique. Based on that mapping, the influence of each input factor P_i on the output Y is quantified and that value is regarded as sensitivity measure.

The difference between GSA techniques lies on step 2 and step 4.

III. Software implementation

A. Software architecture

Developing mathematical models is a process of multiple steps where various computational tools are utilized such as simulation, parameter estimation, sensitivity analysis, bifurcation analysis. Our ultimate goal is to develop a software platform embracing a collection of such tools, each perform a particular task that can be stringed together in workflows. Depending on size of investigated models and computational complexity of analysis, some tasks can be performed on single machines while others need shifting to HPC systems providing web-service. The software platform therefore should be designed for deploying on both environments.

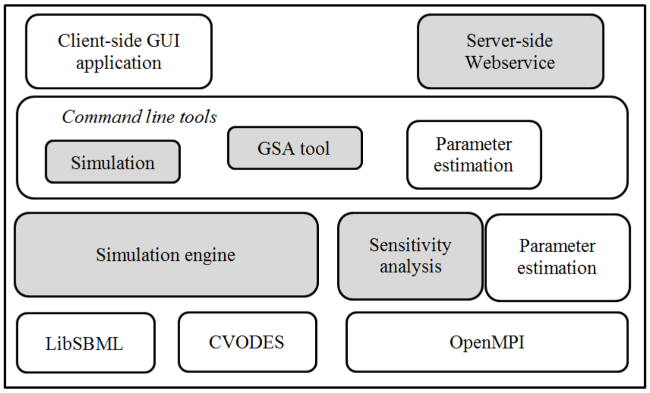

Figure 2: The software platform for biological system modeling

The GSA package, being a part of, thus shares several components with our under-developing system. Figure 2 shows the architecture the whole platform where modules in shaded blocks belong to GSA package. Basically, it consists of several core libraries and applications which are built on top of the libraries. The package was developed for deploying on Unix systems using C++ programing language.

The core part of software package is sensitivity analysis module which implements three aforementioned GSA methods and provides a generic API to applications. The module can serve as an independent library for global sensitivity estimation of generic models in the form of $y=h(P)$. The library does not perform model evaluation but exposes a callback function to be implemented by its callers. In our system, evaluating a model is to solve a set of ODEs in equation (1) then extract features from output time course data as explained in section II. The simulation engine module is responsible for solving ODE systems. Although alternative simulation engines are available in forms of command line tools, we opt to develop a new engine as redundant steps such as reading SBML files, solver initialization can be dismissed when running simulations in batches.

The engine is built on top of two open source libraries: libSBML [15] and CVODES [16]. The libSBML provides API to manipulate models in SBML format while CVODES is a solver for stiff and non-stiff ordinary differential equation systems (initial value problem) given in explicit form as shown in equation (1) . Since SBML only implicitly imply ODEs, the module needs to convert model into explicit ODEs structure which is consequently optimized to serve as input for CVODES. It also perform mathematical evaluation for the right hand side of the ODEs as request by CVODEs as well as handling simulation events supported by SBML standard.

Built on top of core libraries is a collection of command line interface tools. There tools serve as back end services for either GUI applications on desktop environment or web-service on HPC systems. Currently a simulation tool and a GSA tool are included in our software package.

B. *Parallelization*

The computation cost of global sensitivity analysis lies heavily on the two steps: batch evaluating models and estimate sensitivity measures. In our software, these steps are parallelized to run on multiple processors. The parallelization is accomplished using OpenMPI library. The library implements Message Passing Interface (MPI) standard and provide a C++ API to allow its applications to run in distributed memory multiprocessor architecture. The parallelization with OpenMPI allows the software to work on multicores desktop environment as well as HPC systems consisting of cluster nodes.

In the model evaluation step, the parallelization scheme is simple because of the “embarrassingly parallel” characteristic of the task. The ‘master’ processor generates model input samples, divides them equally then distribute to available processors. These “workers” then perform model evaluations with received sample by itself.

The parallelization for the sensitivity estimation step is implemented for the variance-based methods and regional-based method. The implementation is more complicated as data communication between processors is needed. For variance-based methods, the parallelization is indispensable as the amount of data generated may be huge. For the regional-based method, the main benefit of parallelization comes from a parallel quicksort algorithm that is used during the Smirnov test. However, in cases where the number of model evaluations is relatively small, for example when the investigated model is simple, or in the case of Morris method, parallelization is not necessary and sensitivity measures are estimated at the master node after gathering all simulation result from “workers”. Our GSA tool can operate in both modes: paralleled or unparalleled regarding to the complexity of the jobs.

IV. DEPLOYMENT AND TESTING

For testing the software package, we conduct experiments on a computational model of JAK-STAT signaling pathway [17]. The model in SBML format was downloaded from BioModels [18], it has 34 species and 72 kinetic parameters defined in 46 reactions. For global sensitivity analysis we varied all 72 kinetic parameters in ranges of 0.1 to 10 times of their based values. For the model output, we selected the dimerized phosphorylated $STAT_{1n}$ protein, a transcription factor that dictates many cellular processes. Sensitivity is calculated for all kinetic parameter regarding to all 100 time points of the output time course evolution. Regional method is used for all experiments and the number of sampling points N is set to 100.000.

The computation time of sensitivity analysis is divided into two periods. The first one is the total time needed for evaluating the model regarding to the N input samples while the second period is the time needed for sensitivity estimation after model evaluation has been completed. To evaluate the efficiency of our parallelization, we used the speed-up in clock time with increasing numbers of processors. The speed-up is defined as the clock time for a run with one processor divided by the clock time for the same run

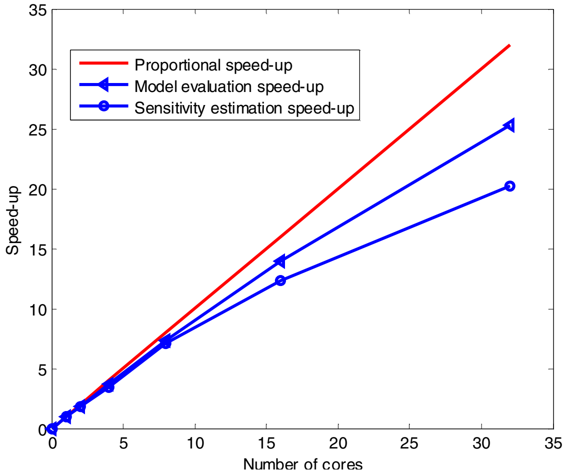

Figure 3 Scaling of run time with number of cores

on multiple processors. As show shown in Figure 3, the speed-up in model evaluation step and sensitivity estimation steps are relatively linear with the number of processors.

The software package was deployed on our cluster system which consists of 30 nodes, each has from 4 to10 cores. The total number of cores is 272. The system is running CentOS operating system. For exposing web-service to external users, Opal [19] is used to wrap our command line interface tools then deployed on a Tomcat web server. The provided web-service can be accessed by using Opal web-service client tool and ready for integrating to workflow systems.

V. Conclusion

In this work, we developed a software package for estimation of global parameter sensitivity. Most available GSA methods were implemented and parallelized to work on high performance computing systems. The software package consists of an open source library and a command line tool which can run on multicore desktop environment as well as HPC cluster systems. The tool was deployed on our cluster system to provide web-service. With a built-in efficient ODE solver engine and interface to SBML file format, the software will benefit system biologists in both convenience and performance.

References

[1] H. Kitano, “Computational systems biology.,” *Nature*, vol. 420, no. 6912, pp. 206-10, Nov. 2002.

[2] A. Saltelli et al., *Global Sensitivity Analysis: The Primer*. Wiley, 2008.

[3] K. H. Cho, S. Y. Shin, W. Kolch, and O. Wolkenhauer, “Experimental design in systems biology based on parameter sensitivity analysis using a Monte Carlo method: a case study for the TNF alpha-mediated NF-kappaB signal transduction pathway,” *SIMULATION*, vol. 79, pp. 726-739, 2003.

[4] D. Hu and J. M. Yuan, “Time-dependent sensitivity analysis of biological networks: coupled MAPK and PI3K signal transduction pathways,” *J Phys Chem A*, vol. 110, no. 16, pp. 5361-5370, 2006.

[5] Y. Zheng and A. Rundell, “Comparative study of parameter sensitivity analyses of the TCR-activated Erk-MAPK signalling pathway,” *Syst Biol (Stevenage)*, vol. 153, no. 4, pp. 201-211, 2006.

[6] Z. Zi, “Sensitivity analysis approaches applied to systems biology models.,” *IET systems biology*, vol. 5, no. 6, pp. 336-6, Nov. 2011.

[7] Jrc, “Simlab (2011) Software package for uncertainty and sensitivity analysis.” 2011.

[8] Z. Zi, Y. Zheng, A. Rundell, and E. Klipp, “SBML-SAT: a systems biology markup language (SBML) based sensitivity analysis tool.,” *BMC bioinformatics*, vol. 9, p. 342, Aug. 2008.

[9] H. Schmidt and M. Jirstrand, “Systems Biology Toolbox for MATLAB: a computational platform for research in systems biology,” *Bioinformatics*, vol. 22, no. 4, pp. 514-515, 2006.

[10] K. A. Kim et al., “Systematic calibration of a cell signaling network model.,” *BMC bioinformatics*, vol. 11, no. 1, p. 202, Jan. 2010.

[11] I. M. Sobol, “Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates,” *Mathematics and Computers in Simulation*, vol. 55, no. 1–3, pp. 271-280, 2001.

[12] M. J. W. Jansen, “Analysis of variance designs for model output,” *Computer Physics Communications*, vol. 117, no. 1–2, pp. 35-43, Mar. 1999.

[13] K. Chan, A. Saltelli, and S. Tarantola, “Winding Stairs: A sampling tool to compute sensitivity indices,” *Statistics and Computing*, vol. 10, no. 3, pp. 187-196.

[14] M. D. Morris, “Factorial Sampling Plans for Preliminary Computational Experiments,” *Technometrics*, vol. 33, no. 2, p. 161, May 1991.

[15] B. J. Bornstein, S. M. Keating, A. Jouraku, and M. Hucka, “LibSBML: an API Library for SBML ,” *Bioinformatics* , vol. 24 , no. 6 , pp. 880-881, Mar. 2008.

[16] S. Cohen and C. Hindmarsh, “Cvode, A Stiff/nonstiff Ode Solver In C,” *Computers in Physics*, vol. 10, no. 2, pp. 138-143, Mar. 1996.

[17] S. Yamada, S. Shiono, A. Joo, and A. Yoshimura, “Control mechanism of JAK/STAT signal transduction pathway,” *FEBS Letters*, vol. 534, no. 1–3, pp. 190-196, 2003.

[18] N. Le Novere et al., “BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems,” *Nucleic Acids Res*, vol. 34, no. Database issue, pp. D689-91, 2006.

[19] J. Ren, N. Williams, L. Clementi, S. Krishnan, and W. W. Li, “Opal web services for biomedical applications.,” *Nucleic acids research*, vol. 38, no. Web Server issue, pp. W724-31, Jul. 2010.