Simulation of Genetic Regulatory Networks

Rafat Parveen

Address
rafatparveen@yahoo.co.in
Asstt. Professor, Department of Computer Science,
Jamia Millia Islamia ,New Delhi, India.

Abstract: **Dizzy is a chemical kinetics simulation software framework. On up gradating this package to simulate the dynamics of complex gene regulatory networks. Using Tauleap simplex and Tauleap complex algorithms, implemented in Java. Procedure have been improved for determining the maximum leap size which accelerates the speed of simulation. This paper focuses mainly on simulating Genetic Regulatory Networks using stochastic methods of simulation and introducing τ to accelerate the speed of simulation**.

Keywords: Gene Regulatory Network, Endomesoderm, Sea Urchin, Stochastic Simulation , BioTapestry.

1. Introduction: Simulation is a powerful approach for understanding the complexity of biological systems. Recently, several successful attempts have been made for simulating complex biological processes like gene regulatory networks, metabolic pathways and cell signaling pathways[1][2] .The network models have not only generated experimentally verifiable hypothesis but have also provided valuable-insights into the behavior of complex biological systems. Many recent studies have confirmed the phenotypic variability of organisms to an inherent stochasticity that operates at a basel level of gene expression. Due to this reason, development of novel mathematical representations and efficient algorithms are critical for successful simulation of biological systems. Genetic Regulatory Networks (GRNs) control cellular state form and functions. They are responsible for executing embryonic developmental programs and, changing cellular state and metabolic processes based on environmental conditions. A specific example is the early cell specification process within the sea urchins embryo. GRNs typically involve feedback interactions among multiple genes [2][3]. The situation is frequently more complex in adult organisms, where feedback loops intertwine genetic networks closely.
Signaling and metabolic events change the state of a GRN, which in turn modifies the structure of the upstream [3]. BioTapestry is a software tool for modeling the genetic regulatory networks. The application of Bio-tapestry tool to enable computerized modeling of GRNs, we can model a network consisting of only up to 50 genes. I have upgraded the tool by using Kinetic Logic Model Framework and a number of other algorithms such that a GRN model of more then 186 genes can now be obtained. The output of BioTapestry is used as an input to Dizzy package in order to simulate the modeled GRNs. On extending the Dizzy software tool by using stochastic Tauleap complex method in order to simulate GRNs. This paper is divided into different sections. Section 2 comprises a brief overview of the Dizzy software system, whereas in section 3 explains the different simulation algorithms and in section 4 the Simulation Methodology is described. In section 5,a new Tau selection procedure is proposed , in sections 6 sea urchins gene expression data and finally in section 7 we have discuss the results and conclusions.

2. Overview of the Dizzy Software System

In this section, we give an overview of the major features of Dizzy, a software framework for simulating the dynamics of complex Genetic Regulatory Network systems. Dizzy provides a collection of simulators for solving the dynamics of a model. Features of Dizzy simulator are:

a. ***Modular simulation framework*****:** Dizzy employs a modular design in which each simulator is a software unit that conforms to a simple, well-defined interface specification. This architecture facilitates an iterative model development cycle in which the model is analyzed using various simulation algorithms [4].

b. ***Templates reusable and hierarchical model elements:*** Dizzy's model definition language permits the definition of reusable, parameterized model elements called templates. This enables the construction of a prepackaged library of templates that can simplify the task constructing a complex model.

c. ***Multi-step and delayed reaction processes*****:** Dizzy enables the simulation of complex multi-step processes such as elongation and translocation during transcription or translation, through two methods. One may define it as a multi-step reaction process, a reaction process with an intrinsic, phenomenological time delay [5][6].

d. ***Estimation of steady-state stochastic noise*****:** Dizzy provides a feature for estimating or calculating the steady-state stochastic fluctuations of the species in a biochemical model, requiring only the solution of the deterministic dynamics [7][8].

e. *Integrated, graphical, and portable software framework*: Dizzy has several important software features including integration with external software tools, a graphical user interface (GUI) and a high level of portability.Many software tools are available for solving the deterministic and stochastic dynamics of complex biochemical networks but not for GRNs. A detailed overview of the most common algorithms for simulating GRNs is presented in section 3. We compare some of the most widely used simulation software tools against a specific list of simulation algorithms and features described above.To the best of our knowledge, Dizzy is the only software tool available that includes all the features enumerated above. In addition, it includes novel implementations of the number of simulation algorithms[9].

3. Simulation Algorithms: A Number of algorithms can be used for simulating the GRNs. These can be divided into two broad categories: a. Deterministic and b. Stochastic Algorithms

a. Deterministic Algorithms:

If no noise or any stochastic variations are present in the process, then we may use Deterministic Algorithms to solve a group of non-linear differential equations. If the system includes both very fast and very slow dynamics, that is some reactions are much faster then others, the system is called stiff. Stiff systems are hard to simulate since the fast dynamics require for short step size and the slow dynamics increase the total simulation time interval. Using a small step size, the simulation of the whole process becomes very slow[10]. Consequently, some numerical algorithms are developed especially for the simulation of this kind of systems. The deterministic algorithms available in Dizzy are listed below.

i. *Fifth order Runge-Kutta Method:* This method is particularly useful for simulating models in which a derivative function is discontinuous & the step size is adaptively controlled, based on a fourth order error estimation formula. Both relative and absolute error tolerances may be independently specified, as well as the initial step size[11].

ii. Fifth Order RK Fixed: In this method, the differential equations are solved using a finite difference method, with a fixed step-size. The step size is specified by the user, as a fraction of the total time interval for the simulation.

iii. ODE to Java-dopr54-adaptive: In this algorithm control adaptive step-size is used. Implemented by Murray Patterson and Raymond Spiteri.

iv. ODE to Java-imex443-stiff: An implicit-explicit ODE solver with step doubling. Works well for models with a high degree of stiffness.

b. Stochastic Algorithms:

Gene regulation is an inherently stochastic process, which cannot be exactly simulated by deterministic algorithms. In addition, the stochastic algorithms are designed for continuous changes in the state[12]. Some genes in the network may be weakly expressed but the model must handle the exact numbers of genes. In these cases the stochastic simulation methods have to be used .For biological systems involving genes of small populations, the stochastic simulation algorithm (SSA) derived by Gillespie is an essentially exact procedure for studying noise in gene networks systems[13][14]. However, the computational load of the SSA is often very high when it is applied to simulate large biological systems. Thus, it is imperative to design efficient numerical methods for simulating stochastic Gene Regulation Networks. There are two significant approaches for reducing the computational time of SSA describe in methodology section. In dizzy, the following stochastic algorithms are realized[15]

i. Gibson-Bruck : An algorithm used for simulating the large scale models but are less dynamic.

ii. Gillespie : This algorithm is useful for simple systems with less complexity

iii. Tauleap-Simple: An approximate accelerated stochastic simulator implemented using the Gillespie Tau-Leap algorithm. This implementation is intended for models in which the models are less complex.

iv. Tauleap-Complex: An approximate accelerated stochastic simulator implemented using the
Gillespie Tauleap algorithm. This implementation is used for large complex models.

4. Simulation Methodology

In a Genetic Regulatory Networks system, the state vector $\mathbf{X}(t) = (\mathbf{X_1}(t), \ldots, \mathbf{X_N}(t))$, where $X_i(t)$ is the number of gene of species S_i in the system at time t, evolves stochastically because of the inherent random interactions of genes. Random genes interactions give rise to random chemical transmutations in accordance with some specified set of reaction channels $\{P_1,...,P_M\}$The dynamics of genes R_j are mathematically defined by a *propensity function* a_j together with a *state-change vector* $v_j = (v_{1j},...,v_{nj})$ $:_{aj}(X)dt$ gives the probability that one R_j reaction will occur in state $\mathbf{X}$ during the next infinitesimal time interval dt, and $\boldsymbol{\tau}_{ij}$gives the change in the S_i molecular population produced by one R_j reaction[16].

For simulating the stochastic evolution of $\mathbf{X}(t)$, there exist several exact procedures that actualize every molecular reaction event[17][18]. But efforts to model the complex biological networks inside living cells, where small number of genes can set the stage for major stochastic effects[19] **,**have revealed the need for faster, possibly less meticulous stochastic simulation strategies.The newly proposed "leaping" methodology attempts to sacrifice accuracy for greater speed, and to do so in a way that segues as the system size becomes infinite to standard solution methods for the conventional deterministic reaction rate equation. The "$\boldsymbol{\tau}$ -leap method," for instance, tries to leap down the history axis of the system by some chosen time $\boldsymbol{\tau}$ that encompasses many reaction events. But theoretical considerations demand that the size of $\boldsymbol{\tau}$ be constrained by a *Leap Condition*, which says that the state change in any leap should be small enough that no propensity function will experience a macroscopically significant change in its value. The mathematical rationale for the $\boldsymbol{\tau}$ -leap method [19] is the fact that, to the extent that the Leap Condition is satisfied, then given $\mathbf{X}(t)$ $\boldsymbol{\tau}$ $\mathbf{x}$, the number of times $K_j(\boldsymbol{\tau}, \mathbf{x})$ that genes R_j will be express in $(t, t + \boldsymbol{\tau})$ can be approximated by a *Poisson* random variable:

$$K_j(\tau; \mathbf{x}) \approx \rho_j(a_j(\mathbf{x}), \tau) \quad (1)$$

This is so because the generic Poisson random variable $\rho(a, \tau)$ can be defined as the number of events that will occur in a time τ, given that the probability for an event to occur in the next infinitesimal time dt is adt, where a can be any non-negative *constant*.

This last requirement is approximately ensured by the Leap Condition, and the consequent approximation (1) allows us to estimate the state change in the leap,

$$X(t+\tau) - x \equiv \Lambda(\tau; x) = \sum_{j=1}^{M} K_j(\tau; x)\, \upsilon_j \quad (2)$$

by simple Poisson sampling[8]. But for this approach to be practicable, we need a reliable, expeditious, and preferably automatic way of determining the *largest* value of τ that is compatible with the Leap Condition.A plausible mathematical framing of the leap condition would be require the leap time τ to be such that

$$|\, a_j(\mathbf{x} + \boldsymbol{\Lambda}(\tau; \mathbf{x})) - a_j(\mathbf{x}) \,| \le \varepsilon\, a_0(\mathbf{x}), \quad \forall_j = 1,\ldots,M \quad (3)$$

where τ is a pre-specified *error control parameter*. But of course, smaller values of τ also imply shorter leaps, and therefore longer simulation times. How can we find the largest value of ε that is consistent with (3) for a specified value of τ? This would be a reasonably straightforward problem were it not for the fact that the left-hand side of (3) is a *random variable* (since $\boldsymbol{\Lambda}(\tau; \mathbf{x})$ is a random variable). In any case, we would like to make our determination of τ without performing repeated "trial" leaps, checking after each one to see if condition (3) is satisfied and adjusting τ accordingly, such a post-leap procedure not only would consume much time and many random numbers, but it might also discriminate against statistically rare but nonetheless legitimate large changes in the system's state.In this section we present a new τ -selection procedure that is more robust.

6. Sea Urchins Gene Expression Data

5. The New Tau-Selection Procedure

The new τ -selection procedure requires us to determine in advance first the M^2 functions

$$f_{jj'}(x) = \sum_{i=1}^{N} \frac{\partial aj(x)}{\partial xi}$$

$$j,j' = 1,\ldots,M) \quad (4)$$

and then the $2M$ functions

$$\mu_j(\mathbf{x}) = \sum_{J'=1}^{M} fjj'(x)aj'(x)$$

$$(j=1\ldots,M) \quad (5a)$$

$$\sigma^2_j(\mathbf{x}) = {}^{\Delta}\Sigma^2 f_{jj'}(x)a_{j'}(\mathbf{x})$$

$$(j= 1,\ldots,M) \quad (5b)$$

$$(j,j' = 1,\ldots,M) \quad (4)$$

and then the $2M$ functions

$$\mu_j(\mathbf{x}) = \sum_{J'=1}^{M} fjj'(x)aj'(x)$$

This obviously represents some computational overhead, but the task is not quite as daunting as it might at first appear, the functional dependence of a_j on each x_i is typically be very simple often constant, sometimes linear,but rarely more then quadratic. Furthermore, for large systems the matrix ν_{ij} will typically be sparse. In any case, with the functions (4) and (5) determined, then given a current state $\mathbf{X}(t) = \mathbf{x}$, the largest τ that is compatible with the Leap Condition (3) is taken to be

$$\tau = \underset{J\varepsilon[1,M]}{M} \left\{ \frac{\in a0(x)}{\mu j(x)}, \frac{\in 2a02(x)}{\sigma 2\, j(x)} \right\} \quad (6)$$

Acceptance of this τ -value is, however, subject to the provision that if it is less then a few multiples of $1/a_0(\mathbf{x})$, which is the mean time step for the exact stochastic simulation algorithm .

# time	Gene 01	Gene 02	Gene 03	Gene 04	Gene 05	Gene 06	Gene 07	Gene 08	Gene 09	Gene 10	Gene 11	Gene 12	Gene 13	Gene 14	Gene 15
0.000	100.350	99.933	100.070	0.320	0.001	0.001	0.001	0.001	0.604	0.604	0.001	0.001	0.001	0.001	0.604
10.101	108.310	98.566	101.680	7.932	0.337	0.337	0.337	0.619	14.447	14.447	0.619	0.619	0.619	0.619	14.447
20.202	114.820	97.661	103.050	14.929	1.200	1.200	1.200	2.153	26.291	26.291	2.153	2.153	2.153	2.153	26.291
30.303	118.640	97.243	103.910	19.476	2.052	2.052	2.052	3.621	33.530	33.530	3.621	3.621	3.621	3.621	33.530
40.404	122.580	96.920	104.850	24.665	3.311	3.311	3.311	5.729	41.342	41.342	5.729	5.729	5.729	5.729	41.342
50.505	126.180	96.750	105.810	30.029	4.938	4.938	4.938	8.372	48.905	48.905	8.372	8.372	8.372	8.372	48.905
60.606	129.180	96.732	106.710	35.186	6.823	6.823	6.823	11.334	55.673	55.673	11.334	11.334	11.334	11.334	55.673
70.707	131.620	96.839	107.570	40.143	8.938	8.938	8.938	14.548	61.700	61.700	14.548	14.548	14.548	14.548	61.700
80.808	133.550	97.047	108.390	44.907	11.257	11.257	11.257	17.955	67.041	67.041	17.955	17.955	17.955	17.955	67.041
90.909	135.010	97.334	109.200	49.486	13.756	13.756	13.756	21.503	71.745	71.745	21.503	21.503	21.503	21.503	71.745
101.010	136.040	97.681	109.980	53.888	16.415	16.415	16.415	25.146	75.858	75.858	25.146	25.146	25.146	25.146	75.858
111.110	136.670	98.071	110.740	58.120	19.211	19.211	19.211	28.844	79.425	79.425	28.844	28.844	28.844	28.844	79.425
121.210	136.940	98.488	111.490	62.190	22.129	22.129	22.129	32.561	82.487	82.487	32.561	32.561	32.561	32.561	82.487
131.310	136.880	98.919	112.230	66.105	25.150	25.150	25.150	36.266	85.082	85.082	36.266	36.266	36.266	36.266	85.082
141.410	136.520	99.353	112.960	69.872	28.258	28.258	28.258	39.933	87.247	87.247	39.933	39.933	39.933	39.933	87.247
151.520	135.880	99.779	113.680	73.496	31.441	31.441	31.441	43.538	89.015	89.015	43.538	43.538	43.538	43.538	89.015
161.620	135.000	100.190	114.390	76.986	34.684	34.684	34.684	47.062	90.420	90.420	47.062	47.062	47.062	47.062	90.420
171.720	133.900	100.570	115.090	80.346	37.977	37.977	37.977	50.488	91.490	91.490	50.488	50.488	50.488	50.488	91.490
181.820	132.590	100.920	115.780	83.582	41.307	41.307	41.307	53.800	92.255	92.255	53.800	53.800	53.800	53.800	92.255

7. Results and Conclusions

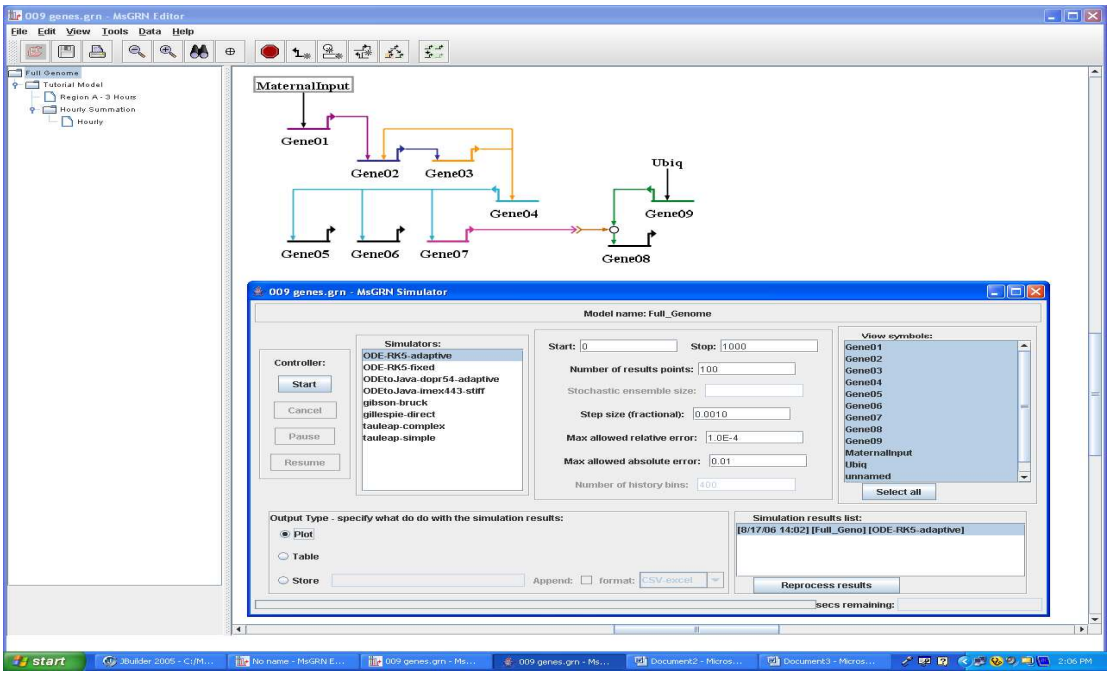

Fig.1 A screen capture of the Dizzy program showing a simulation of a model of genetic regulatory network consisting of different number of genes in sea urchin's embryo.

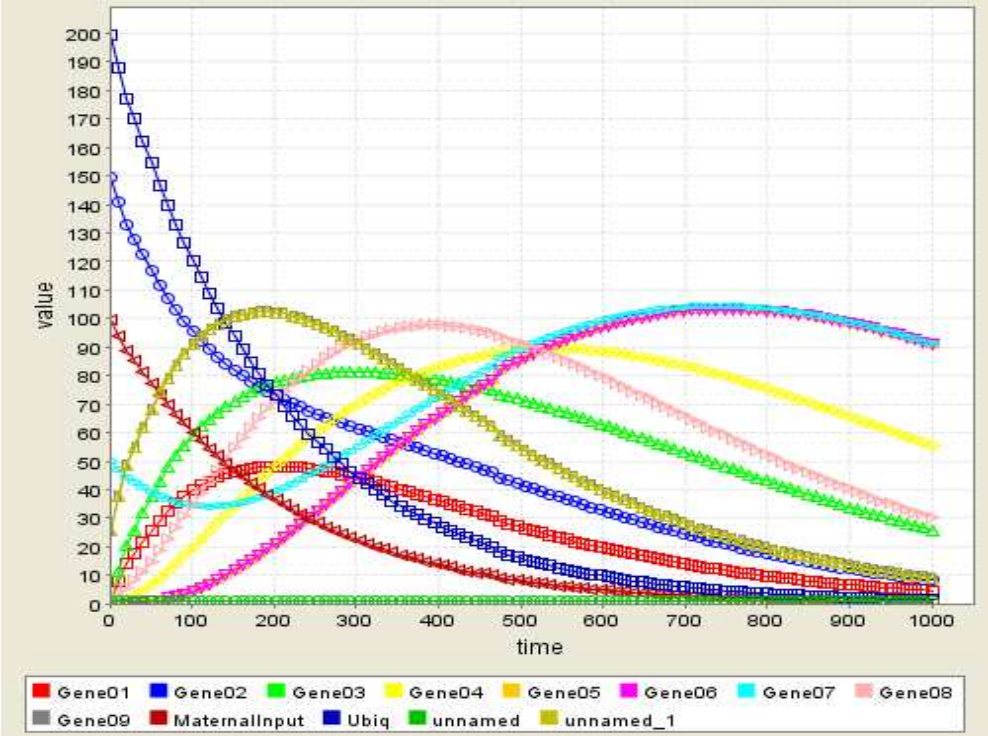

Fig.2 Simulation of GRN consisting of 9 genes of sea urchins embryo

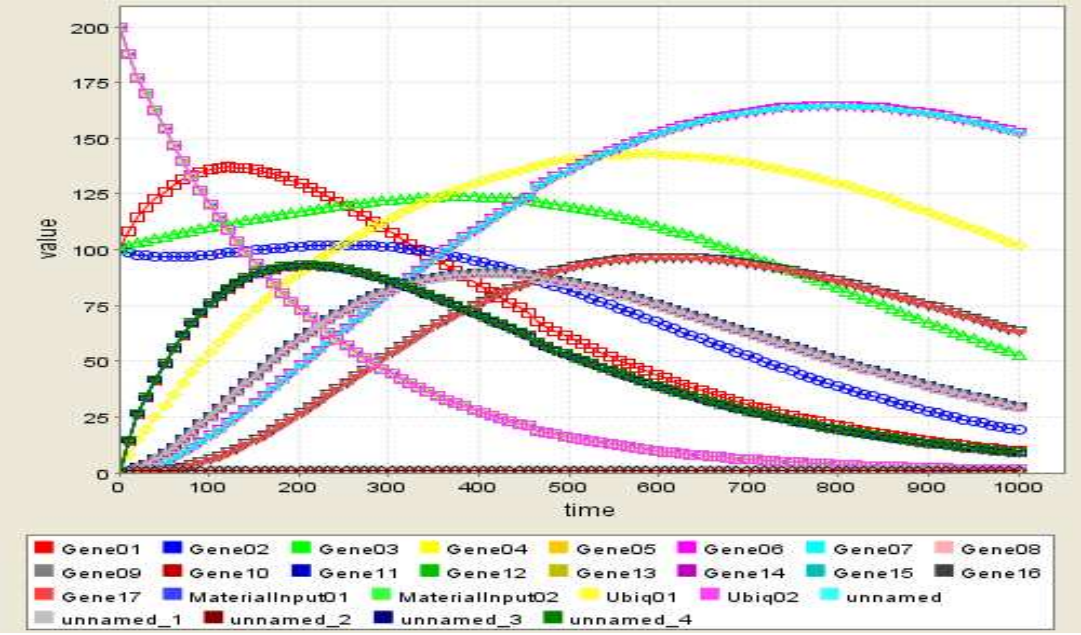

Figure 3. Simulation of GRN consisting of 17 genes of sea urchins embryo

Algorithms	**Computational cost**	**Modeling Knowledge**	**Speed**	**Accuracy**
Tauleap Simplex	High	Medium	Fast	High
Tauleap Complex	Low	High	Very Fast	Medium

Table1: Comparison of various stochastic simulating algorithms

In this paper we have presented a comprehensive software tool for conducting stochastic simulations of the dynamics of complex gene regulatory networks. The tool is particularly well suited for simulating the dynamics of integrated large-scale genetic, metabolic and signaling networks. In this paper we have implemented & tested various forms of stochastic algorithms and their application to simulation of biological systems. Each algorithm imposes a certain constraint on the computational power, knowledge of the system and input of the numerical parameters. In addition, the algorithms provide different abstractions of the system and produce solutions with very accuracy. Tauleap methods, Tauleap Simplex and Tauleap Complex algorithms are among the fastest simulation algorithms. However, due to various numerical treatments to the algorithms, both Tauleap methods require substantial modeling knowledge to ensure the accuracy of the solutions. Besides that, both the algorithms are efficient algorithms, which increase the speed of simulation without sacrificing the accuracy of solutions.

As the number of genes in a network increases the network becomes more complex because the connecting lines start criss crossing leads to more complex network.By simulating these genetic regulatory networks we can choose the less complex network that corresponds to the less complex biological system and our aim in systems biology is to find the less complex system so that we can use that, in preventive medicine because gene target prediction becomes easier. If we compare the results obtained in fig.2 and figure 3 which shows that we can easily choose the less complex network because as the number of genes increases the lines in graphs starts overlapping consequently our graphs becomes more complex, that indicates more complex network. In this way we have simulated the GRNs.

References:

[1] . E. H. Davidson et al., “A genomic regulatory network for development," Science,vol.295, pp.16691678, 2002.

[2]. S. R. Biggar and G. R. Crabtree,” Cell signaling can direct either binary or graded transcriptional responses, " EMBO J., vol. 20, no. 12, pp. 31673176, 2001.

[3]. P. de Atauri, D. Orrell, S. Ramsey, and H. Bolouri, Evolution of `design' principles in biochemical networks," IEE Systems Biology., vol. 1, pp. 2840, 2004.

[4]. H. Bolouri and E. H. Davidson, “Transcriptional regulatory cascades in development: Initial rates, not steady state, determine network kinetics," Proc. Nat. Acad. Sci. USA, vol. 100, no. 16, pp. 93719376, 2003.

[5]. P. Guptasarma, Does replication-induced transcription regulate synthesis of the myriad low copy number proteins of E. coli?," BioEssays, vol. 17, pp. 987997, 1995.

[6]. M. B. Elowitz, A. J. Levine, E. D. Siggia, and P. S. Swain, Stochastic gene expression in a single cell," Science, vol. 297, pp. 11831186, 2002.

[7]. E. M. Ozbudak et al., “Regulation of noise in the expression of a single gene," Nature Gen., vol. 31, pp. 6973, 2002.

[8]. W. J. Blake, M. Kaern, C. R. Cantor, and J. J. Collins,” Noise in eukaryotic gene expression," Nature, vol. 422, pp. 633637, 2003.

[9]. D. T. Gillespie, “Approximate accelerated stochastic simulation of chemically reacting systems," J. Chem. Phys., vol. 115, no. 4, pp. 17161733, 2001.

[10]. E. L. Haseltine and J. B. Rawlings, Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics, " J. Chem. Phys., vol. 117, pp. 69596969, 2002.

[11]. D. T. Gillespie and L. R. Petzold, “ Improved leap-size selection for accelerated stochastic simulation," J. Chem. Phys., vol. 119, no. 16, pp. 82298234, 2003.

[12]. J. Puchalka and A. M. Kierzek, Bridging the gap between stochastic and deterministic regimes in the kinetic simulations of the biochemical reaction networks," Biophys. J., vol. 86, pp. 13571372, 2004.

[13]. T. R. Kiehl, R. M. Mattheyses, and M. K. Simmons, Hybrid simulation of cellular behavior,"Bioinf., vol. 20, no. 3, pp. 316322, 2004.

[14]. C. V. Rao and A. P. Arkin, Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the Gillespie algorithm," J. Chem. Phys., vol. 118, no. 11, pp. 49995010, 2003.

[15]. K. Vasudeva and U. S. Bhalla, Adaptive stochastic-deterministic chemical kinetic sr.bioeng.washington.edu, 2003.

[16]. D. T. Gillespie, “A general method for numerically simulating the stochastic time evolution of coupled chemical reactions," J. Comp. Phys., vol. 22, pp. 403434, 1976.

[17]. D. A. McQuarrie, “Stochastic approach to chemical kinetics," J. Appl. Prob., vol. 4, p. 413, 1967.

[18]. M. A. Gibson and J. Bruck, “Efficient exact stochastic simulation of chemical systems with many species and many Channels," J. Phys. Chem. A, vol. 104, pp. 18761889, 2000.