

# DISCRETE NONDETERMINISTIC MODELING OF THE FAS PATHWAY

JOHN JACK , ALFONSO RODRÍGUEZ-PATÓN ,  
OSCAR H. IBARRA and ANDREI PĂUN

*Department of Computer Science/IfM, Louisiana Tech University,  
P.O. Box 10348, Ruston, LA 71272, USA,*

*Universidad Politécnica de Madrid - UPM, Facultad de Informática  
Campus de Montegancedo S/N, Boadilla del Monte, 28660 Madrid, Spain*

*Department of Computer Science, University of California,  
Santa Barbara, CA 93106, USA,*

*Bioinformatics Department,  
National Institute of Research and Development for Biological Sciences,  
Splaiul Independenței, Nr. 296, Sector 6, Bucharest, Romania,*

Computer modeling of molecular signaling cascades can provide useful insight into the underlying complexities of biological systems. We present a refined approach for the discrete modeling of protein interactions within the environment of a single cell. The technique we offer utilizes the Membrane Systems paradigm which, due to its hierarchical structure, lends itself readily to mimic the behavior of cells. Since our approach is nondeterministic and discrete, it provides an interesting contrast to the standard deterministic ordinary differential equations techniques. We argue that our approach may outperform ordinary differential equations when modeling systems with relatively low numbers of molecules – a frequent occurrence in cellular signaling cascades.

Refinements over our previous modeling efforts include the addition of nondeterminism for handling reaction competition over limited reactants, increased efficiency in the storing and sorting of reaction waiting times, and modifications of the model reactions. Results of our discrete simulation of the type I and type II Fas-mediated apoptotic signaling cascade are illustrated and compared with two approaches: one based on ordinary differential equations and another based on the well-known Gillespie algorithm.

*Keywords:* Signal transduction pathway; simulation of pathways; Membrane Systems; ODE; discrete simulation; nondeterministic simulation.

## 1. Introduction

Cellular apoptosis, or programmed cell death, is an important process in biological systems. It is an effective method for the elimination of unwanted or damaged cells

in an organism. Kerr [1] labeled apoptosis as being distinct from necrotic cell death, citing a lack of inflammation of the tissue among other differences. The study of apoptotic signaling cascades is especially interesting to the fields of biology and medicine since defects in apoptotic signaling has been linked to cancer and various autoimmune disorders.

In the past decade, there has been a wealth of information discovered on the Fas-mediated apoptotic pathway. Deviations from normal cellular activity induced by the human immunodeficiency virus [2] including suppression of Fas-mediated apoptosis in HIV-infected CD4+ T cells [3] and the death of healthy, uninfected 'bystander' cells [4]. Besides autoimmune disorders, cancerous cells have been shown to be resistant to Fas-induced apoptosis [1]. Because many of the proteins involved in this pathway have been identified, it is an ideal candidate to illustrate the effectiveness of our modeling technique. We hope to use our simulator to provide a deeper understanding of the subtleties behind Fas signal transduction.

In this paper, we provide the framework for our discrete nondeterministic modeling method: a Membrane Systems approach. Our *Nondeterministic Waiting Time* (NWT) algorithm is a refined version of our previous modeling efforts [5]. Section 2 provides the refinements to our previous Membrane System approach. Section 3 presents a full description of our algorithm. In Section 4, we provide results from applying our algorithm to a small signaling cascade with low molecular multiplicities, illuminating the differences in results obtained by our approach and an ordinary differential equations simulation technique. Results from our simulation of the Fas pathway are given in Section 5. Section 6 contains a discussion on future avenues of research.

N.B., we will use the words rule and reaction interchangeably throughout this paper. In the context of Membrane Systems, we have rules associated with symbols, whereas in the context of the cell we have reactions associated with proteins/molecules.

## 2. Refinements On Previous Work

we described a discrete approach for simulating molecular signal cascades which employs the Membrane Systems paradigm. We also provided some preliminary results on Fas pathway simulation. Here we provide a simulation technique which is more robust and efficient than our previous effort. The claim of robustness is satisfied by the addition of nondeterminism, and efficiency by implementing a nonstandard heap to sort the reactions. Since our approach is nondeterministic and discrete, we believe it can offer a unique perspective on molecular signaling cascades. Hence, we argue that our algorithm is more advantageous than standard modeling techniques involving ordinary differential equations.

The advantage is understood when one considers reaction competition over limited reactants – that is, a situation where two or more reactions are attempting to simultaneously use reactants of low molecular multiplicity. Ordinary differential equations are appropriate when one seeks the *average* behavior of systems invol-

ing large populations – e.g., modeling dynamics of large cell populations over time. However, molecular signaling cascades can often involve very low molecular multiplicities – e.g., the  $\sim 700$  molecules of Vpr [2] affecting apoptosis in HIV infection

Using ordinary differential equations to model these situations may be misleading. We will clarify the difference between discrete and continuous modeling in the following example:

Assume we have two different reactions competing over a single molecule. An ODE method will allow both reactions to be applied, borrowing a *fraction* of each molecule to satisfy each equation. In contrast, our discrete approach maintains molecular integrity. Our NWT algorithm allows the single molecule to satisfy only one of the two reactions (nondeterministically chosen), thus sending the biological system on a different evolutionary path than that of the ODE simulation. The aggregate effects of these nondeterministic decisions over time, could lead to interesting results describing the molecular behavior of a cell.

Our second improvement, using a specialized heap to store reactions, allows the NWT algorithm to be computationally competitive with ordinary differential equations methods. Our algorithm employs a heap, but with two nonstandard properties:

- (i) once initialized, the heap does not grow or shrink;
- (ii) multiple nodes throughout the heap must be updated simultaneously (during the same update step of the heap).

The implementation of a priority queue for sorting values associated with reactions has its roots – which we will elaborate on further during the description of our algorithm in the next section. By storing reactions in this way, our mesoscopic approach has an execution time similar to the generally faster macroscopic approaches – techniques using ODEs.

Additionally, we have made improvements to the Fas model

Incorporating these model refinements allows us to compare our discrete simulator and the continuous ODE technique provided in [9].

### 3. Our Membrane Systems Approach

We will now provide the reader with a brief description of our proposed modeling technique, followed by a statement of the algorithm and a discussion on its main ideas.

#### 3.1. *Description of technique*

We have designed a Membrane System which tracks the changes of molecular multiplicities over time. The evolution of the system is led by the simulation of individual chemical reactions that are asynchronous and occur over variable lengths of time. The amount of time required for a given reaction to occur is governed by the number of the reactants present in the system. In this way, our simulator obeys the *Law of Mass Action* – reaction rate depends proportionally on reactant multiplicities.

The driving force behind our simulator is the reaction *Waiting Time* (WT). Throughout the entire simulation run, each reaction has a WT associated to it. The WT informs the simulator the next instant a reaction is slotted to occur. We will first describe the mathematics behind WT calculation, then provide an example to help clarify how the calculation is used.

To each reaction ( $R$ ), there is an associated kinetic rate constant ( $k_R$ ). These kinetic rate constants are provided in the literature with units based on concentrations, which are required for an ODE-based simulation. The units for these kinetics rate constants are  $s^{-1}$  for first order reactions,  $nM^{-1}s^{-1}$  for second order reactions and  $nM^{-1}nM^{-1}s^{-1}$  for third order reactions. Since our simulator employs individual molecules, we need to convert these constants into a suitable form, based on multiplicities instead of concentrations. Our simulator uses the kinetic rate constant to compute an appropriate kinetic constant per reaction ( $const_R$ ) according to the following equation:

$$const_R = \frac{k_R}{V^{i-1} \times N^{i-1}} \quad (1)$$

where  $V$  is the volume of the system,  $N$  is Avogadro's constant ( $6.0221415 \times 10^{23}$ ) and  $i$  is the number of reactants involved in the reaction.

During initialization, the kinetic constant ( $const_R$ ) is calculated and stored for every reaction in the model. For all the simulations described in this paper, we assume the volume is constant. Thus, the kinetic constant for any reaction is indeed a constant. Using this kinetic constant and the molecular multiplicities of the system, the simulator calculates the initial WT for every reaction. For a second order reaction, the calculation of WT is as follows:

$$wt_R = \frac{1}{const_R * |A| * |B|} \quad (2)$$

where  $A$  and  $B$  are the two reactants required for reaction  $R$ , and  $|A|$  and  $|B|$  are the number of molecules present in the system at the moment of WT calculation. Waiting time calculations for first or third order reactions are similar, having one or three reactants, respectively.

Suppose a reaction ( $R1$ ) has a waiting time of five seconds. The simulator will apply  $R1$  exactly once at time  $t+5$ , where  $t$  represents the simulation time when the WT was calculated. After  $R1$  is applied, the number of reactants for it has changed, so its WT is recalculated. Additionally, the WT for all reactions using products or reactants of  $R1$  must be recalculated, since the multiplicities have changed.

### 3.2. Nondeterministic Waiting Time (NWT) algorithm

We present our algorithm below:

1. **Initialization:** Read information on pathway (SBML code). Compute the kinetic constant ( $const_R$ ) and initial WT for all reactions in the system. Build a min-heap of reactions (sorted by the WT). Set simulation time to zero ( $t = 0$ ).

2. **Select Rule:** Select the reaction with the lowest WT. If there is a tie, go to step 3. If not, proceed to step 4.
3. **Handle Tie:** If there are enough reactants available to satisfy all reactions in the tie, execute all the tied reactions in step 4. If there are not enough reactants to accommodate all the tied reactions, nondeterministically apply reactions until reactant supply is exhausted.
4. **Apply Rule:** Update the multiplicities of the reactants and products involved in the reaction(s) from step 2. Aggregate the simulation time ( $t = t + \text{WT}$ ).
5. **Update Rules and Maintain Heap:** Recalculate the WT for any reactions whose reactants include a reactant or product of the applied reaction(s). For each such reaction compare the new WT with the existing WT and keep the smallest of the two values. After WT calculation, ensure the min-heap property is maintained around each recalculated node.
6. **Check Done:** If the desired simulation time has not been reached, go back to step 2.

Step one, the initialization phase, is run only once during execution. During this step, we establish an array of reactions and build a min-heap with the standard bottom-up approach (sorted by WTs). Step two, selecting a rule, can be accomplished in constant time, since we merely access the top node of the heap – the first index of the array. Determining if there is a tie for minimum waiting time is accomplished simply by recursively checking the children of a tied node (starting with the head node). Since ties are relatively infrequent and small in the number of reactions involved, the recursion is not computationally or memory intensive. Given a tie, it is computationally inexpensive to check if there is a sufficient number of reactants by initializing a reactant list in each reaction node pointing to current multiplicity.

When reactions are involved in a tie, pointers to the reaction nodes are placed in a temporary array. Therefore, nondeterministically applying as many rules as possible consists of generating random integers from 1 to  $x$ , where  $x$  is the number of reactions in the temporary array, and attempting to apply (if sufficient reactants exist) the rule whose index is randomly chosen. Whether a rule is applied or not – insufficient reactants – it is removed from the temporary array until there are no reactions left to apply.

After applying the reaction(s), we must recalculate all reactions affected by the multiplicity changes of the system. Some reactions will end up with a new WT, which could be smaller than the previous one; thus they may need to be repositioned in the heap. Therefore, as we recalculate each WT in step five, we must perform some heap maintenance.

In order to efficiently handle the dynamics of the reaction waiting times during a simulation run, we proceed with updating the heap in a nonstandard manner. Gibson and Bruck described a method for updating a priority queue containing

the stochastic values of their Gillespie Algorithm adaptation. we define the following functions for our heap implementation:

- *Swap(A,B)*: Exchange the positions of reactions  $A$  and  $B$  in the heap.
- *BuildHeap()*: Establish the min-heap property, based on reaction WT. (Bottom-up technique).
- *UpdateHeap(A)*: Upon recalculation of the WT for reaction  $A$ , ensure that the min-heap property is satisfied regarding  $A$  – i.e., the parent and children (if they exist) of  $A$  are not in violation of the min-heap property.

Each node in the heap contains all the information for a particular reaction including: (a) the simulation time for the next execution of the reaction ( $t+WT$ , where  $t$  is the simulation time when the WT was last calculated), (b) the kinetic constant ( $const_R$ ) for calculation of WT, (c) an array pointing to the reactants and one pointing to the products, telling the simulator what multiplicities to update when the reaction is applied, (d) index of the reaction in the heap.

As we stated earlier, the heap is initially built with a bottom-up approach. However, the maintenance of our heap is accomplished in a nonstandard way. The heap does not grow or shrink during the simulation run because we begin and end with the same number of reactions. When a reaction is applied and its WT is recalculated, the new WT is often very close to the old one – a consequence of the stoichiometry of biological systems. Therefore, the applied rule will frequently end up near the top of the heap after WT recalculation and heap maintenance. Because of this, we do not remove the top node of the heap when applying a rule. Instead, we leave the node at the top of the heap and, upon recalculation of the WT, bubble the node down as needed. Additionally, when we recalculate the rules affected by the multiplicity changes from an applied rule, we leave the nodes in place and bubble up or down as necessary until the min-heap property is satisfied.

The standard heap approach, where the top nodes are removed after being applied and new nodes are added to the bottom and bubbled up, requires  $\log(r)$  resorting time, where  $r$  is the total number of reactions. As a consequence of the kinetics of signaling cascades, an increase or decrease of one molecule for the reactants does not drastically change the WT of a reaction. Therefore, our *UpdateHeap* method can accomplish the resort in  $\log(r')$ , where  $r'$  is a number much smaller than  $r$ . Moreover, the node will not move very far up or down the heap each time the *UpdateHeap* method is called.

There are bounds on the number of calls to the *UpdateHeap* method. To clarify, we provide several assumptions about signaling cascades which are often valid:

1. Each reaction involves at most six distinct proteins in the pathway considered.
2. The number of reactions having the same reactant is at most five (but usually three).
3. Due to the variety of molecular multiplicities and reaction rates, there are not many reactions occurring at the same instant.

Our previous algorithm had a runtime of  $O(n^2 \log n)$ . However, based on this new heap implementation and the assumptions listed above, our new algorithm has a runtime of  $O(n \log n)$ . The previous algorithm was capable of simulating eight hours of Fas signal transduction in thirty to forty minutes. The modified approach presented in this paper completes the same simulation in three to four minutes.

#### 4. Example of Nondeterminism

In the interest of understanding the behavior of our NWT algorithm in contrast to a deterministic ordinary differential equations method, we have designed a small signaling cascade (see Fig. 1). In this model,  $m1$  and  $m3$  competitively bind to  $m2$ . The  $m1$  protein attempts to activate  $m2$ , which leads to the upregulation of  $m4$ . The  $m3$  protein acts as an inhibitor to the upregulation of  $m4$ , through binding and inhibition of  $m2$  activation.

Although this is a simplified model, designed to illustrate the nondeterminism of our algorithm, the role of  $m2$  is similar to the transactivator protein of HIV, Tat, which regulates the viral gene expression. Tat has been shown to increase HIV gene expression by 200-fold [10]. Without the increased transcription rate, the infected cell can not create new virions.

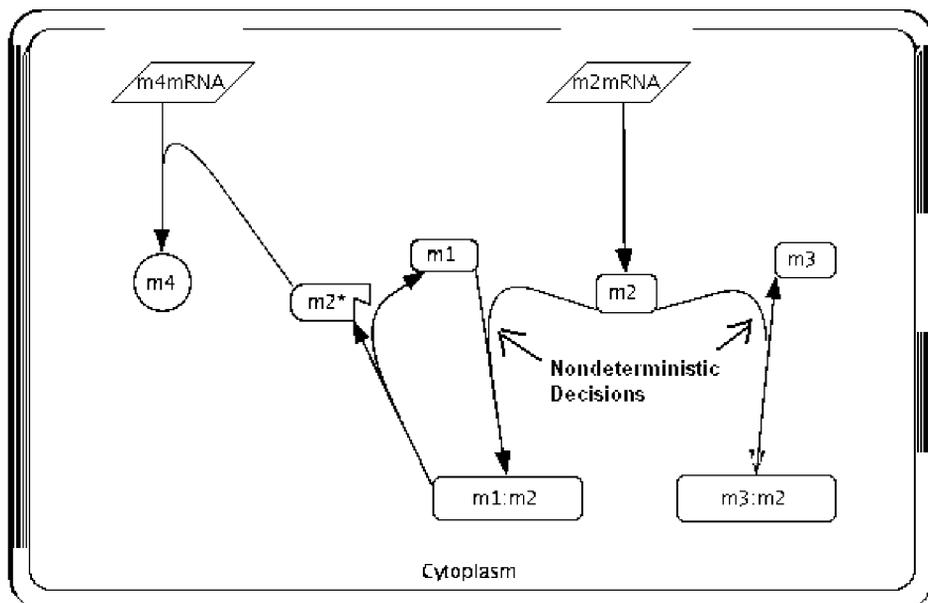


Fig. 1. An example signaling cascade. Here  $m1$  and  $m3$  competitively bind to  $m2$ . When  $m1$  binds to  $m2$ , the result is the activation of  $m2$ . After activation,  $m2^*$  proceeds to upregulate  $m4$  expression. The  $m3$  protein acts as an inhibitor for the upregulation of  $m4$ , since it binds to  $m2$  and prevents its activation.

We translated the reactions depicted in Fig. 1 into a system of ordinary differ-

ential equations

$$\begin{aligned}
 \frac{d[m1]}{dt} &= -k_1[m1][m2] + k_3[m1 : m2] \\
 \frac{d[m2]}{dt} &= -k_1[m1][m2] - k_1[m3][m2] + k_2[m2_{mRNA}] \\
 \frac{d[m3]}{dt} &= -k_1[m3][m2] + k_3[m3 : m2] \\
 \frac{d[m1 : m2]}{dt} &= k_1[m1][m2] - k_3[m1 : m2] \\
 \frac{d[m3 : m2]}{dt} &= k_1[m3][m2] - k_3[m3 : m2] \\
 \frac{d[m2^*]}{dt} &= k_3[m1 : m2] \\
 \frac{d[m4]}{dt} &= k_4[m2^*][m4_{mRNA}]
 \end{aligned} \tag{3}$$

with the following kinetic rate constants:

Kinetic Rates	
$k_1$	$10nM^{-1}s^{-1}$
$k_2$	$0.001s^{-1}$
$k_3$	$0.1s^{-1}$
$k_4$	$1000nM^{-1}s^{-1}$

We coded the system of ordinary differential equations (3) in MATLAB and solved them using the ode45 method provided by the software. The model was also encoded into SBML, using the CellDesigner software v3.5.1 [18]. The SBML code was used to initialize our simulator.

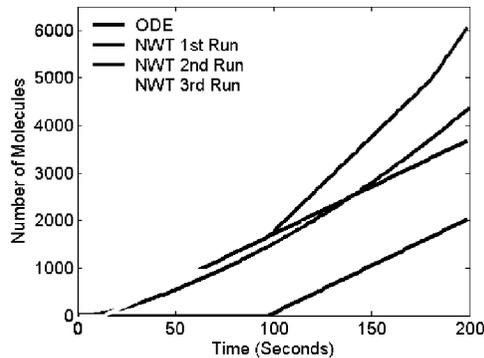


Fig. 2. The graph clearly shows the distinct evolutionary paths of the three separate NWT simulation runs (red, blue and cyan). The nondeterministic results are significantly distinct from the deterministic ODE method (black).

Results of three nondeterministic simulations are provided in Fig. 2 as well as the deterministic results of the ODE simulation. The ramifications of our nondeterministic approach are clear, as each run resulted in different multiplicities of  $m4$ .

The ODE simulation, allowing the use of fractional molecules, continuously allowed  $m1$  to bind to  $m2$  throughout the entire run. In contrast, our simulator accurately depicted the competition between  $m1$  and  $m3$  for the binding of the low number of  $m2$  molecules.

The above example is used to illuminate the difference between our NWT algorithm and an ordinary differential equations method. In the next section, we provide a comparative analysis between our approach and an ODE approach for simulation of the Fas-mediated apoptotic pathway

## 5. The Fas-mediated Apoptotic Pathway

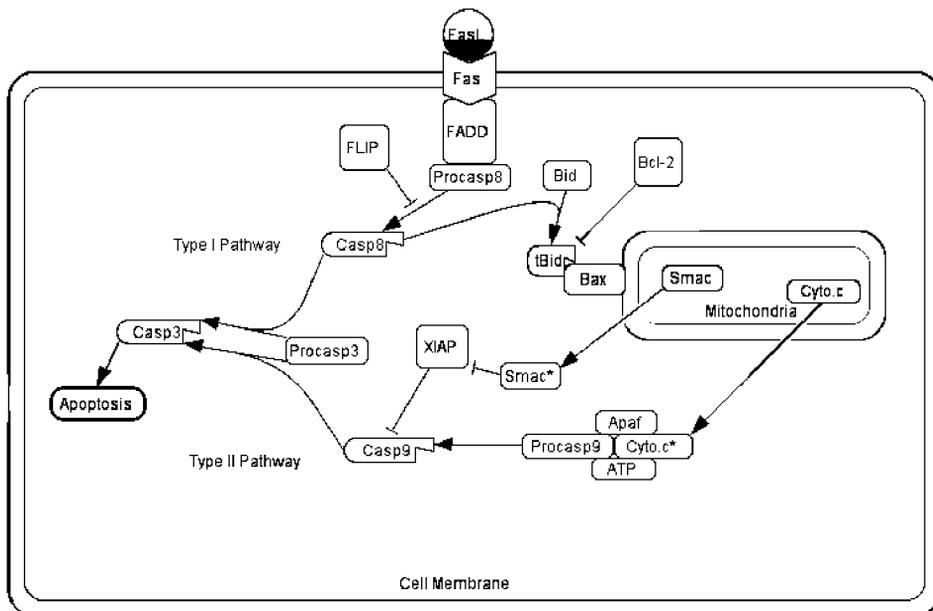


Fig. 3. Picture of the Fas-mediated apoptotic signal cascade. The pathway begins with the Fas ligand binding to the receptor site on the cell surface, and ends with the cleavage of Caspase-3 into active form. Both the type I and type II pathways are seen here. The former shows Caspase-8\* directly cleaving Caspase-3, while the latter requires amplification of signal through the mitochondria.

The pathway is illustrated in Fig. 3, and we provide the following description of the signal cascade

The pathway begins when a ligand, FasL, binds to a transmembrane receptor, Fas. When this occurs, a conformational change takes place in the receptor producing the complex Fasc. The cytoplasmic domain of this complex recruits Fas-associated death domain (FADD), with a maximum of three FADD per binding site (trimer). While FADD is bound to the complex Fasc, Caspase-8 and Flip are recruited competitively. Once at least two molecules of Caspase-8 have been recruited to a binding site, a dimer,  $\text{Caspas-8}_2^{P_{41}}$ , is released into the cytoplasm, and is then phosphorylated into active form (Caspase-8\*). The binding of FLIP to the

Fas complex is considered to be an inhibitor of apoptosis, because it decreases the number of sites available for Caspase-8 recruitment.

Unless sufficiently inhibited, the signaling cascade can continue in two different ways – the type I or type II pathway. If the initial concentration of Caspase-8 is large enough, Caspase-3 will be directly phosphorylated by the Caspase-8\* (the type I pathway). Otherwise, Caspase-8\* can truncate molecules of Bid (tBid). Each tBid molecule binds with two Bax molecules, which leads to the release of Cytochrome c from the mitochondria (type II pathway). Once released, Cytochrome c binds to Apaf and ATP, forming a complex that can recruit and phosphorylate Caspase-9 (Caspase-9\*). Caspase-9\* continues the cascade through the phosphorylation of Caspase-3. We consider the activation of Caspase-3 to be the end of the signaling cascade. Hence, the cell is dead once all of the Caspase-3 molecules are activated.

Besides Flip, there are other inhibiting factors at play: Bcl-2 hinders the release of Cytochrome c from the mitochondria and XIAP blocks Caspase-9\* from binding with Caspase-3. In other words, if sufficient levels of FLIP, Bcl-2, and/or XIAP exist, the apoptotic pathway can be blocked, and the cell lives.

The pathway described above was modeled with 101 different rules working on 53 distinct proteins and protein complexes. Fei Hua et al., in [9], provided the results of an ODE simulation, as well as some experimental data (from the Jurkat cell line) which they used to fit their model. We compared our results with the results from [9], simulating the same 101 rules and same initial conditions as the ODE simulation.

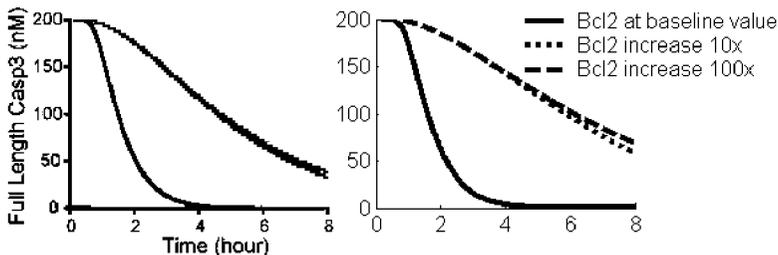


Fig. 4. On the right is the Membrane System and on the left is the ODE simulation (used with permission [9]). The two graphs show the decline over time of full length Caspase-3, which means there is an increase in active Caspase-3. At baseline Bcl-2 concentration, full length Caspase-3 approaches zero approximately four hours into the simulation. However, apoptosis is significantly inhibited as Bcl-2 levels are increased by 10-fold and 100-fold.

### 5.1. Results of discrete method

we simulated three different initial concentrations of Bcl-2: the baseline value (75nMs), an increase by 10-fold (750nMs), and an increase by 100-fold (7500nMs). Assuming a cell volume of  $10^{-12}$  liters, we converted the concentrations into molecular multiplicities: baseline value (45166 molecules), 10-fold (451660 molecules), and 100-fold (4516606 molecules). We expected to see a decline in Caspase-3 activation as Bcl-2 concentration was increased by 10-fold and 100-fold;

we provide the results of our simulations in (Fig. 4). We also provide the results of simulations with a decrease of 10-fold and 100-fold in comparison to the baseline Bcl-2 multiplicity (Fig. 5). Notice, the graphs based on our Membrane System simulations are comparable to the ODE results from [9].

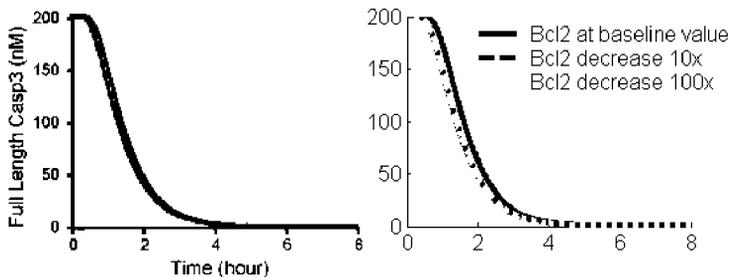


Fig. 5. On the right is the Membrane System and on the left is the ODE simulation (used with permission [9]). These results illustrate how the type I pathway is insensitive to decreases of Bcl-2 concentrations by 10-fold and 100-fold. We see agreement between the two different simulators.

### 5.2. Bcl-2's effects on the Type II pathway

Next, we analyzed the Caspase-3 activation kinetics by considering different mechanisms through which Bcl-2 can block the type II pathway. In [16], [14], and [3] the authors suggested that Bcl-2 might bind with (a) Bax, (b) Bid, (c) tBid, or (d) both Bax and tBid to block the mitochondrial pathway. We implemented four different sets of rules to test each Bcl-2 binding mechanisms. We refer the interested reader to [9] for the details of the rules.

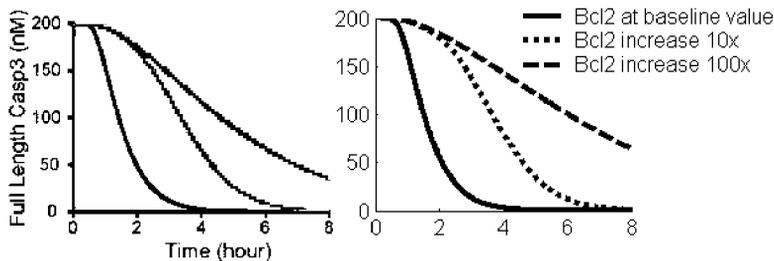


Fig. 6. On the right is the Membrane System and on the left is the ODE simulation (used with permission [9]). Here we see the effects of Bcl-2 binding to Bax only. It seems that Bcl-2 binding only with Bax is the second most effective method for blocking the apoptotic pathway.

The dynamics of Caspase-3 activation were studied by increasing the baseline Bcl-2 concentration by 10-fold and 100-fold. The conclusion of [9] is that Bcl-2 binding to both Bax and tBid (d) is the most efficient mechanism for inhibiting apoptosis. Our Membrane System agrees with the observations. The results of (d) are illustrated in Fig. 4, and (a) - (c) can be seen in Fig. 6 - Fig. 8. A comparison of (a) - (d) at baseline Bcl-2 concentration is shown in Fig. 9.

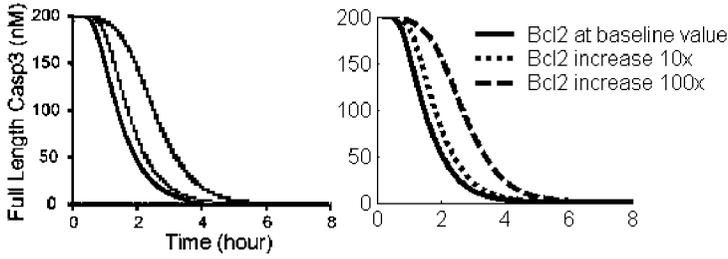


Fig. 7. On the right is the Membrane System and on the left is the ODE simulation (used with permission [9]). These two graphs show Bcl-2 binding to tBid only. We see considerably less inhibition of the pathway in this mechanism. The release of Cytochrome c is contingent on the binding of one tBid to two Bax molecules, and thus, blocking Bax would be more effective than blocking tBid.

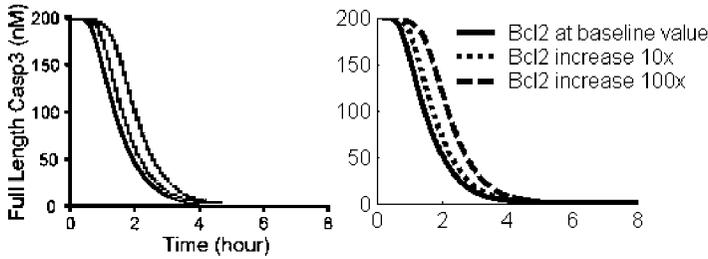


Fig. 8. On the right is the Membrane System and on the left is the ODE simulation (used with permission [9]). There is very little change in apoptotic behavior when Bcl-2 is allowed to bind only with Bid. This is because Bcl-2 binding to Bid is a reversible reaction, and once Bid is truncated to tBid, it is no longer available to bind with Bcl-2.

### 5.3. Modeling the behavior of the Type I pathway

Some cells are not sensitive to Bcl-2 over expression, as described in [10]. In these cells, Caspase-3 is activated through the type I pathway, bypassing the role of the mitochondria and Bcl-2. Scaffidi et al. have suggested in [11] that the type of pathway is chosen based on the concentration of Caspase-8 generated in active form following the binding of Fas ligand to its receptor site. High concentration of active Caspase-8 allows for direct activation of Caspase-3 (type I), but if the concentration of Caspase-8 is sufficiently low, amplification of the death signal through the mitochondria is required to induce cell death (type II). We tested this hypothesis by increasing the initial concentration of Caspase-8 by 20-fold (from 33.33nMs to 666.6nMs), which was expected to lead to increased active Caspase-8 throughout the simulation run.

We ran two different version of the increased Caspase-8 model, using the baseline concentration of Bcl-2 and an increase of Bcl-2 by 100-fold, in order to gauge the sensitivity of the type I pathway to Bcl-2 upregulation. Fig. 10 shows that Caspase-3 activation was not sensitive to the increase in Bcl-2 concentration, which is the hallmark for type I pathway dominant behavior. N.B., for these simulations Bcl-2 was allowed to bind to both Bax and tBid, which was shown above to be the most efficient mechanism for Bcl-2 inhibition of apoptosis.

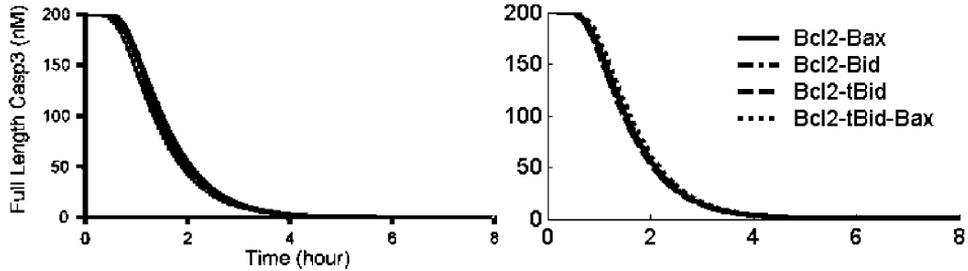


Fig. 9. On the right is the Membrane System and on the left is the ODE simulation (used with permission [9]). At baseline Bcl-2 concentration, each of the four mechanisms for Bcl-2 inhibition are similar. Both simulators agree across all rulesets: Bcl-2 binding with Bax only, Bid only, tBid only, or Bax and tBid.

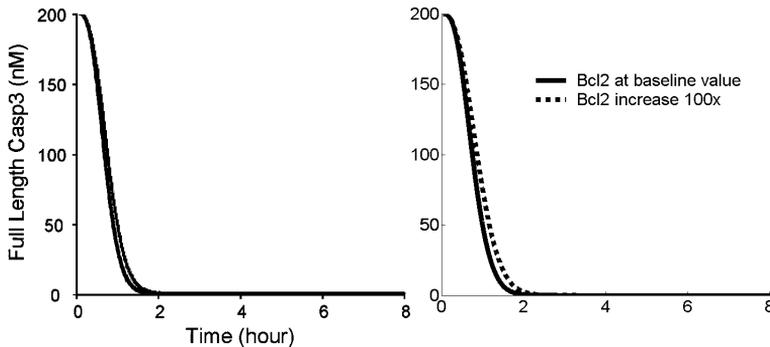


Fig. 10. On the right is the Membrane System and on the left is the ODE simulation (used with permission [9]). Unlike the type II pathway, the type I pathway is unaffected by a 100-fold Bcl-2 increase. The results of our simulations show that increasing Bcl-2 does not affect the type I pathway.

## 6. Discussion And Final Remarks

We have provided a discrete nondeterministic method for modeling molecular signaling cascades, highlighting the changes from our previous technique. Using a small example signaling cascade, we were able to demonstrate the effectiveness of our simulator for modeling low molecular multiplicity systems. We also gave our results from simulating the Fas-mediated apoptotic pathway.

Our Membrane System has yielded results comparable to an ordinary differential equations technique. The sixteen distinct simulations show similar apoptotic behavior to the ODE method (Fig. 4 - Fig. 10). However, albeit the activation of Caspase-3 is similar between the two techniques, the molecular interactions throughout are different. We have compared the results of our simulator with the experimental results [9] the ODE results from the same paper, and the stochastic approach described [10]. The Caspase-3 results are as expected, but the activation of Caspase-8 raises our interest. Refer to Fig. 11 for a comparison.

The consistency between the framework and the experimental results of [9] validates our model. Our Nondeterministic Waiting Time algorithm shows that Membrane Systems are an intriguing alternative to ordinary differential equations

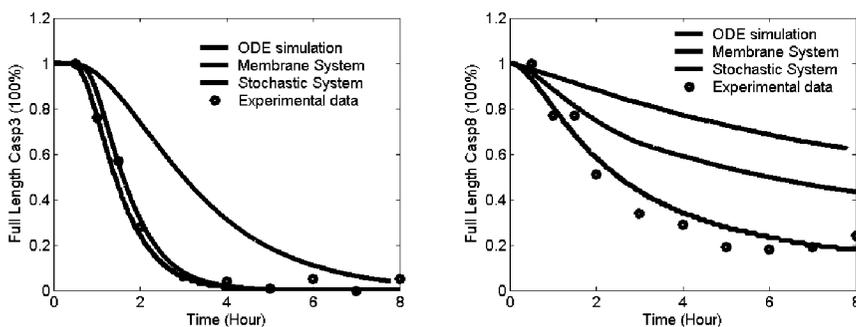


Fig. 11. The experimental data and ODE simulation results were obtained from Fei Hua et al. We see that the decrease of full length Caspase-3 is similar in all three simulation results. Interestingly, the decline of full length Caspase-8 is less prominent in the two Membrane System simulations. The contrast could be a result of the discrete nature of the Membrane Systems. As for both results being different than the experimental data, we believe that further investigation of kinetic rates of the reactions will allow for better agreement between simulation and experimentation.

methods. We have argued that the discrete nature of our technique might be better for simulating the evolution of systems involving low numbers of molecules. In the future, we would like to add a stochastic element to the Nondeterministic Waiting Time algorithm by allowing the simulation module to choose stochastically between rules that are not tied.

Another future thrust of our group will be the extension and refinement of the pathway discussed here. We plan to model the behavior of the caspase-based apoptotic pathway in HIV-infected cells. The few proteins encoded in HIV appear to have severe consequences on Fas-induced apoptosis. The so-called 'latently' infected T cells are especially interesting in the eradication of the AIDS epidemic.

We will also be applying our model to signaling cascades other than the Fas-mediated apoptotic pathway. In particular, we are interested in modeling oscillatory dynamics, such as the p53 digital response described in

## Acknowledgements

We gratefully acknowledge support in part from a NSF GK-12 Ph.D. fellowship, support from NSF Grants CCF-0430945, CCF-0523572 and CCF-0524136, support from LA BoR RSC grant LEQSF (2004-07)-RD-A-23, support from INBRE Program of the NCCR (a division of NIH), support from CNCSIS grant RP-13, support from CNMP grant 11-56 /2007, support from Spanish Ministry of Science and Education (MEC) under project TIN2006-15595, and support from the Comunidad de Madrid (grant No. CCG07-UPM/TIC-0386 to the LIA research group).

## References

- B. Bonnotte, N. Favre, S. Reveneau, O. Micheau, N. Droin, C. Garrido, A. Fontana, B. Chauffert, E. Solary, and F. Martin, "Cancer cell sensitization to Fas-mediated apoptosis by sodium butyrate," *Cell Death and Differentiation*, **5:6** (1998) 480–487.

- J.A. Briggs, M.N. Simon, I. Gross, H.G. Krausslich, S.D. Fuller, V.M. Vogt, and M.C. Johnson, "The stoichiometry of Gag protein in HIV-1," *Nature Structural & Molecular Biology*, **11**:7 (2004) 672–675.
- E.H. Cheng, M.C. Wei, S. Weiler, R.A. Flavell, T.W. Mak, T. Lindsten, and S.J. Korsmeyer, "BCL-2, BCL-XL sequester BH3 domain-only molecules preventing BAX- and BAK-mediated mitochondrial apoptosis," *Molecular Cell*, **8**:3 (2001) 705–711.
- S. Cheruku, A. Păun, F. Romero-Campero, M. Pérez-Jiménez, and O. Ibarra, "Simulating FAS-Induced Apoptosis by Using P Systems" *Proceedings of Bio-inspired computing: theory and applications (BIC-TA)* September 18-22, 2006, Wuhan, China, also extended version submitted to *Progress in Natural Science*, **17**:4 (2006) 424–431.
- L. Conti et al., "The HIV-1 vpr Protein Acts as a Negative Regulator of Apoptosis in a Human Lymphoblastoid T Cell Line: Possible Implications for the Pathogenesis of AIDS," *Journal of Experimental Medicine*, **187**:3 (1998) 403–413.
- T.H. Finkel, G. Tudor-Williams, N.K. Banda, M.F. Cotton, T. Curiel, C. Monks, T.W. Baba, R.M. Ruprecht, and A. Kupfer, "Apoptosis occurs predominately in bystander cells and not in productively infected cells of HIV- and SIV-infected lymph nodes," *Nature Medicine*, **1**:2 (1995) 129–134.
- M.A. Gibson and J. Bruck, "Efficient Exact Stochastic Simulation of Chemical Systems with Many Species and Many Channels," *Journal of Physical Chemistry A*, **104**:9 (2000) 1876–1889.
- Y. Han, M. Wind-Rotolo, H.C. Yang, J.D. Siliciano, and R.F. Siliciano, "Experimental approaches to the study of HIV-1 latency. *Nature Reviews Microbiology*, **5**:2 (2007) 95–106.
- F. Hua, M.G. Cornejo, M.H. Cardone, C.L. Stokes, and D.A. Lauffenburger, "Effects of Bcl-2 Levels on Fas Signaling-Induced Caspase-3 Activation: Molecular Genetic Tests of Computational Model Predictions," *The Journal of Immunology*, **175**:2 (2005) 985–995 and correction **175**:9 (2005) 6235–6237.
- J. Karn, "Tackling Tat," *Journal of Molecular Biology* **293**:2 (1999) 235–254.
- J.F. Kerr, "A histochemical study of hypertrophy and ischaemic injury of rat liver with special reference to changes in lysosomes," **90**:2 (1965) 419–435.
- J.F. Kerr, A.H. Wyllie and A.R. Currie, "Apoptosis: a basic biological phenomenon with wide-ranging implications in tissue kinetics," *British Journal of Cancer*, **26**:4 (1972) 239–257.
- L. Ma, J. Wagner, J.J. Rice, W. Hu, A.J. Levine, and G.A. Stolovitzky, "A plausible model for the digital response of p53 to DNA damage," *Proceedings of the National Academy of Sciences*, **102**:40 (2005) 14266–14271.
- Z.N. Oltvai, C.L. Milliman and S.J. Korsmeyer, "Bcl-2 heterodimerizes in vivo with a conserved homolog, Bax, that accelerates programmed cell death," *Cell*, **74**:4 (1993) 609–619.
- C. Scaffidi, S. Fulda, A. Srinivasan, C. Friesen, F. Li, K.J. Tomaselli, K.M. Debatin, P.H. Kramer, and M.E. Peter, "Two CD95 (APO-1/Fas) signaling pathways," *The EMBO Journal*, **17**:6 (1998) 1675–1687.
- K. Wang, X.M. Yin, D.T. Chao, C.L. Milliman, S.J. Korsmeyer, "BID: a novel BH3 domain-only death agonist," *Genes & Development*, **10**:22 (1996) 2859–2869.
- N. Selliah and T.H. Finkel, "Biochemical mechanisms of HIV induced T cell apoptosis," *Cell Death & Differentiation*, **8**:2 (2001) 127–136.
- The Systems Biology Institute, "CellDesigner<sup>TM</sup>" [Software]. Retrieved August 17, 2007, from the CellDesigner Web site, <http://www.celldesigner.org>.

J. Wagner, L. Ma, J.J. Rice, W. Hu, A.J. Levine, and G.A. Stolovitzky, "p53-Mdm2 loop controlled by a balance of its feedback strength and effective dampening using ATM and delayed feedback," *Systems Biology, IEE Proceedings*, **152**:3 (2005) 109–118.

A. Zhang, Y. Wu, H.W. Lai, D.T. Yew, "Apoptosis - A Brief Review," *Neuroembryology and Aging* **3**:1 (2004) 47–59.