DiMSim: A Discrete-Event Simulator of Metabolic Networks

Xiao-Qin Xia and Michael J. Wise*

Department of Genetics, University of Cambridge, Cambridge CB2 3EH, United Kingdom

Received December 10, 2002

A novel, scalable, quantitative, discrete-event simulator of metabolic and more general reaction pathways—DiMSim—has been developed. Rather than being modeled by systems of differential equations, metabolic pathways are viewed as bipartite graphs consisting of metabolites and reactions, linked by unidirectional or bidirectional arcs, and fluxes of metabolites emerge as the product of flows of the metabolites through the individual reactions. If required, DiMSim is able to model reactions involving single molecules up to molar concentrations so it is able to cope with the special characteristics of biochemical systems, including reversible reactions and discontinuous behavior, e.g. due to competition between reactions for limited quantities of reactants, product or allosteric inhibition and highly nonlinear behavior, e.g. due to cascades. It is also able to model membrane-bound compartments and the channels used to transport metabolites between them (both passive diffusion and active transport). While Michaelis−Menten kinetics is supported, DiMSim makes almost no assumptions other than each reaction having a fixed stoichiometry and that each reaction takes a stated amount of time.

INTRODUCTION

Simulation is widely used to model the dynamics of systems and to predict system behaviors. At one level, cells of organisms can be viewed as mechanisms driven by metabolic pathways which are finely regulated by feedback mechanisms. Regulation is probably the most significant feature of metabolic pathways, compared to other chemical reactions, allowing the cells to maintain stability under a range of different situations (or *homeostasis*).

A considerable body of knowledge about metabolic pathways has been published over the years, and there now are a number of databases of enzymes, proteins, and pathways, such as KEGG,[9] EcoCyc,[11] MetaCyc,[10] and EMP.[19] However, such information is only static; the user cannot easily visualize the dynamics of a metabolic system. One way to study the dynamics of a system is to simulate it.

As mentioned in Sauro,[18] computer simulation of metabolic and chemical systems has a history almost as long as the history of the computer itself. Since the earliest simulations, which were done using analogue computers, there has been a stream of projects to model metabolic pathways. These efforts have resulted in a number of systems based on Ordinary Differential Equations (ODE). Recent examples are E-Cell,[21] Metatool,[15] Gepasi3,[13] MIST,[2] and SCAMP.[18] However, simulation systems based on differential equations face a number of problems when applied to this problem domain. Apart from the problem of scalability as the number of equations grows and the problem of "stiffness", there is the cost of determining the parameters for the equations and their sensitivity to change.[4] There is also a problem of interpretation by users. Systems of differential equations have a number of parameters that must be fitted from experimental data. However, the parameters may have no meaning to the biologists, who are therefore unable to gauge whether the values are appropriate. These factors led Raczynski to conclude that "the ODE models, perhaps good while describing the radioactive decay or electrical circuits with concentrated parameters, may not work in models as complex as the social systems and living organisms" and therefore that object-oriented, "behavioral" simulation can be the solution in many cases.[16]

Finally, "...the existing information on some of the metabolic pathways is not complete, for example there may be unknown catalyzing components, uncertainty about the role of known components, unreliable experimental data, etc.".[17] This analysis led Reddy et al. to employ Petri nets[14] as a tool for modeling and reasoning about metabolic pathways, e.g. to detect bottlenecks. In other words, the Petri net approach eschews simulation and instead deals with the analytic properties of the graphs. A very different approach, but one still aimed at reducing the impact of variable data quality, has been to use knowledge-based systems approaches, particularly qualitative reasoning, to model networks of metabolic reactions.[7] These analyses are unduly pessimistic. In this paper, we describe a scalable, quantitative, discrete-event simulation system based on modeling metabolic pathways as a network of reactions. The system requires only a few, well understood constants with which it is able to emulate in-silico reaction pathways which are close to or far from equilibrium.

THE BASIC MODEL: DATA-FLOW COMPUTATION

The data-flow computation model was widely investigated in the 1970s and 1980s and provides the basic intuitions underlying the current system. (For a review of the data-flow computation model see, for example, ref 22.) As a simple example to illustrate the application of data-flow computation in metabolic pathway simulations consider a compartment containing two reactions (Figure 1). There are

* Corresponding author e-mail: mw263@cam.ac.uk.

10.1021/ci025650w CCC: $25.00
Published on Web 03/20/2003

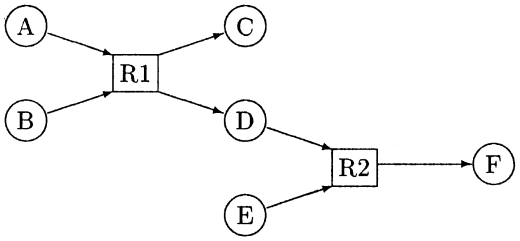

Figure 1. A simple pathway consisting of two successive reactions.

six reactants (substrates and products, represented as circles) and two reactions (R1 and R2), shown as rectangles.

Two time concepts are used in the simulations. The principal one, *simulation time*, is the time taken by the sequence of reactions followed by downstream reactions; *running time* is the real time spent by the computer as it undertakes the simulation. For example, assume that the enzymes required by the two reactions are present in suitable concentrations and the environmental conditions for the reactions are favorable. Assume also that at the outset there 60 000 molecules of A, 40 000 of B, and 30 000 of E, but no molecules of C, D, and F.

When the simulation is launched, the system searches for reactions which may fire, i.e., can take place. Clearly R1 may fire, since its substrates (A and B) are present while R2 may not fire, because one of its substrates, D, is not present in the compartment. R1 then begins to transform A and B into C and D. The numbers of products are determined by B, the substrate with the fewest molecules, so 40 000 molecules of C and 40 000 molecules of D are produced, while B is consumed completely, and A has 20 000 molecules remaining.

At the second time point, the system continues to search for fireable reactions. This time, R2 is found instead of R1. The reaction results in 30 000 molecules of F, while the number of E is reduced to zero and D has 10 000 molecules remaining. The system will not stop the cycle of searching for fireable reactions and triggering them until no reaction can fire or a specified running time or simulation time has elapsed.

Under this model, rather than being modeled directly, the flux of any metabolite is an emergent property, being the sum of molecules produced or consumed by reactions involving the metabolite. In addition, each reaction can be viewed as a primitive numerical integrator.

REFINEMENTS

The rather simplistic model provided by data-flow computation fails to reflect biological realities in a number of important ways and is therefore amended in DiMSim.

1. Reversible Reactions and Irreversible Reactions. While some reactions can be treated as being one-way—specifically those which involve energy expenditure—most metabolic reactions are reversible. For example, in the first three steps of the glycolysis pathway

$$\mathrm{ATP} + \mathrm{G} \rightarrow \mathrm{ADP} + \mathrm{G6P}$$

$$\mathrm{G6P} \leftrightarrow \mathrm{F6P}$$

$$\mathrm{ATP} + \mathrm{F6P} \rightarrow \mathrm{ADP} + \mathrm{FBP}$$

where G, G6P, F6P, and FBP are respectively D-glucose, D-glucose 6-phosphate, D-fructose 6-phosphate, and D-fructose 1,6-bisphosphate in brief. Which direction a reversible reaction will go depends on the equilibrium constant (K_{eq}) and the value $\Pi[P]/\Pi[S]$, in which $[P]$ and $[S]$ are the concentrations of products and substrates at equilibrium. [In a compartment of known size, $[x]$ is equivalent to the number of molecules of x, so concentration and number of molecules are here used interchangeably.] When the ratio of the concentrations equals K_{eq} the reaction is in equilibrium, and there is no net reaction. However, if at any time the value $\Pi[P]/\Pi[S]$ is less than K_{eq}, the reaction will go forward; the reaction go in the reverse direction when $\Pi[P]/\Pi[S]$ is larger than K_{eq}. In the case of irreversible reactions, the reaction will go in the forward direction if the ratio is less than K_{eq} but will not proceed in the reverse direction even if the ratio is greater than K_{eq}.

Having determined the direction of a reaction, the quantities of molecules to be produced in the reaction are determined by the minimum input metabolite count (so B in the example above) and by the need to restore the balance of the substrates versus products corresponding to K_{eq}. Addressing the second requirement involves solving the equation

$$\frac{\Pi([P] + \Delta P)}{\Pi([S] - \Delta S)} = K_{eq} \qquad (1)$$

where ΔP is the increase in the product and ΔS is the corresponding decrease in the substrate that are required to restore equilibrium. This equation can be easily converted to a polynomial form, so it can be calculated by formulas (when the power is no more than three) or numerical methods such as Newton−Raphson method and binary chop method, both of which are used in the system. GMP (the GNU multiprecision library) can be used to ensure accurate results.

2. Fireable Reactions. Whether a reaction can fire or not depends on the state of its reactants and enzymes. First of all, all the substrates and enzyme must be available. For reasons of computational efficiency, it is possible for the user to set a minimum number input molecules for each reaction; if insufficient are present, the reaction will not be launched. However, even if all the reactants are present, their concentrations must meet thermodynamics requirements. In other words, for a net reaction to proceed the reactants must not be in equilibrium, i.e., the value $\Pi[P]/\Pi[S]$ must less than K_{eq} for an irreversible reaction or not equal to K_{eq} for a reversible reaction. In addition, for reasons of computational efficiency, the user can set a value, ϵ so that if the value $\Pi[P]/\Pi[S]$ lies in the range $[K_{eq}-\epsilon, K_{eq}+\epsilon]$, the reaction is considered in equilibrium and will not fire.

3. Inhibition and Activation. Regulation and feedback are significant features of metabolic pathways. One important regulatory mechanism is inhibition, which falls into two main types: reversible inhibition and irreversible inhibition. Reversible inhibition has several subtypes: competitive, noncompetitive, mixed, and uncompetitive. All the inhibition are considered as pure inhibition, i.e., inhibitors are able to inactivate enzymes completely.

A competitive inhibitor competes for an enzyme with a substrate; it does not influence enzyme by reducing its the turnover number k_{tnv}, which is defined as the maximum number of substrate molecules that an enzyme molecule can catalyze in 1 s. [k_{tnv}, labeled MA—or Molecular Activity—in

Table 1. Effect of Inhibitors on Apparent K_m and Apparent k_{tnv}[a]

inhibition type	K_m^{app}	k_{tnv}^{app}
noncompetitive	K_m	$k_{tnv}/(1 + [I]/K_{IU})$
mixed	$K_m(1 + [I]/K_{IC})/(1 + [I]/K_{IU})$	$k_{tnv}/(1 + [I]/K_{IU})$
uncompetitive	$K_m/(1 + [I]/K_{IU})$	$k_{tnv}/(1 + [I]/K_{IU})$

[a] K_{IU} is defined as the enzyme−substrate-complex:inhibitor dissociation constant $K_{IU} = [ES][I]/[ESI]$, while K_{IC} is the enzyme:inhibitor dissociation constant $K_{IC} = [E][I]/[EI]$.

the EMP database, may be contrasted with k_{cat}, which is labeled as KC, or Catalytic Constant. The latter is the maximum number of substrate molecules catalyzed per second per active site of an enzyme molecule. k_{tnv} and k_{cat} are generally the same, but may differ, e.g. phosphofructokinase from *D. discoideum*, where k_{tnv} is 4 times k_{cat}.] Rather, the enzyme molecules are distributed between the substrate and the inhibitor. In other words, we can simulate competitive inhibition simply by reducing the number of enzyme molecules available for the substrate, which depends on the number of substrate molecules and their binding affinity with enzyme molecules. One indirect measure of the binding affinity of a substrate with an enzyme is the inverse of the Michaelis constant, K_m. The inhibition constant, K_I, plays an analogous role to K_m in the competition between the normal substrates of a reaction and competitive inhibitors. Therefore, if $[I]$ is the number of molecules of an inhibitor which competes with substrate S for enzyme E, the apparent number of enzyme molecules available to substrate S is $[E]^{app} = [E] \times [S]/K_m/[S]/K_m + [I]/K_I$.

Noncompetitive inhibition, mixed inhibition and uncompetitive inhibition do not involve competition for the enzyme between the substrates and the inhibitor but instead change the apparent Michaelis constant and/or the apparent maximum velocity of enzymes.[6] The relationships are summarized in Table 1.

It is also possible to simulate compulsory activation, in which the free enzyme without an activator bound to it does not bind substrate. Correspondingly, the activation constant K_A, the dissociation constant between activator and enzyme, is used to determine the apparent K_m:$K_m^{app} = K_m*(1+K_A/[A])$, in which $[A]$ is the concentration of the activator.

4. Reaction Velocity. Tracking the velocity of a reaction is perhaps the most important and challenging task in metabolic simulation. While the well-known Michaelis−Menten equation

$$v = \frac{V_{max} \times [S]}{[S] + K_m} = \frac{V_{max}}{1 + \frac{K_m}{[S]}} \tag{2}$$

provides a simplified and idealized model for the dynamic behavior of an enzymatic reaction, enzymatic reaction velocity can be influenced by a number of factors, such as reactants, coenzymes, ions, pH, temperature, and activators/inhibitors. Secondary mechanisms may also play roles in determining reaction velocity, e.g. the order in which different substrates bind with an enzyme, substrate protection, and substrate/product inhibition.[1] Furthermore these factors will inevitably interact and result in complicated behaviors. It is not feasible to include all possible factors in a single rate equation so most rate equations encompass only a few factors and are subject to various conditions. Typically, an equation for a multisubstrate reaction is bound to a particular reaction mechanism, such as compulsory-order mechanism, random-order mechanism, double-displacement mechanism, etc.,[1] and requires a series of rate constants that are usually not available in current databases or from the literature. Furthermore, the available rate equations, which are usually derived theoretically or deduced in vitro, may not fit the cellular environment well because the cell is a far more crowded environment than a solution and molecular crowding has a complex effect on reaction rates in a cell.[3] Therefore, to make simulations possible we have created a rate equation that considers some of the most common factors and uses relatively few parameters. The equation is necessarily incomplete. We begin with an equation for a reaction involving a single substrate and a single product. Since most biochemical reactions are reversible, product inhibition is likely to be of the competitive type, so the rate equation can be written as[5]

$$v = \frac{(V/K_{m,S}) \times ([S] - [P]/K_{eq})}{1 + [S]/K_{m,S} + [P]/K_{m,P}} \tag{3}$$

in which $K_{m,S}$ and $K_{m,P}$ are respectively the Michaelis constant of enzyme with substrate and product. V is the limiting rate in the forward direction − either the maximum velocity ($V_{max} = [E] \times k_{tnv}$) or the apparent maximum velocity ($V_{max}^{app} = [E]^{app} \times k_{tnv}^{app}$) if there are modifiers, typically inhibitors.

Equation 3 can be rewritten as

$$v = V \times \left(1 - \frac{[P]/[S]}{K_{eq}}\right) \times \frac{[S]/K_{m,S}}{1 + [S]/K_{m,S} + [P]/K_{m,P}} \tag{4}$$

Equation 4 is the product of three terms: V is the actual or apparent maximum velocity; the second term, $(1 - [P]/[S]/K_{eq})$, represents the distance of the reaction from equilibrium, and can be replaced by $(1 - \Pi[P]/\Pi[S]/K_{eq})$ for a reaction with more than one substrate or product. The last term $[S]/K_{m,S}/1 + [S]/K_{m,S} + [P]/K_{m,P}$, may be roughly viewed as the fractional saturation or concentration effects of reactants on the velocity of the reaction or the fraction of enzyme bound by substrates. Its extension to multiple substrates and products is $\sum_i([S]/K_{m,S})/1 + \sum_i([S]/K_{m,S}) + \sum_j([P]/K_{m,P})$. If the reaction is irreversible, the term $\sum_j([P]/K_{m,P})$ can be omitted. In the presence of modifiers, the rate equation becomes

$$v = V_{max}^{app} \times \left(1 - \frac{\prod[P]/\prod[S]}{K_{eq}}\right) \times \frac{\sum_i([S]/K_{m,S}^{app})}{1 + \sum_i([S]/K_{m,S}^{app}) + \sum_j([P]/K_{m,P}^{app})} \tag{5}$$

5. Allosteric and Cooperative Effects. Regulatory enzymes are usually allosteric proteins, which generate sigmoid dynamic behaviors. The sigmoid curve is due to the cooperative effects of substrates binding with the enzyme, which means that binding substrate molecules facilitates further substrate molecules being bound. The Hill equation

addresses this behavior. The fractional saturation, i.e., the proportion of enzyme bound with substrate S is

$$\bar{Y}_s = \frac{[S]^H}{K_{0.5}^H + [S]^H} = \frac{1}{1 + \left(\frac{K_{0.5}}{[S]}\right)^H} \quad (6)$$

in which H is the Hill constant and $K_{0.5}$ is the half saturation concentration (known as SO5 in EMP database). Like the Michaelis constant, K_m, $K_{0.5}$ is the value of the substrate concentration at which $v = 0.5V$. If $H = 1$, $K_{0.5}$ is equal to Michaelis constant. The rate equation can be written as

$$v = V_{max}^{app} \times \bar{Y}_s = \frac{V_{max}^{app}}{1 + \left(\frac{K_{0.5}}{[S]}\right)^H} \quad (7)$$

For an allosteric inhibitor I, the apparent maximum velocity is proportional to the enzyme that remains unbound, i.e.

$$V_{max}^{app} = V_{max} \times (1 - \bar{Y}_s) = \frac{V_{max} \times K_{0.5}^H}{K_{0.5}^H + [I]^H} = \frac{V_{max}}{1 + \left(\frac{[I]}{K_{0.5}}\right)^H} \quad (8)$$

where the $K_{0.5}$ is actually $I_{0.5}$, the half inhibiting concentration.

To some degree the ratio between concentration and $K_{0.5}$ reflects the state of fractional saturation of an enzyme by a substrate or inhibitor. By comparing eq 7 with eqs 2 and 8 with the equations in Table 1 it is evident that the rate equations can model cooperative effects if the ratios are raised to the power H. So a simple way to address cooperative effects is to replace all the terms of the ratios of concentration value and K_m or K_I with the same concentration value and $K_{0.5}$, raised to power H. However, for reasons of availability of data, K_m and K_I may still be used as a substitute for $K_{0.5}$.

6. Competition between Reactions for Inputs. Equation 5 together with the equations for V_{max} or V_{max}^{app} reveal that concentration of enzyme, k_{tnv}, K_m, and many other factors have an impact on reaction velocity, reflecting the ability of a reaction to garner substrates and to make products. Where a metabolite acts as the common substrate for more than one reaction, competing reactions share the common metabolite in proportion to their velocities. The proportion of the substrate available for the ith reaction is $v_i/\sum_j v_j$.

ALGORITHMS AND OBJECTS

1. Basic Algorithm. As discussed above in the data-flow computation model, the computation is organized as cycles of searching for fireable reactions and triggering them. However, unlike the atemporal data-flow computation model, a reaction takes a finite amount of time which will be called its transition time, $T_t = 1/k_{cat}$, which is the time required by a reaction at a given active site to transform a set of substrates into a set of products. Furthermore, although the DiMSim model is able to deal with single molecules a reaction usually involves a large number of molecules, bearing in mind that 1 M concentration of any substance in a 1-L compartment contains Avogadro's number (6.0221×10^{23}) of molecules. Therefore, while all the reactions in a real system such as a cell are taking place simultaneously and the state variables change continuously, it is assumed in the simulation that all the reactions that are fireable will proceed in parallel, and in particular that all the transformations for given reaction inputs will proceed lock-step. The justification is that, while in reality some molecules will arrive sooner and are processed sooner and some will arrive later, the transition time represents the expected value of a probability distribution. In other words, a single cycle for a given reaction will take T_t.

Fireable reactions are placed in a min-heap[20] ordered by increasing completion time, i.e., the reaction at the top of the heap is the next to complete. The processes can be written in pseudocode as follows:

```
// find all reactions which are able to fire
// and store them in a min_heap
def AddReactions(h, rs, t)
    for each r in rs do
        if r is able to fire then
            calculate the transition time T_t
            complete time C_t = t + T_t
            store (C_t, r) in h
        endif
    endfor
h = new min_heap
rs = all reactions
t = 0
max_t = max simulation time set by user
AddReactions(h, rs, t)
// trigger reactions in the min_heap
while h is not empty do
    t, r = the item at the top of h
    remove the top item from h
    ms = substrates and products of r
    // calculate st, the number of substrate
    // molecules that will be consumed
    v = velocity of r
    max_x = v * T_t
    avail_x = number of metabolites available for r
    delta_x = value calculated from equation (1)
    delta_x = min(max_x, avail_x, delta_x)
    adjust concentrations of ms according to delta_x
    if t >= max_t then
        break
    rs = reactions related to ms
    AddReactions(h, rs, t)
endwhile
```

So if a reaction is always fireable, the transition time is also the interval for the reaction to be triggered. This may cause an apparent waste of catalytic ability, i.e., if a reaction has a relatively long transition time, its substrates may accumulate to an improbably high concentration during the long interval even though there is sufficient enzyme available for the reaction to fire. On the other hand, we have observed that in these situations it is also the case that the reactions will be near equilibrium so the numbers of molecules required to restore equilibrium will be small. The system has therefore been altered so as to allow the user to set a maximum transition time (MT); reactions with a transition

time longer than MT will use MT as their transition time and will produce proportionally fewer product molecules but will then be available to process another tranche. Viewed another way, if each reaction is seen as a primitive numerical integrator, specifying a maximum transition time has the effect of setting a maximum step size.

2. Simulation of Channels. Reactions occur in compartments that are surrounded by membranes so channels exist to permit flows of specific metabolites across the otherwise impermeable membranes. There are three types of channel models, corresponding to the three types of transport in organisms: passive diffusion, facilitated transport, and active transport. These are treated in two ways. If a channel is driven by transporter protein, it is dealt with the same way as a common enzymatic reaction. If no transporter protein is used, the process is viewed as diffusion and a permeability coefficient P with unit *cm*/*s* is needed. The velocity of a diffusion is calculated as $v = P \times A \times ([S]-[P])$, where A is the surface area of the compartment membrane across which the channel is located, and $[S]$ and $[P]$ are, as usual, the concentrations of substrate and product.

The thickness of membrane divided by the permeability coefficient is used to calculate T_t. Diffusion-type channels may compete for substrates with other reactions so another parameter, half-saturation substrate concentration, is used as the equivalent of K_m.

3. Objects in the Model. Abstracting from the biology, there are two material concepts and two process concepts involved in the metabolic pathways of living cells. The material concepts include *metabolites* (the molecules involved in reactions) and *compartments* (the membrane-bound spaces containing molecules, starting with the cell membranes). The process concepts are *reactions*, which transform metabolites, and *channels*, which exist in membranes to facilitate the transfer of metabolites between compartments, sometimes with the assistance of enzymes or other proteins. Mirroring these abstractions, the classes involved in the system fall into four types: metabolite, compartment, reaction, and channel.

Metabolite. This class is used to represent any kind of molecule, so the attributes include molecule number/concentration, the compartment in which it is located, and dictionaries containing references to the reactions particular instances may be involved in. The typical access to an object of this class is through the method to change its number/concentration. Metabolites include not only substrates and products but also enzymes and which may also be the substrates or products of metabolic pathways. However, because the activity of enzymes can be modified by inhibitors, both enzyme objects and inhibitor objects can contain references to other metabolite objects.

Compartment. This class is a container for lists of metabolites, reactions, channels, and subcompartments which are located in this compartment. It also offers a place for reactions to occur, so volume is a basic attribute of a compartment object. This object also offers methods and logic to organize the reactions and launch a simulation. As an aid to the modeling process, a complex pathway in one compartment may also be divided into virtual units called subsystems. There is no physical boundary between a subsystem instance and its enclosing compartment, and nearly all the methods and many of the parameters of the enclosing compartment are available to the subsystem. The purpose of subsystems is to allow users to hide the details of whole subnetworks, e.g. TCA cycle, and thus treat them as if they are a single "reaction".

Reaction. This class defines the basic parameters and behaviors of a biochemical reaction. Its attributes consist of reaction type (reversible/irreversible), equilibrium constant, and some biochemical reaction constants and dictionaries with references to its enzymes, substrates, products, and other metabolites. A basic method decides how to transform the molecule numbers of its reactants. In essence, a reaction never moves metabolite molecules; it can only reduce the molecule numbers/concentrations of some metabolites and correspondingly increase the numbers/concentrations of some other metabolite object, all of which are located in the same compartment as the reaction object. The behavior of a reaction is determined by both its own parameters and the properties of its enzymes whose activities are regulated by other environmental factors, mainly inhibitors.

Channel. This class, derived from the reaction class, is modified to deal with metabolites in different (usually adjacent) compartments. It is generally used to transfer metabolites from one compartment to another compartment by changing the molecule numbers/concentrations in the participating compartments. If no product is specified or K_{eq} is set by the user to zero, a channel will act as a sink reaction, which simply fires in the forward direction without considering whether its substrates and products are in equilibrium or not. Sink reactions are useful at the boundaries of a simulation to model the flow of products out of the system under investigation.

FEATURES OF THE SYSTEM

Figure 2 is a screen capture of the graphical user interface. Every object involved in a pathway is shown as an icon in the graph. Relationships among objects are represented by arrows. Different colors are applied to arrows to express different relationships, i.e., black arrows start from substrates, violet arrows point to products, blue arrows are from enzymes to reactions, and red arrows from inhibitors to enzymes. Unidirectional arrows are used for irreversible reactions, while bidirectional arrows mean reactions are reversible.

To simulate a metabolic pathway in DiMSim, the tools offered in the tool-bar are used to generate all the objects involved in the pathway and to define the relationships among these objects. Then some necessary attributes of each object, for example, concentration and turnover number, have to be set through a pop up menu appeared after double-clicking the mouse on the object icon. The simulation can then be started.

Figure 2 depicts a number of interconnected pathways including the Glycolytic Pathway subsystem and four compartments, i.e., C1, C2, C3, and the outermost compartment(OC) that contains all other objects. M1 is transferred from C1, to C2 by Channel_1, and to OC by Channel_2, then to C3. G (Glucose) is then transferred from C3 to OC and then used by glycolytic pathway which is enclosed as a subsystem. The two products of glycolytic pathway, ATP and Pyr, are eventually moved into C2 by two sinks (Sink_1 and Sink_2).

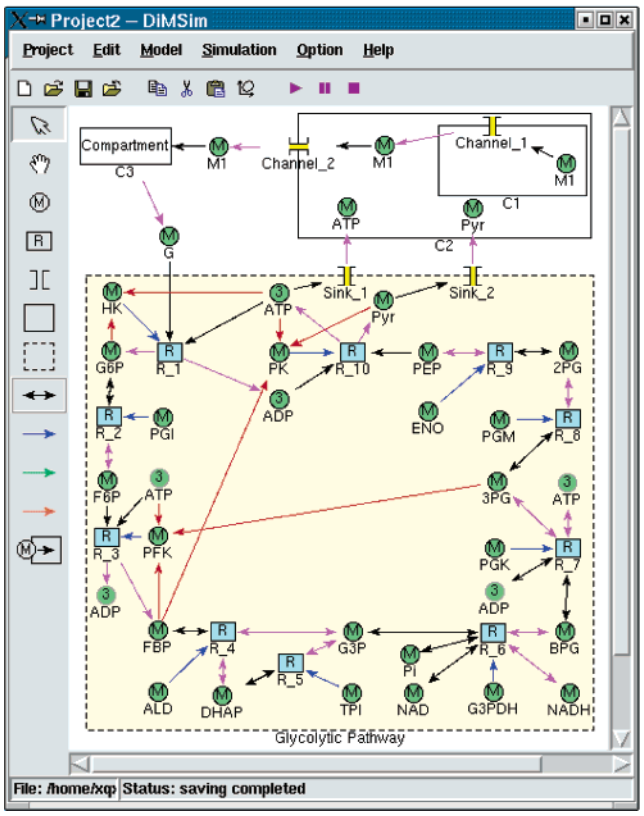

Figure 2. An example of the user interface.

DiMSim has features to simplify the construction of simulations. In particular, a compartment/subsystem can be iconified if a detailed view is not required (see Compartment_3 in Figure 2). In addition, a simulation run involving a pathway in a compartment/subsystem can be started separately. Finally, one can modify the structure of the graph even when a simulation is being carried out and view the effects of the modification.

EXAMPLES

To show the performance of DiMSim, some examples are given below. These experiments are carried out on a PentiumIII PC (1 GHz clock, 512MB RAM.)

1. Michaelis−Menten Dynamics. By setting suitable values, DiMSim can model a typical Michaelis−Menten dynamics behavior described by eq 2. For a simple reaction S → P, the initial concentrations of S, P, and the enzyme are 1.0 mM, 0 mM, and 0.001 mM, K_m is 0.1 mM, K_{eq} is 10^{23}, and both k_{tnv} and k_{cat} are 1000/min. The dynamics of velocity vs concentration of substrate is shown by the first curve named Michaelis in Figure 3.

2. Simple Competitive Behavior. The dynamics of reversible reactions can be illustrated by a pathway consists of two reactions: R1 (A ↔ B) and R2 (A ↔ C). Assume the initial concentrations of A, B, and C are 1 mM, 0 mM, and 0.5 mM, while the enzymes are each 0.1 mM; the constants for the both reactions and their enzymes are K_{eq} 1.0, K_m 0.1 mM, k_{tnv} 1000/min, and k_{cat} 1000/min. The reactions reach equilibrium state after in about 10 s. For this simple pathway, at the beginning, both R1 and R2 will go forward, this causes [A] to drop quickly while [B] and [C] are increasing. Since R1 and R2 have the same parameters and [C] has a initial value higher than [B], [A] approaches [C] first. When [A] decreases to a value equal to [C], R2 will stop. But R1 keeps consuming A, and [A] drops below [C], this causes R2 to reverse and [C] decreases. This state lasts till the system falls into equilibrium, i.e., [A], [B], and [C] converge to 0.5 mM. This dynamical behavior is perfectly reproduced by DiMSim (see curves named A, B, and C in Figure 3).

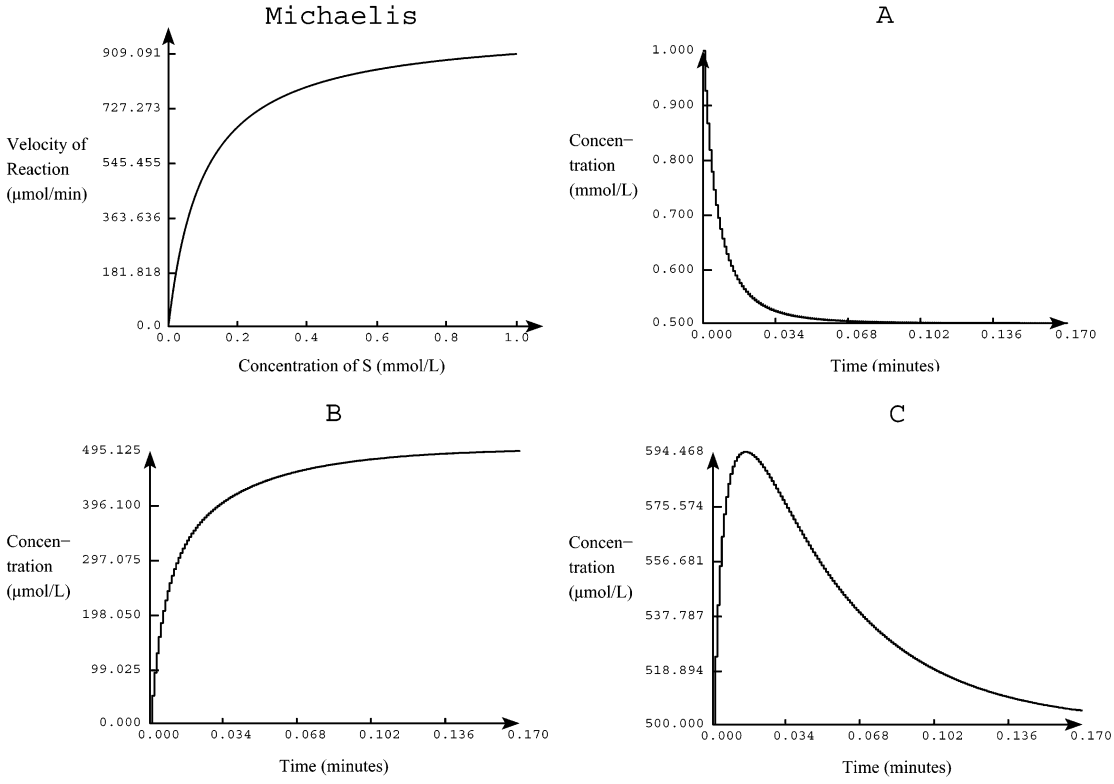

Figure 3. Dynamics of metabolites.

Table 2. Parameters of Enzymes Used in the Simulation

enzyme	catalytic constant	turnover number	equilibrium constant	concn (mM)
hexokinase (HK)	6450	12900	850.0	0.0001
phosphoglucose isomerase (PGI)	399000	798000	0.51	0.002
phosphofructokinase-1 (PFK)	7030	28100	310.0	0.001
aldolase (ALD)	667	2670	6.43e-005	0.002
triose-phosphate isomerase (TPI)	227000	454000	0.0472	0.002
glyceraldehyde-3-phosphate dehydrogenase (G3PDH)	5340	21400	0.0786	0.02
phosphoglycerate kinase (PGK)	102950	102950	2060.0	0.006
phosphoglycerate mutase (PGM)	52000	104000	0.169	0.004
enolase (ENO)	6000	48000	2.5	0.003
pyruvate kinase (PK)	2840	11400	363000.0	0.007

3. Glycolytic Pathway. A classical example of a metabolic pathway is the glycolytic pathway, which consists of the following 10 reactions (also shown in Figure 2):

1. ATP + G → ADP + G6P
2. G6P ↔ F6P
3. ATP + F6P → ADP + FBP
4. FBP ↔ DHAP + G3P
5. DHAP ↔ G3P
6. G3P + Pi + NAD ↔ BPG+ NADH
7. ADP + BPG ↔ ATP + 3PG
8. 3PG ↔ 2PG
9. 2PG ↔ PEP
10. ADP + PEP → ATP + Pyr

where DHAP, G3P, Pi, BPG, 3PG, 2PG, PEP, and Pyr are respectively dihydroxyacetone phosphate, D-glyceraldehyde 3-phosphate, orthophosphate, 1,3-bisphosphoglycerate phosphate, 3-phospho-D-glycerate, 2-phospho-D-glycerate, phosphoenolpyruvate, and pyruvate in brief. Among the 10 reactions, three are treated as irreversible (shown by one-direction arrows in Figure 2 and the equations above), and their enzymes are inhibited by ATP (R_3 and R_10) or G6P (R_1). The first reaction and the third reaction use ATP as a substrate, while the sixth reaction and the last reaction produce ATP. Eventually two ATP molecules will be produced from the breakdown of one molecule of D-glucose, and the overall pathway can be written as G + 2ADP + 2Pi + 2NAD → 2Pyr + 2ATP + 2NADH.

Assume that the volume of the compartment is 1×10^{-9} mL, the surface area of the compartment membrane is 6×10^{-6} cm^2, and the thickness of membrane is 6 nm. Assume also that the initial concentrations of D-glucose, ATP, ADP, NAD, NADH, and Pi are set according to the data from human red cell,[8] 0.0004 mM, 1.60 mM, 0.29 mM, 1 mM, 0.0023 mM, and 1 mM, respectively. Considering these metabolites may play roles in many other pathways, their concentrations except ATP's are kept constant during the course of the simulation. All other metabolites which can be produced in the pathway are initially empty (0 mM). The concentrations of enzymes are also from ref 8, while other parameters of corresponding enzymes are from EMP (http://emp.mcs.anl.gov) (Table 2). Michaelis constants between substrates/products and enzymes are also from EMP. Parameters of inhibitors are adopted arbitrarily but with reference to data from EMP (Table 3). The half-saturation substrate concentration of Sink_1 or Sink_2 is 0.0003 cm/min and 0.01 mM, and their permeability coefficients are respectively 0.000333 cm/min and 0.00333 cm/min.

Table 3. Inhibition Constants and Hill Constants of Inhibitors

	HK		PFK		PK	
inhibitor	K_I	H	K_I	H	K_I	H
G6P	0.02	2.0				
FBP			0.0014	2.0	0.01414	2.0
3PG			0.07071	2.0		
Pyr					0.0037	2.0
ATP	1.414	2.0	1.387	2.12	1.4142	2.0

Table 4. Concentrations at 5 min, 24 h, and the Expected Concentrations of Some Metabolites

metabolites	10 min	24 h	expected[a]
G6P	0.0485	0.0494	0.0486
F6P	0.0242	0.0247	0.0198
FBP	0.0189	0.019	0.0146
DHAP	0.158	0.159	0.16
G3P	0.0073	0.0073	0.00728
BPG	0.0002	0.0002	0.000243
3PG	0.0708	0.0712	0.0773
2PG	0.0113	0.0113	0.0113
PEP	0.0278	0.0279	0.0192
Pyr	0.06	0.0601	0.06

[a] The expected concentrations are from ref 8.

After running for a few minutes, the system reaches steady-state. The steady-state concentrations are listed in Table 4.

The steady-state concentrations are close to expected values. We also noted that the three reactions controlled by inhibitors are far from equilibrium, while all other reactions are very near to equilibrium. Considering many of these intermediate products also involved in many pathways in living red cells other than just the glycolytic pathway modelled in this simulation, the results are plausible.

The simulation indicates that the pathway is robust enough to resist considerable disturbance. If started with the expected concentrations, the intermediates will reach 90% of the steady-state level in 3 min. After steady state has been reached, it can be quickly recovered even if we change the concentration of an intermediate product to zero. The time needed to recover G6P, F6P, FBP, DHAP, G3P, BPG, 3PG, 2PG, or PEP from 0 to 90% of the steady level is correspondingly 2.5 min, 1.5 min, 0.5 min, 1.5 min, 1 s, 0.5 s, 3 min, 10 s, and 1 min, obviously related to their concentrations.

4. Uncatalyzed, Cyclic Pathways. While Michaelis–Menten kinetics has been implemented for enzymatic reactions—as embodied in the equations described above—

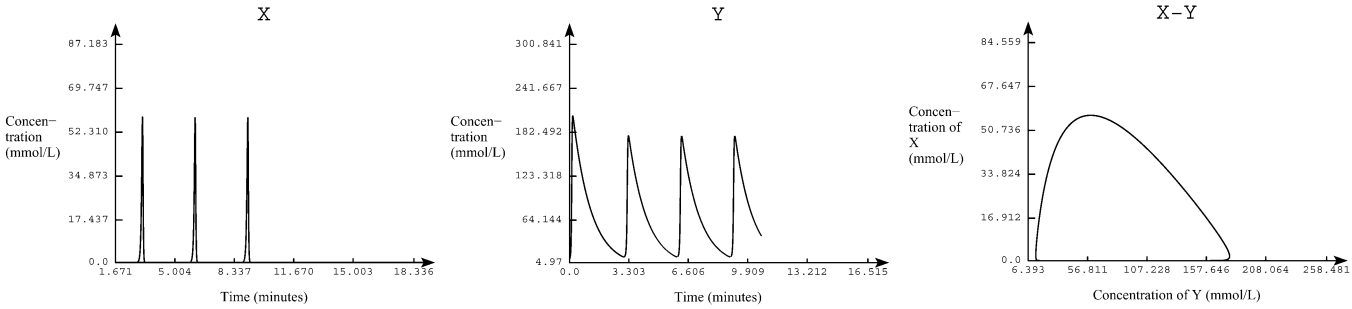

Figure 4. Lotka-Voltera system outputs.

DiMSim is able to model uncatalyzed reactions. It is also able to model pathways which give rise to cyclic flux graphs of the sort predicted by Lotka-Voltera[12] systems of reactions such as

$$X + A \rightarrow 2A$$

$$X + Y \rightarrow 2Y$$

$$Y \rightarrow B$$

Figure 4 shows three graphs resulting from one simulation of this system: [X] and [Y], each versus time, and [Y] versus [X]. After an initial, transient response the system settles into a cyclic behavior that persists over time; while the graphs show the state after 10.623 min of simulated time the heights of the peaks remain unaltered across much longer simulation runs, thus also demonstrating that there is no loss in accuracy which would otherwise cause the peaks to steadily decrease and with them the amplitude of the cycle.

DISCUSSION

Although the metabolic networks of DiMSim are similar to Petri nets, there are significant differences between them; the DiMSim networks are by now also very different from the graphs used in the data-flow computation model. For example, in DiMSim arcs are used to define relationships between objects, e.g. inhibitors/activators to enzymes, or from enzymes to reactions. They can connect metabolites (the equivalent of places in Petri nets) or be bidirectional when connecting metabolites with reversible reactions (the equivalent of transitions in Petri nets). Arcs are colored to distinguish different kinds of relationships. Since arcs represent relations, they no longer necessarily specify fixed inputs or outputs. Reddy et al.[17] uses self-loops to represent enzymes; DiMSim, on the other hand, denotes an enzyme by a blue arc pointing from the enzyme to a reaction. Petri nets only use two kinds of nodes (places and transitions) and one kind of directed edges (arcs); by contrast, DiMSim offers these and other objects, such as Compartment and Subsystem, to model metabolic pathways.

Using an ODE-based system to simulate metabolic pathways requires an often complex set of equations, which means a number of parameters have to be determined experimentally or set arbitrarily. More importantly, these parameters may have little biological significance so biological insight will not be able guide parameter choice. This problem is exacerbated for complex networks of pathways. Furthermore, the parameters need to be recalculated if there are any significant changes in the environment. By contrast, DiMSim uses a per-reaction rate equation which is based on a few basic constants in common use by biologists, e.g. equilibrium constant, Michaelis constant, and turnover number, which makes it understandable to users and predictable when the constants are adjusted. The outputs of the system are concentrations of metabolites of interest, generally viewed as a chart.

Another benefit is scalability, i.e., the computational cost of increasing the size of the networks being modeled grows proportionally because each added node (reaction) only impacts a limited number of other nodes and hence the number of reactions that must be checked when a reaction completes. Furthermore, because reactions operate in a manner similar to primitive numerical integrators we believe that the DiMSim model is also more numerically stable and able to tolerate errors in the input data than ODE-based systems.

One other thing is worth noting. DiMSim makes almost no assumptions about the kinetics of the system being modeled. As noted above, while Michaelis−Menten kinetics has been implemented for enzymatic reactions, Lotka-Voltera systems of uncatalyzed reactions can also be modeled. What makes this possible is the fact that the only assumptions built into DiMSim are that each reaction has a fixed stoichiometry (fan-in and fan-out, in graph-theoretical terms) and that each reaction takes a specified, finite amount of time.

DiMSim promises to be particularly effecting in helping users understand (and though that intervene) in complex biological systems involving multiple interacting pathways. Once comprehensive models for pathways of interest have been built, simulation experiments can be carried out which may not even be possible in the physical system. For example, if multiple reactions produce a metabolite into a single pool, with other reactions consuming that metabolite, it is well nigh impossible to determine the relative impact of each reaction by examining the flux of the metabolite in the pool. In DiMSim, this can be done quite easily. Similarly, to model how metabolites on different pathways react to a particular condition, multiple charts can be set up to track each metabolite versus time, or pairwise, each against a common-denominator metabolite, as in the Lotka-Voltera example above.

Another situation is where one or more intermediate reactions are not known. In this case one possibility would be to create a notional reaction in that place and then tune the parameters via comparisons with experimental data as they become available. (The subsystem facility may be useful here. In the larger pathway the will appears as a single reaction, but internally it can grow from a single notional reaction to a subpathway as more data is available and the model is refined.) Extending this methodology, where a

pathway is suspected via data from the literature (or sources such as EcoCyc/MetaCyc or Kegg), or via mRNA expression-array data, a framework-pathway can be created and then refined as kinetic data becomes available, most importantly turnover numbers and K_{eq}.

AVAILABILITY

DiMSim will soon be available under academic and nonacademic licenses. For further information, e-mail the corresponding author.

ACKNOWLEDGMENT

We would like to thank Dr. Andrew McDonald, Department of Biochemistry, Trinity College, Dublin, first for making us aware of the Lotka-Voltera reaction systems, and in particular the set of equations used in this study, and second for being the first external user of DiMSim, and in that role providing a great deal of extremely useful feedback. We would also like to acknowledge the generous support for the Research Fellowship and Research Associateship provided by Bristol-Myers Squibb.

REFERENCES AND NOTES

(1) Cornish-Bowden, A. *Fundamentals of Enzyme Kinetics*; Portland Press: 1995.
(2) Ehlde, M.; Zacchi, G. Mist: A user-friendly metabolic simulator. *Comput. Applications Biosci.* **1995**, *11*, 201−207.
(3) Ellis, R. J. Macromolecular crowding: Obvious but underappreciated. *Trends Biochem. Sci.* **2001**, *26*, 597−604.
(4) Erb, R. S.; Michaels, G. S. Sensitivity of Biological models to errors in parameter estimates. *Pacific Symposium on Biocomputing*; Hawaii, U.S.A., 1999; pp 53−64.
(5) Fell, D. *Understanding the Control of Metabolism*; Portland Press: 1997.
(6) Garrett, R. H.; Grisham, C. M. *Biochemistry*; Saunders College: 1999.
(7) Heidtke, K. R.; Schulze-Kremer, S. Biosim- a new qualitative simulation environment for molecular biology. *Sixth International Conference on Intelligent Systems for Molecular Biology (ISMB'98)*; Montreal, Canada, 1998; pp 85−94.
(8) Joshi, A.; Palsson, B. O. Metabolic dynamics in the human red cell. part iv-data prediction and some model computations. *J. Theor. Biol.* **1990**, *142*, 69−85.
(9) Kanehisa, M.; Goto, S. Kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.* **2000**, *28*, 27−30.
(10) Karp, P. D.; Riley, M.; Paley, S. M.; Pellegrini-Toole, A. *Nucleic Acids Res.* **2002**, *30*, 59−61.
(11) Karp, P. D.; Riley, M.; Saier, M.; Paulsen, I. T.; Collado-Vides, J.; Paley, S. M.; Pellegrini-Toole, A.; Bonavides, C.; Gama-Castro, S. The ecocyc database. *Nucleic Acids Res.* **2002**, *30*, 56−58.
(12) Lotka, A. J. Undamped oscillations derived from the law of mass action. *J. Am. Chem. Soc.* **1920**, *42*, 1595−1599.
(13) Mendes, P. Biochemistry with numbers: Simulation of biochemical pathways with gepasi3. *Trends Biochem. Sci.* **1997**, *22*, 361−363.
(14) Peterson, J. L. Petri nets. *Computing Surveys* **1977**, *9*, 223−252.
(15) Pfeiffer, T.; Sanchez-Valdenebro, I.; Nuno, J. C.; Montero, F.; Schuster, S. Metatool: for studying metabolic networks. *Bioinformatics* **1999**, *15*, 251−257.
(16) Raczynski, S. When system dynamics ode models fail. *Simulation* **1996**, *67*, 343−349.
(17) Reddy, V. N.; Mavrovouniotis, M. L.; Liebman, M. N. Petri net representations in metabolic pathways. *First International Conference on Intelligent Systems for Molecular Biology*, July 6−9, 1993, Bethesda, MD, pp 328−336.
(18) Sauro, H. M. Scamp: A general-purpose simulator and metabolic control analysis program. *Comput. Applications Biosci.* **1993**, *9*, 441−450.
(19) Selkov, E.; Galimova, M.; Goryanin, I.; Gretchkin, Y.; Ivanova, N.; Komarov, Y.; Maltsev, N.; Mikhailova, N.; Nenashev, V.; Overbeek, R.; Panyushkina, E.; Pronevitch, L.; Selkov, E., Jr. The metabolic pathway collection: an update. *Nucleic Acids Res.* **1997**, *25*, 37−38.
(20) Rivest, R. L.; Cormen, T. H.; Leiserson, C. E.; Stein, C. *Introduction to Algorithms (2e)*; MIT Press: 2001.
(21) Tomita, M.; Hashimoto, K.; Takahashi, K.; Shimizu, T. S.; Matsuzaki, Y.; Miyoshi, F.; Saito, K.; Tanida, S.; Yugi, K.; Venter, J. C.; Hutchison, C. A. E-cell: Software environment for whole-cell simulation. *Bioinformatics* **1999**, *15*, 72−84.
(22) Treleaven, P. C.; Brownbridge, D. R.; Hopkins, R. R. Data-driven and demand-driven computer architecture. *ACM Comput. Surveys* **1982**, *14*, 93−143.

CI025650W