

Quantitative, scalable discrete–event simulation of metabolic pathways

Peter A. Meric

Basser Department of Computer Science
University of Sydney
Sydney NSW 2006 Australia
pmeric@cs.usyd.edu.au

Michael J. Wise *

Centre for Communications Systems Research
10 Downing St
Cambridge CB2 3DS UK
M.Wise@ccsr.cam.ac.uk

Abstract

DMSS (Discrete Metabolic Simulation System) is a framework for modelling and simulating metabolic pathways. Quantitative simulation of metabolic pathways is achieved using discrete–event techniques. The approach differs from most quantitative simulators of metabolism which employ either time–differentiated functions or mathematical modelling techniques. Instead, models are constructed from biochemical data and biological knowledge, with accessibility and relevance to biologists serving as key features of the system.

Introduction

DMSS seeks to create a scalable, object–oriented, quantitative discrete–event simulation framework based on a model of metabolic pathways as a graph (or network) of reactions. The system will facilitate the creation and evaluation of models through the simulation of experiments based on these models.

Similar approaches have been used to model metabolic pathways, in particular the approach based on Petri nets (a type of graph). This method has been suggested for modelling of pathways and for use in discrete–event simulation systems (Hofestädt & Thelen 1998; Reddy, Mavrouniotis, & Liebman 1993; Regan, Bogle, & Dunnill 1993).

Other approaches to modelling and simulating metabolic pathways include differential equation and Metabolic Control Theory systems, for example (Mendes 1997; Ehlde & Zacchi 1995; Sauro 1993). In addition, there have been a number of attempts to use knowledge–based and reasoning systems, for both qualitative and quantitative simulation (Heidtke & Schulze–Kremer 1998; Farza & Cheruy 1992)

DMSS is designed to be used by bench biologists, allowing complex biological models to be expressed in biological terms (for example, as metabolite concentrations or chemical reactions), rather than as sets of obscure

equation constants. That is, the biological relevance of the input model, simulation process and thus the results is imperative. The first step to achieving this goal is the ability to create biologically relevant models, hence a model must be defined in terms of biological constants. The system should not require this knowledge to be described in mathematical terms, particularly not with respect to time. In contrast, the constants used in differential–equation based mathematical models arise as a consequence of requiring the formulae to describe the reaction model for a range of reaction–related parameters. Additionally, the stability and sensitivity of the equations for any given parameters need to be determined and accounted for, as these are often artifacts in such equation–based simulation systems.

DMSS seeks to simulate non–trivial systems, an important issue for metabolic simulation systems—ones which are considered problematic for other techniques (Thomas & Fell 1993; Krylov 1998). The simulation of complex reaction systems can contribute to our understanding of biological systems and be an experimental platform for *in silico* biology.

Model building

Discrete–event simulation of a biochemical system facilitates a bottom–up approach to modelling the system; one of the major aims of DMSS is to be able to simulate a metabolic system from descriptions of fundamental entities and the interactions amongst these entities. These fundamental entities are determined by our knowledge of the biochemical system, generally at the level of the metabolites and enzymes. For example, a model may be defined in terms of enzymes and the reactions they catalyze, the rate of a reaction, and conditions under which a reaction proceeds.

Differential equations have been widely used in the modelling and simulation of metabolic pathways. These equations are almost always differential with respect to time. The applicability of differential equations to biology has been long questioned, particularly with respect to issues of continuity and cycles (Williams 1977). The principal issue is in the use of differential equations to model a system over time. In fact, the use of differential equations will be of use, for example, in specifying

*Dr Wise is Senior Research Fellow at Pembroke College and CCSR under a grant from Bristol-Myers Squibb.

the rate of a reaction with respect to the concentration of certain metabolites. The equations provide a value which is, essentially, instantaneous—it acts more or less as a lookup table. All references hereafter to differential equations are specifically referring to those which are differential to time.

DMSS does not employ kinetic parameters, stoichiometry matrices, flux coefficients or the like—such models have often served to simplify the models of reaction systems and have been used for simulation purposes. These models often need to employ further simplification when modelling a system, for example the omission of system elements which remain relatively unchanged during a simulation—relying on the pseudo-steady state assumption. In addition, the use of differential equations may not work in complex and nonlinear systems. There is the possibility that solutions might not exist for the system of differential equations. In addition, verification of the applicability of such a model to the system is rare, and especially difficult given the possible numbers of nonlinear interactions and feedback loops (Raczynski 1996).

The complexity of systems of differential equations is often such that analytical analysis is not possible. In these cases, numerical investigation is still possible. However, since analytical investigation is desired, a smaller system of differential equations describing the biological system is required (Krylov 1998).

In modelling the interactions of fundamental entities, aggregate behaviour of the system is not being specifically modelled (as is typically the case with simulations based on systems of differential equations). Therefore, if the simulation of such a system shows differing aggregate behaviour, then one might conclude that either the model is incorrect or one's understanding of the biochemical mechanisms is flawed. Attempts to directly model these emergent properties of a system are contrary to the nature of such behaviour—hence, reliance of a model on such behaviour is undesirable; a simulated experiment may produce correct results, however this cannot validate the correctness of a model on its own.

The approach of modelling biological systems from fundamental entities is advantageous for biologists, particularly due to its accessibility—all terminology and definitions used are basic biological and biochemical terms. An issue with modelling at this level is that the required knowledge or data may not be available; in such cases, it may be necessary to model interactions at a higher level of abstraction. For example, the rate of a reaction can be modelled explicitly in terms of concentrations of any competing metabolites. Alternately, this could be modelled using metabolite affinities to enzymes plus the concentration of each given metabolite (as a set of molecules) to the enzymes. In this way the interaction of competing metabolites is modelled explicitly, yet the effect of that competition on the rate of a reaction remains implicit, derived from the more basic interactions.

The sensitivity of the model to the parameters used for simulation is a major issue for systems based on differential equations. It is very difficult to determine the sensitivities in such systems, as examined in (Erb & Michaels 1999). Any sensitivities in the simulation model would ideally correspond to sensitivities present in the biological system in question.

The discrete-event approach is to incorporate any such sensitivities into the model. This is done as part of defining the interactions amongst basic entities. Any sensitivities present in the model are defined by the model, directly modelling the sensitivities of the biological system.

Computational aspects

The system has been developed entirely in Java. This language was chosen for its standard libraries (providing data structures, I/O and threads), portability and memory management. The use of a just-in-time (JIT) compiler¹ results in acceptable performance levels. Multiple threads allow various modules of the simulation system to run concurrently, potentially reducing the running time of a simulation.

Architecture

Qualitative simulators are indicative of trends. Although these are useful to biologists, the potential to derive quantitative results is increasing with advances in knowledge and available data.

Discrete event simulation

The simulation engine is a dynamic model (Law & Kelton 1991), in that it represents a system's state over time. Hence, a simulation run provides the state of the system not only at the simulation's finish time, but at any time between the start and finish times inclusive.

This simulator is entirely deterministic—there are no stochastic elements in the simulation engine. This means that multiple simulation runs of a particular model will produce identical results. In addition to an experiment's reproducibility, this determinism allows for the continuation (extension) of simulation runs.

Time

Time is expressed as an integer value and a time-unit. A time value of three milliseconds is stored with 3 as the integer value and `millisecond` as the time-unit.

The simulator does not use a predetermined unit of time—rather the time-unit chosen is determined according to the smallest timescale in the simulation.

Simulation time corresponds to real (clock) time, except that the interval between updates of the clock's value are not necessarily of equal value, and the updates may occur without respect (or proportion) to real time. In spite of the unequal update values, the simulation

¹This compiles the Java bytecode into native machine code on-the-fly, facilitating direct execution of native code in contrast with having to interpret the bytecode

clock is of fixed units. (Thus, an increase in the clock of 20 ms is exactly equal to four increments of 5 ms each.)

Metabolites are represented by tokens, each token being representative of n metabolites/molecules, where $n \geq 0$. For example, 100 molecules of ATP can be represented by a token (ATP, 100). Enzymes are represented in an analogous manner.

Reactions are represented as a set of metabolite and enzyme tokens—the quantities represented by the tokens are in the required ratios as determined by the reaction.

If a reaction occurs at time t , it will have a particular duration (either constant or calculated at the start of the reaction). The metabolites and enzymes involved in the reaction are set aside, marked as being unavailable. At the reaction end time, $t + \textit{duration}$, the enzymes are made available, the substrates are removed, and the reaction's products are added. The quantity of products is determined by the reaction. There may be a proportion of the substrates that do not react fully but instead dissociate from the enzyme, in which case the appropriate adjustment is made.

It can be seen that a set of reactions is represented very concisely. For example, a set of 100,000 hexokinase reactions is represented as a single entity, a homogeneous set of reactions with a particular start and end time. The reaction will have a hexokinase token for 100,000 molecules, a glucose token for 100,000 molecules, and so forth.

DMSS employs a next-event time-advance mechanism, a widely used approach in discrete-event simulation (Law & Kelton 1991). This allows the simulator to skip over periods of inactivity, where there are no changes to the system's state; changes to the system's state occur at the start and end of reactions. As there are no changes to the system's state during the course of a reaction, this can be considered to be a period of inactivity. However, the inactivity of the simulation environment as a whole needs to be considered. If metabolites and enzymes are available for the purposes of reacting, then the system cannot be considered to be inactive. However, certain reactions may not be able to occur due to lack of a particular metabolite or due to inhibition of enzymes.

Event List

Discrete event simulations usually employ an event list to store all known future events (for example, the arrival of another customer in a bank queue simulation). However, the simulation of reaction systems does not require a future events list—future events are not queued, but will actually just occur given suitable conditions. Reactions do not occur at a predetermined time, but are instead triggered by substrate and enzyme availability.

On the other hand, similar to an event list, DMSS uses a reaction list to keep track of all reactions currently taking place. The reactions in this list are sorted

by increasing finishing time, allowing for fast retrieval of the finish time for the first-completing reaction.

Algorithm

The simulator performs a number of tasks on each iteration, where one iteration is required for a single timestep. The algorithm described below is the process undertaken by the simulator at each timestep.

1. complete any reactions finishing at the current time
 - substrates used in completing reactions are removed
 - products of the reactions are added, and made available
 - enzymes used in the reaction are made available to react again
 - changes to values are logged as required
2. assign available compounds to enzyme reactions
 - account for substrate competition
 - assign metabolites and enzymes to reactions and modify their status accordingly
3. calculate and return timestamp for next iteration
 - the simulation clock is advanced to the next possible event time. This is determined by the availability of metabolites and enzymes.
4. increment clock for next iteration
5. go to step 1 unless clock is equal to or greater than simulation end time

Logging Engine

The logging engine is the module responsible for the storing of simulation data. Data from a single simulation run is stored in a single file (the log file). This log file consists of the experiment details (for example, simulation start and end times), a description of the model, a list of log references, and finally the list of values to be logged. A log reference is a unique, system-assigned identifier which is assigned to every metabolite or enzyme for which simulation data is to be collected. These data collection points are known as watchpoints.

The data values in the log file are a series of log notices, each of which has a timestamp and a list of values. Each of these values is a tuple of (`Log Ref`, \mathcal{X}), where \mathcal{X} is a numeric value of the change in quantity of the compound referred to by the log reference, `Log Ref`, since its previously logged (or initial) value.

The activity of enzymes can also be logged, and is done on a reaction-by-reaction basis. However, an enzyme's watchpoint can also store only the overall activity of the enzyme. For any single-reaction enzymes, this issue is not applicable, but some enzymes are able to catalyze a number of reactions, examples being rubisco and glyceraldehyde 3-phosphate dehydrogenase. For enzymes catalysing multiple reactions, data for individual reactions can be used for comparative analysis of the enzymes' reactions.

Data model

Modelling metabolic pathways

Individual reactions constitute the basis of metabolic pathways. A pathway is formed merely by concatenating the reactions in the system, where the products of one reaction are used as substrates for another. However, it is unnecessary to define this pathway explicitly in DMSS.

Compounds, Reactions and Enzymes

A compound is a molecule without catalytic properties. Compounds are represented by the `Compound` class, which holds information such as the compound's name (or names), formula, and molecular weight. Some of this information may be omitted as it is not used by the simulator.

In a reaction, compounds are used to form the left- and right-hand sides of a reaction. The compounds are identified by name in the simulation model, which corresponds to a reference to `Compound` objects in the `Reaction` class. Reactions may have equivalent EC (enzyme commission) entries, however it is common for EC entries to refer to multiple reactions as they represent specific classes of enzyme reactions (IUBMB 1992).

An enzyme is a molecule with catalytic properties, and may catalyze one or more reactions. Each reaction for an enzyme has to be given a reaction duration, the time taken for a single reaction. This can be specified simply as a constant or as a rate equation. It is important to note that DMSS does not allow the rate equation to be specified as a differential over time, as explained earlier. If an equation is provided, then it will be used to calculate the duration of a reaction every time that reaction is to take place, rather than calculating the duration once. In this way, factors such as substrate and enzyme concentrations may be taken into account by the equation.

Enzymes in DMSS are regarded as compounds, and so the information associated with enzymes is the same as for compounds. In addition, an enzyme will have at least one EC number associated with it, denoting that enzyme's membership of the specified enzyme class. However, EC numbers cannot be used to identify a particular enzyme as the EC entries are a broader classification.

Experiments

An experiment is a defined procedure in a specified environment. DMSS allows for the creation of experiments, which then serve as the basis for simulation. The experiments define the compounds, enzymes and reactions and their quantities in the simulated system. The compounds and enzymes are listed, along with a specified quantity for each. Watchpoints, the compounds and enzymes for which simulation data are to be stored, are then defined. The watchpoints are merely a list of compounds and enzymes. Finally, details of the simulation itself are required—this may include the name, a text

Experiment	CPU time	Simulation duration
Glycolysis	39.80 s	27000 ms
TCA	6.41 s	2000 ms

Table 1: Simulation runtimes

description and the duration of the experiment in terms of simulation time.

Results

Sample models of the glycolytic and tricarboxylic acid pathways were created to demonstrate the function of the simulation system. One experiment using each model was selected for discussion. Quantitative results from the experiments are detailed. Given that the models are of isolated pathways, the results are only valid in this isolated context.

The experimental values chosen for the pathways are currently arbitrary and have no biological basis. The values were carefully chosen to emphasise features of the model in the experimental results.

Plots for the results are drawn using steps, which reflect the instantaneous changes in quantity of compounds in the simulation. This is due to the fact that the simulation engine only modifies quantities at the completion of a reaction.

The abbreviations for compounds and enzymes used are as per common usage, or Stryer's *Biochemistry* where differences arise (Stryer 1995).

Computational aspects

The simulations were run on a Pentium 233Mhz computer. These runtimes include loading into memory of small databases for compounds, reactions and enzymes. Both experiments employed millisecond timesteps—therefore a simulation duration of 1000 milliseconds is requires a maximum of 1000 iterations, however due to the algorithm as explained earlier, the actual number of iterations may be less than this.

Experiment one: Glycolysis

The catabolic glycolytic pathway is a *simple*, linear pathway consisting of a series of ten reactions. Each reaction is catalysed by a different enzyme. The reactions are listed in Table 2.

Analysis

The graphs in Figure 1 depict the concentration of a number of compounds and cofactors in the glycolytic pathway as effected by an injection of glucose plus varying concentrations of the other metabolites. The vertical axes of the graphs represent the quantity of the specified metabolite in terms of millimoles. The horizontal axes represent time in terms of milliseconds. The values used for these metabolite concentrations were based on (Joshi & Palsson 1990). Additionally, the enzyme concentrations and reaction rates were based on data

Reaction	Enzyme
glucose + ATP \rightarrow G6P + ADP + H ⁺	hexokinase
G6P \rightarrow F6P	phosphoglucosomerase
F6P + ATP \rightarrow F1,6BP + ADP + H ⁺	phosphofruktokinase-1
F1,6BP \rightarrow DHAP + GA3P	fructose bisphosphate aldolase I
DHAP \rightarrow GA3P	triose phosphate isomerase
GA3P + NAD ⁺ + P _i \rightarrow 1,3-BPG + NADH + H ⁺	glyceraldehyde 3-phosphate dehydrogenase
1,3-BPG + ADP \rightarrow 3PG + ATP	phosphoglycerate kinase
3PG \rightarrow 2PG	phosphoglycerate mutase
2PG \rightarrow PEP + H ₂ O	enolase
PEP + ADP + H ⁺ \rightarrow pyruvate + ATP	pyruvate kinase

Table 2: List of glycolytic reactions and corresponding enzymes

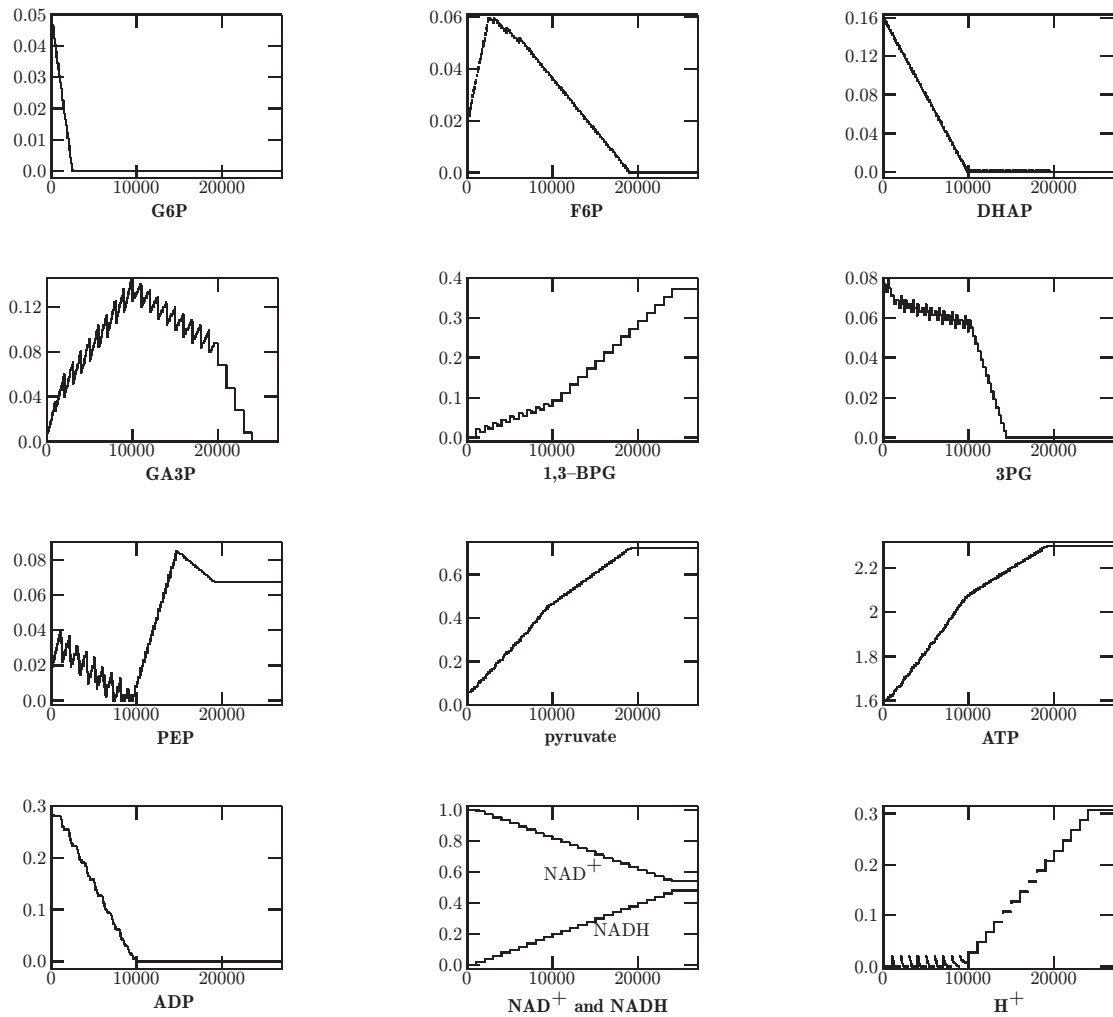


Figure 1: Concentration of glycolytic compounds

from (Beutler 1984). The experiment is based on data from the red blood cell, and demonstrates the use of biologically–reasonable inputs to produce biologically–acceptable results.

A feature of note in the graphs is the concentration of ADP, and the effect of its reduction to zero. The concentration of ADP is near zero from 10000 ms. However, the concentration of pyruvate is still climbing, which is notable as the pyruvate kinase reaction requires ADP. This is explained by the concentration of F6P, indicating that ADP is being produced by the phosphofructokinase reaction. The other ADP–producing reaction, of glucose to G6P, produces very little ADP due to the small amount of glucose available. The reason the ADP concentration graph remains near zero is that the ADP is being consumed as it is produced. When the concentration of F6P falls to zero, ADP is no longer produced and the availability of ADP falls to zero as soon as it is consumed by the pyruvate kinase reaction. The graphs for F6P and pyruvate show corresponding plateaus in concentration for this reason.

The fluctuations in the graphs of GA3P and PEP are caused by the differences in reaction rates and enzyme quantities of the reactions in which those metabolites appear. The sharp increase in PEP concentration at approximately 10000 ms is due to the lack of ADP. The effects of this are also evident in the graphs for 1,3–BPG and 3PG as well as for ATP and H^+ .

The graph of NAD^+ and NADH shows the proportional nature of their concentrations in glycolysis— NAD^+ is consumed at the same rate as NADH is produced.

Experiment two: Tricarboxylic acid cycle

This experiment was devised to demonstrate the ability of the simulation system to deal cycles in the pathways. The values used for this experiment were selected to effectively demonstrate the cyclic behaviour of the pathway, while falling within in a biologically meaningful range. All enzymes are in equal concentration, and have identical reaction times of 100 ms. The concentration of all metabolites in the cycle (as opposed to cofactors) is zero, apart from oxaloacetate, of which there is 1 millimole. The reactions are all substrate limited, there being an excess of enzyme for all reactions. Overall, the limiting factor in this experiment is the quantity of acetyl coenzyme A (acetylCoA), of which there is an adequate supply for three complete cycles. All cofactors are supplied in excess quantities, a necessity given the isolation of the pathway (the cofactors are normally produced in or transported to this compartment).

The eight reactions of the TCA cycle are listed in Table 3. The enzyme responsible for catalysing each reaction is shown next to the reaction.

Concentration of TCA compounds

Figure 2 depicts the concentrations of the compounds in the TCA cycle for the duration of the simulation. The peak and valley quantities are always 1.0 and 0.0 millimoles respectively. Arrows represent the reactions in the cycle, as listed in Table 3. Cofactors have been omitted from the diagram.

The axes on the graphs are labelled consistently—the horizontal axis represents time in terms of milliseconds, and the vertical axis represents the quantity in millimoles.

Analysis

The cycle begins with oxaloacetate, and its reaction with acetylCoA. The initial concentration of oxaloacetate is 1 mmol, which falls to zero at 100 ms. At this time, the concentration of citrate rises from zero to 1 mmol. This pattern then continues around the cycle. This is seen clearly by following the arrows. One cycle has been highlighted on the graphs for emphasis—a period of 800 ms, eight reactions of 100 ms each.

The graphs in Figure 3 show the concentrations of a number of the cofactors in the TCA cycle. CoA is produced by the citrate synthase reaction, consumed in the α –ketoglutarate dehydrogenase reaction, and then produced in the succinyl CoA synthetase reaction. Thus, for each cycle through the pathway, two molecules of CoA are produced for every one consumed (the additional CoA is provided by the acetylCoA, which is consumed by the TCA cycle).

The concentrations of both the GDP/GTP and NAD^+ /NADH pairs are complementary. One molecule of GDP is consumed for the production of a single GTP molecule, while three molecules of NAD^+ are consumed in the cycle for the production of an equivalent amount of NADH.

Conclusion

DMSS is a framework for the quantitative simulation of metabolic pathways. It allows biologists to create models of metabolic systems using only biochemical and biological data and knowledge of the system. The simulation of such a model is able to deliver biologically–significant quantitative results.

Models of metabolic pathways are not expressed with respect to time. Models based on time–differentiated functions may be expressed in other terms—the behaviour of a biological system is not explainable in terms of the passage of time. Cyclic behaviour, for example, could be expressed as a function over time—yet explanation and exploration of such behaviour is hindered by the time–related modelling method.

Reaction	Enzyme
$\text{oxaloacetate} + \text{acetylCoA} + \text{H}_2\text{O} \rightarrow \text{citrate} + \text{CoA} + \text{H}^+$	citrate synthase
$\text{citrate} \rightarrow \text{isocitrate}$	aconitase
$\text{isocitrate} + \text{NAD}^+ \rightarrow \alpha\text{-ketoglutarate} + \text{CO}_2 + \text{NADH} + \text{H}^+$	isocitrate dehydrogenase
$\alpha\text{-ketoglutarate} + \text{NAD}^+ + \text{CoA} \rightarrow \text{succinyl CoA} + \text{CO}_2 + \text{NADH} + \text{H}^+$	α -ketoglutarate dehydrogenase
$\text{succinyl CoA} + \text{P}_i + \text{GDP} \rightarrow \text{succinate} + \text{GTP} + \text{CoA}$	succinyl CoA synthetase
$\text{succinate} + \text{FAD} \rightarrow \text{fumarate} + \text{FADH}_2$	succinate dehydrogenase
$\text{fumarate} + \text{H}_2\text{O} \rightarrow \text{L-malate}$	fumarase
$\text{L-malate} + \text{NAD}^+ \rightarrow \text{oxaloacetate} + \text{NADH} + \text{H}^+$	malate dehydrogenase

Table 3: Reactions in the TCA cycle and their enzymes

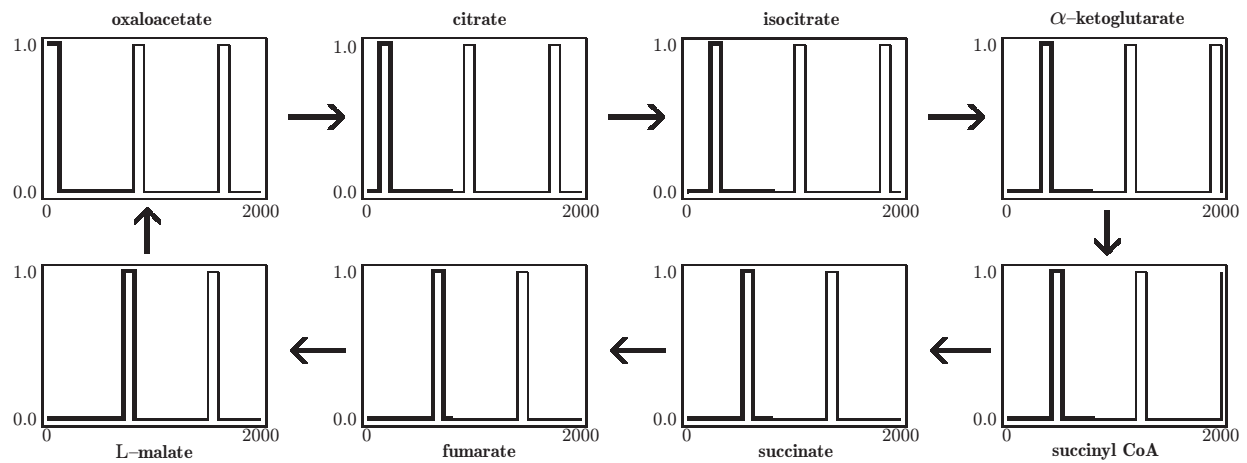


Figure 2: The TCA Cycle

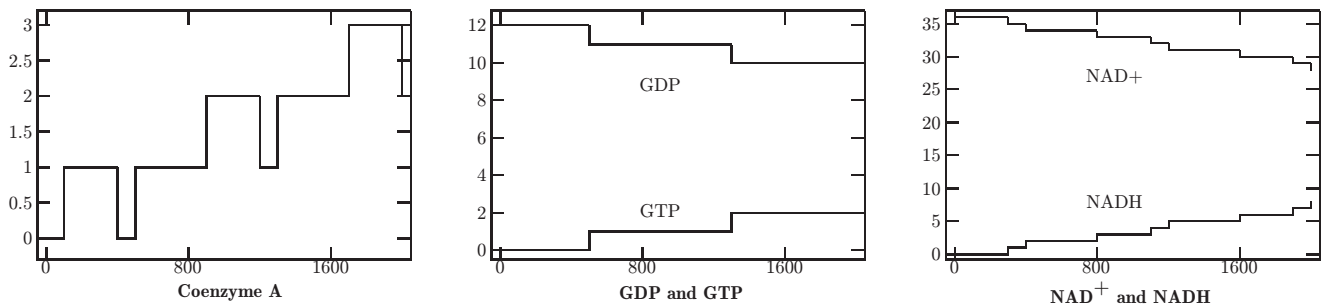


Figure 3: Concentration of selected cofactors

References

- Beutler, E. 1984. *Red Cell Metabolism: A Manual of Biochemical Methods*. Orlando: Grune and Stratton.
- Ehlde, M., and Zacchi, G. 1995. MIST: a user-friendly metabolic simulator. *Computer Applications in the Biosciences* 11(2):201–207.
- Erb, R. S., and Michaels, G. S. 1999. Sensitivity of biological models to errors in parameter estimates. In *Biocomputing '99: Proceedings of the Pacific Symposium*. World Scientific Press.
- Farza, M., and Cheruy, A. 1992. A typical bioprocess analysis through CAMBIO—a knowledge-based software for dynamical modelling and simulation of biochemical processes. In *European Symposium on Computer Aided Process Engineering—2*, S165–S170.
- Heidtke, K. R., and Schulze-Kremer, S. 1998. BioSim—A New Qualitative Simulation Environment for Molecular Biology. In *Proceedings of the Sixth International Conference on Intelligent Systems for Molecular Biology*, 85–94. AAAI Press.
- Hofestädt, R., and Thelen, S. 1998. Quantitative modeling of biochemical networks. *In Silico Biology* 1(0006). <http://www.bioinfo.de/isb/1998/01/0006/>.
- IUBMB. 1992. *Enzyme Nomenclature*. New York: Academic Press.
- Joshi, A., and Palsson, B. A. 1990. Metabolic Dynamics in the Human Red Cell. Part IV—Data Prediction and Some Model Computations. *J. theor. Biol.* 142:69–85.
- Krylov, S. N. 1998. Computer simulation of damped oscillations during peroxidase-catalyzed oxidation of indole-3-acetic acid. *Biophysical Chemistry* 72:285–295.
- Law, A. M., and Kelton, W. D. 1991. *Simulation Modeling and Analysis*. Singapore: McGraw-Hill.
- Mendes, P. 1997. Biochemistry by numbers: simulation of biochemical pathways with Gepasi 3. *Trends in Biochemical Sciences* 22:361–363.
- Raczynski, S. 1996. When System Dynamics ODE Models Fail. *Simulation* 65(5):343–349.
- Reddy, V. N.; Mavrovouniotis, M. L.; and Liebman, M. N. 1993. Petri net representations in metabolic pathways. In *Proceedings of the First International Conference on Intelligent Systems for Molecular Biology*, 328–336. AAAI Press.
- Regan, L.; Bogle, I. D. L.; and Dunnill, P. 1993. Simulation and optimization of metabolic pathways. *Computers chem. Engng* 17(5/6):627–637.
- Sauro, H. M. 1993. SCAMP: a general-purpose simulator and metabolic control analysis program. *Computer Applications in the Biosciences* 9(4):441–450.
- Stryer, L. 1995. *Biochemistry*. New York: W. H. Freeman, 4th edition.
- Thomas, S., and Fell, D. A. 1993. A computer program for the algebraic determination of control coefficients in metabolic control analysis. *Biochemical Journal* 292:351–360.
- Williams, M. B. 1977. *Needs for the Future: Radically Different Types of Mathematical Models*. Number 13 in *Lecture Notes in Biomathematics*. Berlin: Springer-Verlag. chapter 9, 225–240.