Math 4/5779: Mathematics Clinic
Pathway Inference: Computational Issues

Taught by Harvey J. Greenberg

Assisted by Raphael Bar-Or, Lance Lana, Cary Miller, Tod Morrison and Christiaan van Woudenberg

Sponsored by
DMI BioSciences, Genomica and Tech-X

Spring Semester 2002

Final Report Submitted May 29, 2002

Students	
Graduate	Undergraduate
Rico Argentati	Olasumbo Olufunke Adesola
Min Hong	Andrew Been
Lance Lana	Felemon Belay
Tod Morrison	Jennifer Dai
Adolfo Perez	Anissa Larson
Christiaan van Woudenberg	Xuan Le
Tessa Weinstein	Ben Perrone
	Jennifer Phillips
HSC Visitors:	Amy Rulo
George Acquaah-Mensah	Jon Stranske
Jens Eberlein	Kimberly Somers
Aaron Gabow	Xuan Tam
Cary Miller	Rob Wilburn
	UCD Visitor:
	Dung T. Nguyen

This is the main report, and there are two supplements:

Molecular Biology and Chemistry contains a general biology and chemistry background that is designed to provide both some details about the molecular biology and a big picture for insights into pathways.

Bibliography contains the complete bibliography, and many entries are annotated.

Contents

List of Tables

List of Figures

Abstract

This report is a first step toward understanding pathway inference from a computational view. We have some understanding of biochemical pathways and how they relate to biomedical research goals, but there has not been a unified description of how mathematical models and computer science methods can help scientists achieve their goal: *diagnose disease* and *discover therapeutics* to cure or control them. The objectives of this study are to understand the relevant science, survey current mathematical models, identify computational issues, and suggest a strategy for a follow-up study. This report contains varying degrees of completeness, with much left to be done. As more becomes known about biochemical pathways, on-line repositories of information about them will become increasingly useful in improving our understanding of disease states and our ability to discover novel therapeutics. It is in this context that a central focus of this study is on what is currently available and what needs to be done.

Acknowledgments

We thank our sponsors, DMI BioSciences, Genomica and Tech-X, for their support and encouragement. They were represented by Raphael Bar-Or, Rachel Green and Dimitre Dimitrov, respectively. Thanks to our team leaders: Raphael Bar-Or, Lance Lana, Cary Miller, and Christiaan van Woudenberg. Also, thanks to Tod Morrison for software installations and testing and his general leadership throughout the semester. Our guest speakers were particularly helpful:

Katheleen Gardiner, Eleanor Roosevelt Institute, presented *Pathway analysis for Down syndrome research*

Larry Hunter, Director of Center for Computational Pharmacology, School of Medicine, UCHSC, presented *On computational inference of biochemical pathways*

Cary Miller, Center for Computational Pharmacology, School of Medicine, UCHSC, presented *Basic elements of molecular biology*

Imran Shah, Department of Preventive Medicine & Biometrics, School of Medicine, UCHSC, presented *Pathway inference problems*

We especially thank Peter Karp for providing PathoLogic and its EcoCyc and MetaCyc databases, and for visiting us to discuss his software systems.

I personally thank all students. They did a great job, and I learned a lot!

Preface

This report represents the work of 13 undergraduates and seven graduate students at CU-Denver, Mathematics Department. We also had significant contributions from students not taking this for academic credit: Dung T. Nguen is an undergraduate student in Computer Science & Engineering, Raphael Bar-Or is a Ph.D. student in Mathematics, and Cary Miller is a Ph.D. student at the CU School of Medicine. Other students from HSC, listed on the cover of this report, sat in various sessions and contributed to discussions.

The basic objective, given at the beginning of the semester, was to fill in the outline (with possible changes) posted at http://www.cudenver.edu/~hgreenbe/courses/S02/objectives.html. (Browse the entire course site to see the *Conduct of Course* and other essential information.)

The first part of the semester was spent reading articles. I assembled most of the bibliography and selected the articles before the semester began. My selections were based on the following criteria:

- Relevance to pathway analysis
- Mathematics and computation contents
- Level of material
- Collective breadth of materials within the subject of pathway inference

The software that the students reviewed later in the semester was also selected at the outset, by the following criteria:

- Must be free, at least for academic use
- Relevance to pathway analysis
- Installs relatively easily

This report is the product of a lot of teamwork, which itself is an important pedagogical component of every Mathematics Clinic.

Executive Summary

A primary finding is that there does not appear to be much software currently available to support pathway inference, and those that do exist satisfy only a part of the needs scientists have. A concrete example of how pathway inference can help biomedical research was provided by Katheleen Gardiner: understanding the MAP Kinase pathway in the brain, and how controls might be applied to lessen the effects of Down syndrome.

We found that current mathematical models fall into three general categories:

1. Graphs and networks — captures structure
2. Ordinary differential equations — captures reaction kinetics
3. Optimization — captures ranges

These are not completely disjoint. For example, there are relations between boolean networks and ODE equilibria, and stoichiometric equations at equilibrium are constraints in a linear programming model for metabolic pathway analysis. In addition, each of these broad classes could have stochastic elements.

Non-mathematical models that are directly relevant to supporting pathway inference include *ontologies*. This is the foundation for Peter Karp's EcoCyc and MetaCyc databases, which we examined closely. Other models of importance include database schema, particularly extensible markup languages, like XML. One important example is the Systems Biology Markup Language (SBML).

We also found that there is currently no pathway inference support for statistical analysis. There is a particular need for making inferences with incomplete information about some pathway. This was briefly mentioned by Imran Shah, who is developing such a system using machine learning techniques.

The computational issues pertain to the following components:

Representation and overall framework. There are frameworks ranging from ontological to mathematical, and no one framework is dominantly best. This raises issues about developing software for different data schema. The leading ontological system is Peter Karp's PathoLogic, manifest in EcoCyc and MetaCyc databases. The other systems use a simple model specification language in plain text.

Visualization and alternative displays. This is fairly advanced, using graph layout algorithms that were developed in the 1970s as a starting point. Extensions include how to represent secondary reactions and including more zooming and hyperlink capabilities. A strength

of Karp's system is its ability to have the computer draw pathways, rather than enter them manually, as in other pathway databases, like the Kyoto Encyclopedia of Genes and Genomes (KEGG). The reason the former is better is that we expect to build the pathway database rapidly, and manual drawings take a significant amount of time. The other software systems were designed to present kinetic information, and they provide both graphical and tabular outputs for the computed trajectories. Some simple modifications can allow those results to be exported into more sophisticated visualization tools.

Algorithm design and analysis. The relevant algorithms have been taken from other applications and adapted to special issues that arise in pathway analysis, but new analysis is needed to consider the ways in which pathway inference will be conducted. Some sensitivity analysis is available, following standard techniques used by ODE solvers.

There are two more areas that appeared in the original outline: *Knowledge discovery methods and concepts* and *Inference tools, including open architecture for additions*. We found no articles or software about these; they appear to be virtually non-existent for pathway inference, but some results could be contained in articles that we did not read. (A Supplement accompanies this report that contains more than 100 references of interest; only a small portion of them were reviewed during this study.)

In summary, this study reveals opportunities to develop a new generation of information technology to support inferences based on interactions of genes or proteins. While commonly called "pathway inference," it is more accurate to say "network inference" because the interactions might be more complex than a linear sequence of biochemical reactions. Network inference is a part of the more general field of *systems biology*, which is more of an approach to this science than any one problem.

1 Introduction

This reports the results of the Mathematics Clinic: *Pathway Inference: Computational Issues*. Much of the course conduct and materials were posted on the web at http://www.cudenver.edu/~hgreenbe/courses/S02/5779_F02.html. (Another url will be posted when this is archived, and one can visit the Clinic site at http://www-math.cudenver.edu/clinic/.)

This report is organized as follows. The next section provides a quick introduction to pathways and addresses the question, "What is pathway inference?" A broader background in the biology and chemistry is given in a *Supplement*.

Section 3 summarizes mathematical modeling approaches that arise in pathway inference. These methods underlie some of the computational issues, which are addressed in §4. Section 5 gives an evaluation of some of the software we found on the internet. The only system with some proprietary restrictions is Peter Karp's PathoLogic and its EcoCyc and MetaCyc databases. We were licensed to use it free of charge because we are a nonprofit institution. The other systems are in the public domain, and we did not evaluate software that we could not obtain free of cost. The main discussion of the follow-up to these results is the subject of §6.

The final portion of this report consists of three glossaries, followed by an annotated bibliography. A supplement, published separately, consists of the entire bibliography, most of which were compiled before the beginning of the semester.

2 Pathway Inference Problem

A biochemical *pathway* is a sequence of chemical reactions that takes place in living organisms. It could be a simple sequence, like $A \longrightarrow B \longrightarrow C$, or it could have *branches*:

$$\begin{array}{ccccc} A & \longrightarrow & B & \longrightarrow & C \\ \downarrow & & \downarrow & & \\ D & \longrightarrow & E & & \end{array}$$

Here are some examples of questions:

- How fast does E build up?
- Is E mostly due to the path via B or the path via D?
- Which path is more important?

- How do medicines affect different paths?

Mathematically, a pathway is a network, where nodes are reactants and arcs represent some process by which one reactant is made from another. In most of the literature considered in this study, pathways are networks, or subnetworks, but the reality is that there could be many inputs and/or outputs, so a more accurate mathematical representation would be a *hypernetwork* (See our *Mathematics Glossary*, p. 40, for definitions of terms.)

It is also typical for there to be cycles, particularly feedback inhibition loops to have stability. Another question, therefore, might be to ask for the cycles in order to understand how the network regulates itself.

There are different kinds of pathways, and our study considered two: *metabolic* and *signaling*. Figure 1 shows one example of a metabolic pathway, taken from the Kyoto Encyclopedia of Genes and Genomes (KEGG). The numbers inside the nodes are the Enzyme Commission classification of proteins. In a metabolic pathway, the nodes are proteins. The tail of an arc is the protein that is input to the reaction, and it is called a *substrate*; the head of an arc is the protein that is an output of the reaction, and it is called a *product*, or *metabolite*. (See our *Biology Glossary*, p. 30, for definitions of terms.)

A pathway inference question might be, "Where is a good target to reduce the production of lysine?" A related question pertains to side-effects, where we want to know what other pathways contain the regulated protein.

Very little is known about pathways, so a figure like the Lysine Biosynthesis is atypical. Then, the inference question is more challenging, and might require some *artificial intelligence* to provide useful results. (See our *Computer Science Glossary*, p. 37, for definitions of terms.)

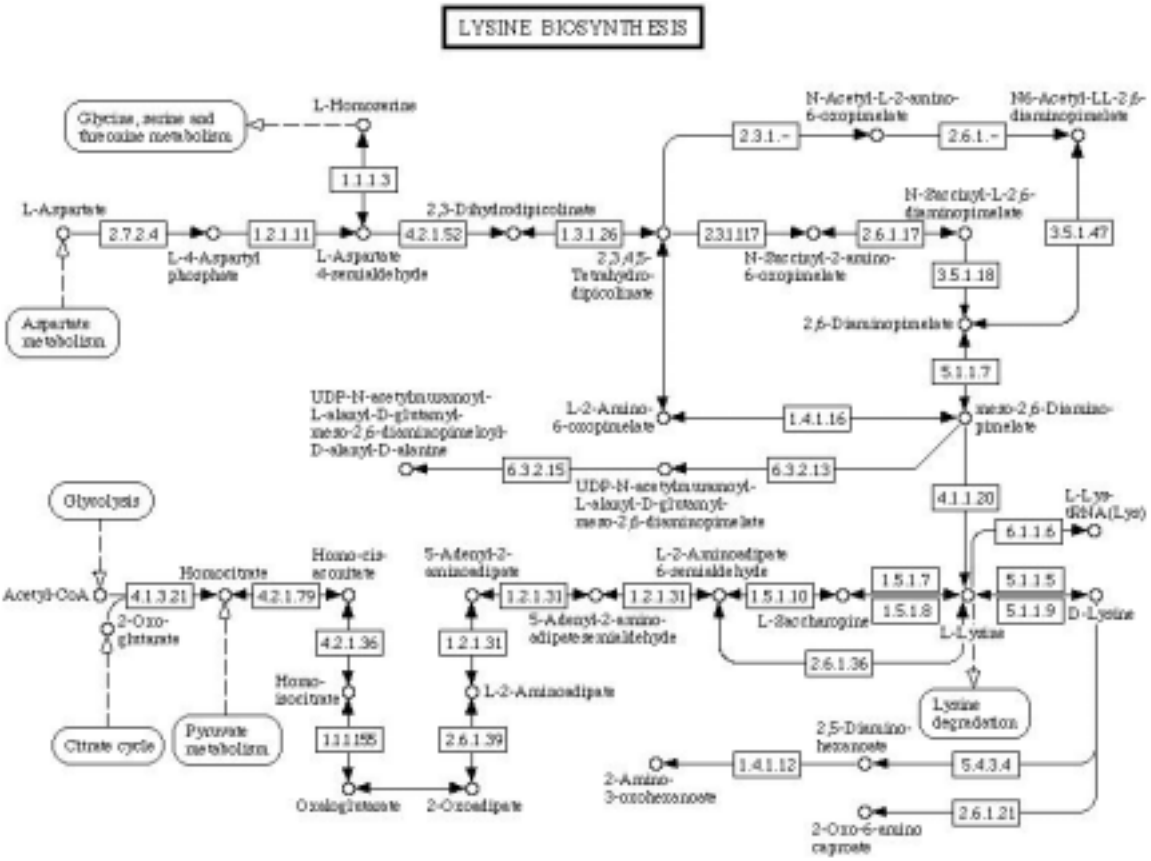


Figure 1: Lysine Biosynthesis Metabolic Pathway (from KEGG [20])

Figure 2 shows a copy of the MAPK pathway presented to us by Katheleen Gardiner. Her research is seeking to reduce learning deficits and neurodegeneration in Down syndrome patients. She is working on finding targets in the MAPK pathway to modulate the expression of chromosome 21 genes. (Down syndrome is caused by having an extra copy of chromosome 21, thus expressing those ∼250 genes about 50% more than normal.) Question marks appear where we do not know what the node represents, and the thing to note is that there are lots of question marks. (There are two colors, both shown as grey in this copy; yellow identifies genes on chromosome 21, and blue identifies genes related to the nerve growth factor. For a color view, see Dr. Gardiner's original presentation at http://www.cudenver.edu/~hgreenbe/courses/S02/notes/katheleen.ppt.)

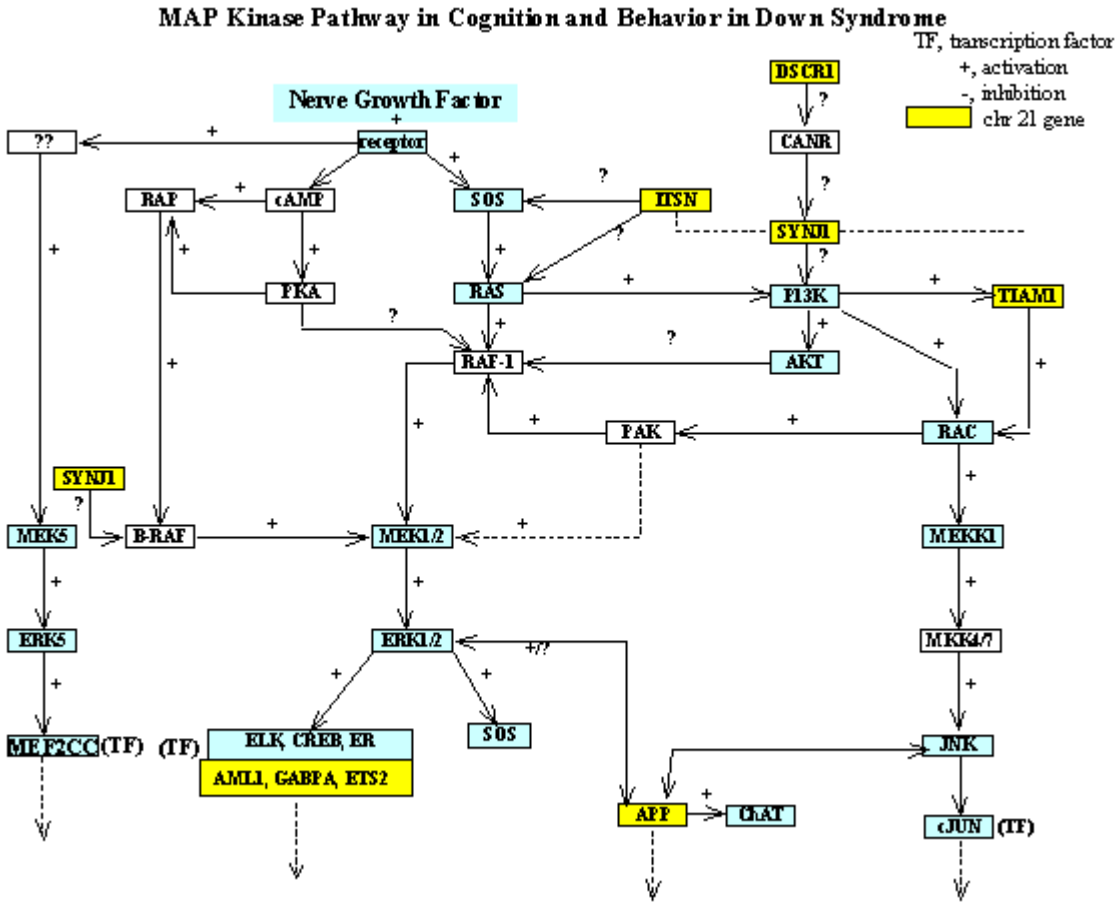


Figure 2: MAP Kinase Pathway in Brain (from K. Gardiner)

A Supplement is included to provide additional background into the biology and chemistry needed to understand pathway analysis beyond this quick introduction. It is not a complete description of the necessary science, but it serves our immediate purposes in making this report reasonably self-contained. Also, consult our *Biology Glossary* (p. 30). Additional background readings for the non-biologist are [4, 10, 12, 23, 28].

We now have a sense of what pathways are and how knowledge about them can be used to diagnose disease and design therapeutics. The *pathway inference* problem is to infer knowledge about pathways, typically with incomplete information about their parts and interactions. This is a very complex subject, and requires many disciplines to produce viable results. Besides the bio-sciences, we require knowledge of computer science and mathematics. For self-containment of this report, we include a *Computer Science Glossary*

(p. 37) and a *Mathematics Glossary* (p. 37).

3 Mathematical Models

With the advent of high throughput methods like microarrays, biology has migrated from a descriptive science to a predictive one; biology researchers finally have data to analyze. This data explosion underscores the need for accurate mathematical models.

A good mathematical model encompasses the following:

- Valid — captures the characteristic behavior of the system, as defined by its intended use
- Predictive — gives information about future events or missing information
- Parsimonious — no more complex than necessary
- Verifiable — relates the values produced by the numerical model to observable and measurable quantities
- Computable — can be solved by new or existing software in "reasonable" time

By necessity, mathematical models are simplifications of the observable world. When modeling a biochemical pathway, we omit the details of how the chemical reactions in that pathway occur and develop the model based on reaction rates and concentrations. We leave out superfluous information that would unnecessarily complicate the model and combine like reactions. In short, the level of detail with which we model a system needs to reflect the kind of behavior we want to capture with a model.

When modeling biological pathways to support pathway inference, we must deal with the fact that it is inherently a multi-level problem. For instance, we may want to model the actual production of a certain protein at the DNA to RNA level — that is, we want to model transcription to elucidate our understanding of a given *cis*-regulatory network. Suppose we know that the protein in our previous model is a substrate in a certain metabolic pathway, and that it is converted into another metabolite. We know that the two metabolites are connected, but we may not know to what extent. Alternatively, we may suspect that we have complete information about the connectivity of a pathway, but not know the reaction rates and fluxes associated with each reaction. There is no one approach sufficient to solve all of these problems. A comprehensive pathway inference support software will

contain a collection of mathematical models, each capturing some characteristic not captured by the others.

The remainder of this section describes some models we found in the literature:

- Boolean Networks — captures qualitative relations
- Differential Equations — captures kinetics of reactions
- Optimization — captures phenotype ranges

3.1 Boolean Networks and Finite Automata

A *boolean network* is a network whose nodes have associated binary-valued state variables. Boolean networks are used to model systems that can be simplified to a collection of on/off objects and relationships between objects that can be expressed as boolean functions. One example is gene regulation, where the expression of one gene can activate or inhibit the expression of another gene. An example of a cell signaling boolean network is shown in Figure 3.

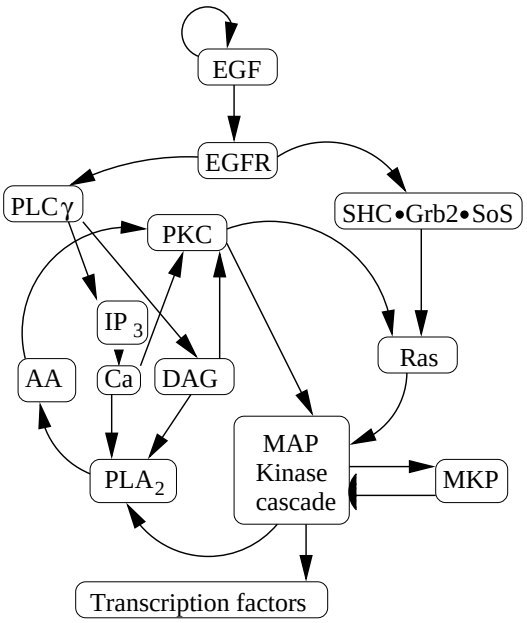


Figure 3: Ras-MAPK Cascade (F. Schacherer [34])

Given an *initial state* (which genes are expressed), denoted s^0, the dynamics

are governed by the following rule:

$$s_i^{t+1} = \begin{cases} 1 & \text{if } \sum_k \delta_{ki} s_k^t \geq 1 \\ 0 & \text{otherwise,} \end{cases}$$

where

$$\delta_{ki} = \begin{cases} 1 & \text{if the network contains activation arc } (k,i) \\ -1 & \text{if the network contains inhibition arc } (k,i) \\ 0 & \text{otherwise.} \end{cases}$$

Figure 4 shows an equilibrium cycle using the above rule and starting in the state shown in the Northwest corner: Ras, MAPK cascade, MKP, PLA_2 and the Transcription factors begin in the on state ($s_i^0 = 1$), indicated by those nodes with black background and white font. The next state is determined by applying the rule to each node, as shown in Table 1: four nodes are turned on (AA, which is an amino acid, was off at $t = 0$), and the MAPK cascade is turned off due to the inhibition from MKP (map kinase phosphate).

s^1_{Ras}	=	0	because no activation arc into Ras was on
$s^1_{\text{MAPK c.}}$	=	0	because the inhibition from MKP cancelled the activation from Ras: $s^0_{\text{Ras}} - s^0_{\text{MKP}} = 0$
s^1_{MKP}	=	1	activated by the MAPK cascade: $s^0_{\text{MAPKc.}} = 1$
s^1_{TF}	=	1	activated by the MAPK cascade: $s^0_{\text{MAPKc.}} = 1$
$s^1_{\text{PLA}_2}$	=	1	activated by the MAPK cascade: $s^0_{\text{MAPKc.}} = 1$
s^1_{AA}	=	1	activated by PLA_2: $s^0_{\text{PLA}_2} = 1$
s^1_{Ras}	=	0	because no activation arc into Ras was on

Table 1: State s^1, Shown in the Northeast Corner of Figure 4

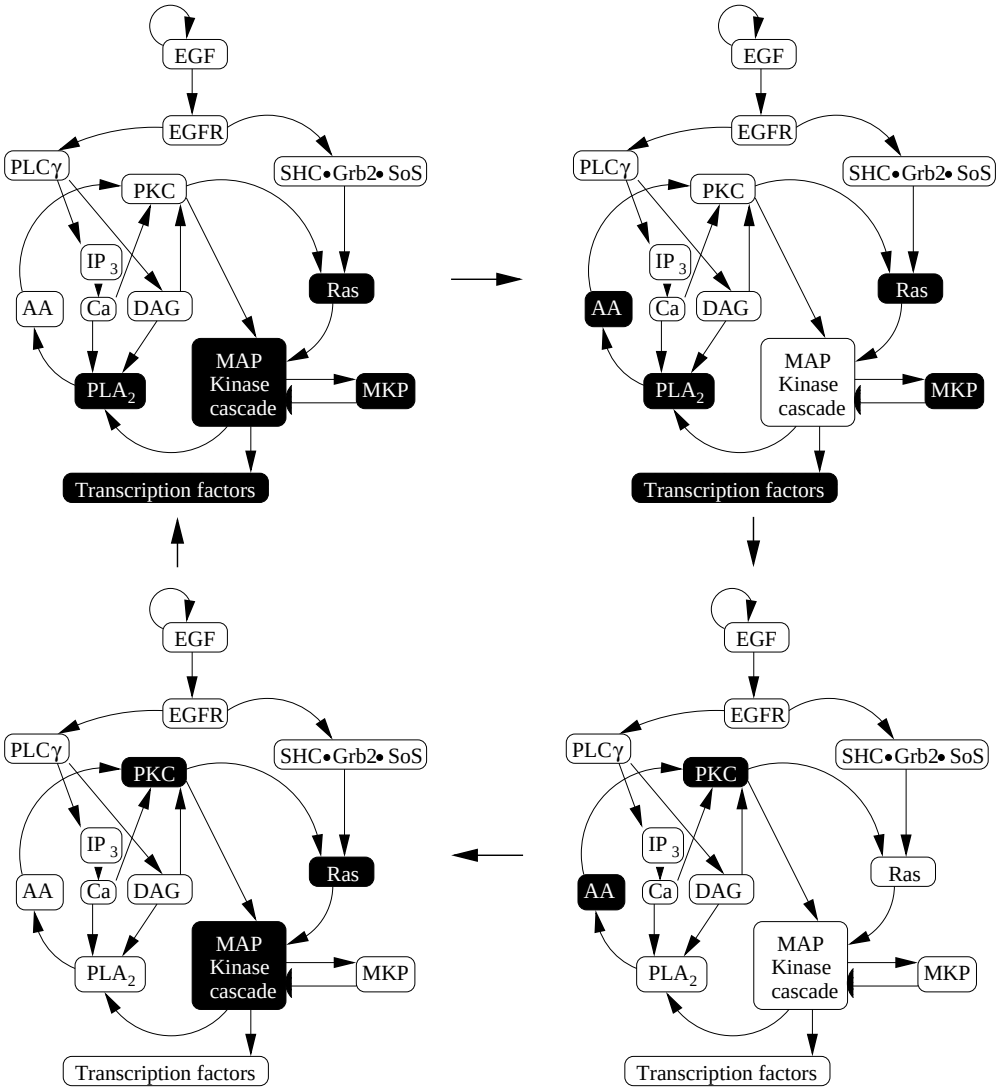


Figure 4: Equilibrium Cycle for the Ras-MAPK in Figure 3

There have been studies about using gene expression data to build boolean network models [1, 5, 22, 36]. We recommend studying the rules for the state transitions, particularly replacing the above simplistic rule with a more complex set of conditions to activate or inhibit gene expression.

The boolean network is a special case of a *finite state machine* (FSM), for which there is a substantial literature in logic and circuit design. An FSM is defined by six objects:

S = finite set of states; s^0 = initial state in S

I = finite input alphabet; O = finite output alphabet

f = transition function from $S \times I$ to S

g = output function from $S \times I$ to O

Then, the dynamics are as follows:

$$\begin{array}{ccccccccccc} i^0 & & i^1 & & i^2 & & & & i^t & & \\ \downarrow & & \downarrow & & \downarrow & & & & \downarrow & & \\ s^0 & \xrightarrow{f} & s^1 & \xrightarrow{f} & s^2 & \xrightarrow{f} & \cdots & \xrightarrow{f} & s^t & \xrightarrow{f} & \cdots \\ \downarrow & & \downarrow & & \downarrow & & & & \downarrow & & \\ o^0 & & o^1 & & o^2 & & & & o^t & & \end{array}$$

We start with state s^0 and we are given the initial input, i^0. Upon applying the state transition function, we obtain $s^1 = f(s^0, i^0)$. The output function produces $o^1 = g(s^0, i^0)$. Continuing in this fashion, we map the stream of inputs and initial state into a stream of states and outputs. Since every set is finite, we must eventually revisit a state. The input stream, however, could make the next state transition different. In the absence of an input stream, as in our example boolean network, once we revisit a state, we have an *equilibrium cycle*:

$$s^t \longrightarrow s^{t+1} \longrightarrow \cdots \longrightarrow s^{t+h} = s^t.$$

The states will repeat the same sequence. If there are no outputs, as in the boolean network, we stop, having detected one equilibrium cycle of states, as in Figure 4.

The boolean property of having only two state values, $S = \{0, 1\}$, is not really limiting. If we have any finite set of values, we can transform into a binary state space. For example, suppose one state variable can assume the values, $\{0, 1, 2, 3, \ldots, 2^k\}$. Then, replace this with new state variables, $r_1, \ldots, r_{k-1}$, where each r_i is in {0,1}, and

$$s = r_1 + 2r_2 + 4r_3 + \cdots + 2^p r_p + \cdots + 2^{k-1} r_{k-1}.$$

(It is easy to allow any finite number of values; it does not have to be a power of 2.)

Thus, finiteness is mathematically equivalent to being binary-valued. In that sense the boolean network is as general as a finite state machine. However, the I/O stream can be an important extension to consider. There is also a lot known about FSMs, so this presents an opportunity for further study.

3.2 Differential Equations and Dynamical Systems

An ordinary differential equation (ODE) describes the rate of change in some variable as a function of all variables and parameters of interest. The general form is:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(x, \theta, t).$$

For example, suppose we represent how the percentage of fibroblast cells in our skin changes over time with the following rate relation:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = x(1 - x),$$

where x is the fraction of fibroblast cells, and $1 - x$ is the fraction of healthy cells, in some portion of skin. When x is 0 or 1, there is no change: $\frac{\mathrm{d}x}{\mathrm{d}t} = 0$. However, if $x(0) > 0$, no matter how small, the cells eventually become fibroblast. This is expressed as the *limit*:

$$\lim_{t \to \infty} x(t) = 1.$$

If we wait long enough, 90% of the cells will be fibroblast; if we wait longer, 99% will be fibroblast. Pick any portion, say p%, and there is a time, T, when p% of the cells will be fibroblast, and that percentage will continue to increase for $t > T$. That is the meaning of this limit.

The system approaches an *equilibrium*, or *steady state*, if

$$\lim_{t \to \infty} \frac{\mathrm{d}x}{\mathrm{d}t} = 0.$$

A typical question is, "Does the system reach an equilibrium?"

This modeling approach can address questions of rates while the system is changing and can provide visualizations of how variables change over time. More complex dynamical systems modeling considers non-smooth kinetic functions, such as frequent jumps in state values.

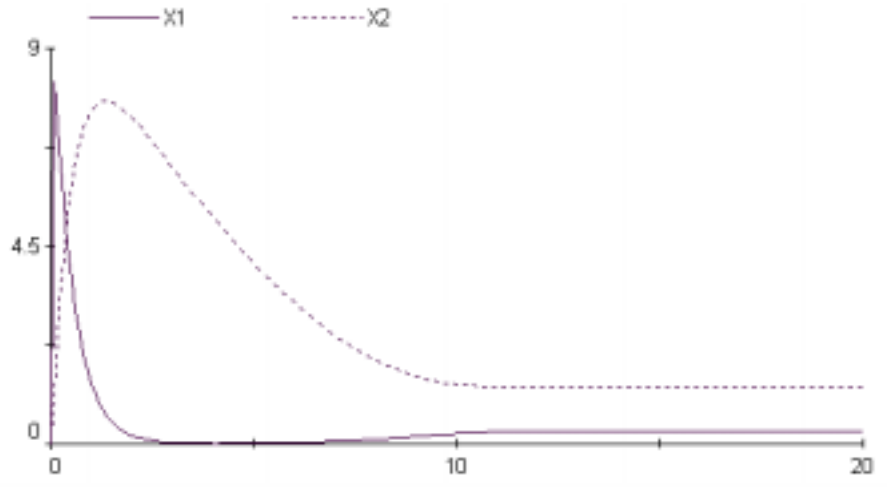
Whereas the ODE model is designed for continuous time and smooth kinetic functions, we can numerically simulate the more complex systems. In fact, there is a huge literature about this, accumulated over many decades, and one avenue for further study is to explore their application to complex biochemical pathways.

The ODE model can also address what types of limiting properties the variables have and how sensitive that behavior is to parameters. Consider the

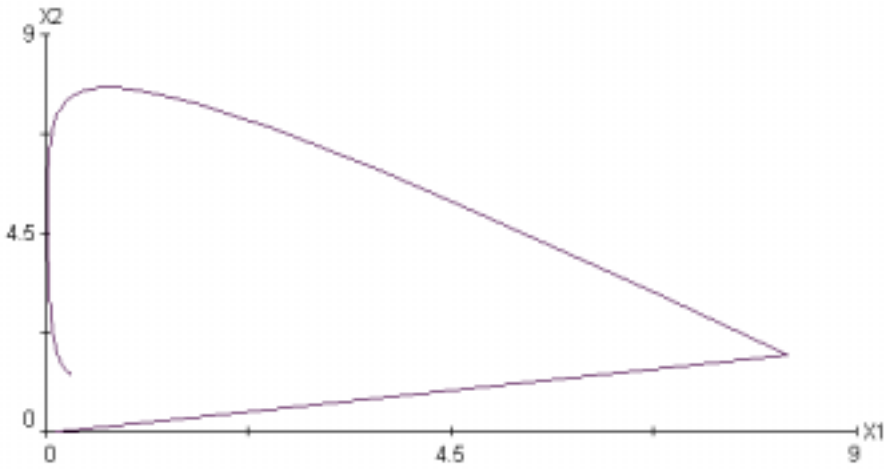
following example:

$$\begin{aligned} \frac{\mathrm{d}x_1}{\mathrm{d}t} &= -2x_1 + x_2^{-2} \\ \frac{\mathrm{d}x_2}{\mathrm{d}t} &= 2x_1 - \tfrac{1}{2}\sqrt{x_2} \end{aligned}$$

Given the initial values, $x(0) = (.1, .01)$, Figure 5 shows a *time plot* (each variable versus time) and a *phase plot* ($x_1(t) \times x_2(t)$).



(a) Time Plot



(b) Phase Plot

Figure 5: Plots of the ODE System (from PLAS [7])

Presently, these plots are the only form of visualization for viewing pathway dynamics. It is also not clear how to describe what is needed for visualizing

ODE model results. Interaction effects, for example, are not easy to visualize, though the concepts described by Uetz et al. [37] might apply here as well.

There are three types of ODE models of biochemical pathways:

1. S-system, based on the Michaelis-Menten rate law
2. Generalized Mass Action, based on a power law
3. Flux-Balance, based on conservation of mass

The Michaelis-Menten rate law is based on the idea that during an enzyme-catalyzed reaction, the substrate S binds to the enzyme E to form the enzyme-substrate complex ES. It is converted to an enzyme-product complex EP while it is part of this complex. The complex then dissociates to reconstitute enzyme E and free product P. This process is summarized in the following reaction [10]:

$$E + S \underset{k_{-1}}{\overset{k_{+1}}{\rightleftharpoons}} ES \underset{k_{-2}}{\overset{k_{+2}}{\rightleftharpoons}} E + P$$

The rate limiting step in this reaction is the dissolution of the complex ES into $E + P$, but cannot be easily measured in experiments. Instead, we approximate the product reaction rate [38] as follows:

$$\frac{\mathrm{d}P}{\mathrm{d}t} = \frac{v[S]}{K_m + [S]} E_0 = -\frac{\mathrm{d}[S]}{\mathrm{d}t},$$

where $[S]$ is the concentration of substrate S, v is the flux, and K_m is the *Michaelis-Menten constant*. This describes a rectangular hyperbola, where K_m is a measure of *substrate affinity*:

$$K_m = \frac{k_{-1}}{k_1} + \frac{k_2}{k_1} = K_S + \frac{k_2}{k_1}.$$

The flux of the reaction, v, is the rate of appearance of P. This rate is known to depend upon the concentration of S in the reaction. The Michaelis-Menten model leads to the conclusion that the relationship between v and the concentration of S in the reaction follows the following equation:

$$v = \frac{V_{\max}[S]}{K_m + [S]},$$

where $V_{\max}$ is the value of v when all of the E has been converted to ES (at saturation).

In summary, the Michaelis-Menten ODE model can be a good approximation as long as the initial assumptions hold. Implementations can be solved very

fast, and results can be displayed with a variety of plots to give visual insights into the kinetic properties. On the other hand, the parameters are difficult to estimate, and the assumptions may not hold over a sufficiently large region of interest.

The *Generalized Mass Action* (GMA) model is described by Voit [39], but it was not included in this study.

Flux-Balance Analysis (FBA) considers equations of the form:

$$\frac{\mathrm{d}[x]}{\mathrm{d}t} = Av \ (v \geq 0),$$

where $[x]$ is the m-vector of metabolite concentrations, v is the n-vector of fluxes, and A is the *stoichiometric matrix*:

$$A_{ij} = \text{change in } [x]_i \text{ induced by one unit of change in } v_j.$$

This study pursued only the equilibrium equations, $Av = 0$, with the works of Palsson et al. [32, 33, 35] (additional papers on FBA by Palsson et al. appear in the bibliography — see *Supplement*). Their approach is to use these equilibrium conditions as constraints in a linear programming model described in the next section.

3.3 Optimization

Linear programming (LP) is a form of optimization modeling that has been used to evaluate a system in which some flow of resources needs to be conserved. It is useful to view LP as a modeling technique for economic systems [30] because we can then view the biochemistry of pathways in another light. For economists, *exogenous* and *endogenous* refer to external and internal factors, respectively. This can be applied to a substructure of a cell in an analogous manner, using cell walls and other structures as borders. (For basic definitions, see the *Mathematics Glossary* on p. 40.)

An LP has two parts:

Objective that represents some quantity that we want to maximize or minimize in the system. Here are example objectives that serve as phenotype range setters:

- maximize ATP production
- minimize by-product production
- minimize mass nutrient uptake
- minimize substrate requirements

Constraints are restrictions on variables, either collectively or individually. Simple bounds on a variable's value, such as requiring that it be non-negative, are individual constraints. Collective constraints arise from conservation laws, among others. The stoichiometric equilibrium constraint, $Av = 0$, stems from conservation of mass.

Putting these together, we have

$$\text{LP:} \quad \underbrace{\text{optimize } cv}_{\text{objective}} : \quad \underbrace{Av = b,\ L \leq v \leq U}_{\text{constraints}},$$

where 'optimize' can be 'minimize' or 'maximize'. The stoichiometric equations, $Av = b$, relate the flux variables, which are required to lie within given bounds. It is possible to have $U_j = \infty$ for any j, which means that there is no upper bound on v_j. Typically, $L_j = 0$, thus simply requiring $v_j \geq 0$, but more generally, we have $L \geq 0$, possibly requiring some least value of flux that is strictly positive.

We allow $b_i \neq 0$ to represent *exchange fluxes*, which are reactions between metabolites in the network and those outside the boundary. This is illustrated in Figure 6. The exchange flux b_A is an input to the metabolite represented by node A, and flux b_E is an output from the metabolite represented by node E. External fluxes b_B and b_D are allowed to be either direction (but only one is specified in a given scenario).

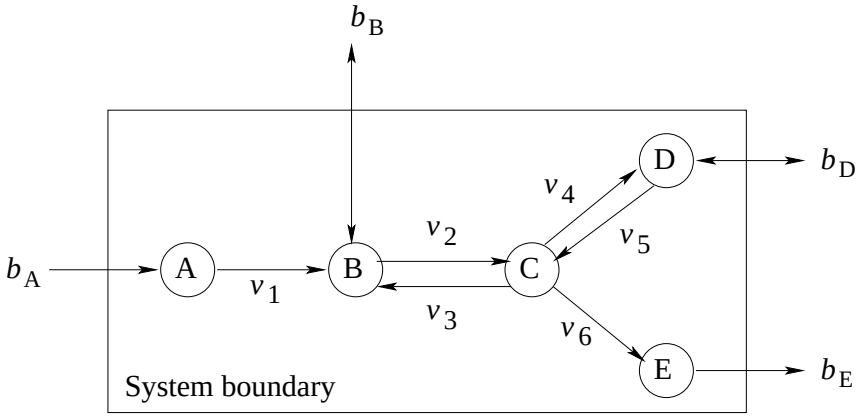


Figure 6: Illustration of Exchange Fluxes (from Schilling et al. [35])

The sign structure of A has a negative coefficient with the following convention:

$$(i) \xrightarrow{\text{process } j} (k) \quad \Leftrightarrow \quad A_{ij} < 0,\ A_{kj} > 0$$

Then, to be consistent, we require $b_A \leq 0$, $b_C = 0$, and $b_E \geq 0$. (The meaning in economic models is that $b_i < 0$ for a "supply" into i; $b_i > 0$ for a "demand" out of i; and, $b_i = 0$ for conservation (mass balance).) Here is a system of equations for this example.

$$\begin{array}{lrrrrrrrrrrrr}
\text{A}: & -v_1 & & & & & & & & & = & -1 \\
\text{B}: & v_1 & -v_2 & + & v_3 & & & & & & = & 1 \\
\text{C}: & & v_2 & - & v_3 & + & v_4 & - & v_5 & - & v_6 & = & 0 \\
\text{D}: & & & & & & v_4 & - & v_5 & & & = & -1 \\
\text{E}: & & & & & & & & & & v_6 & = & 1
\end{array}$$

This system has some of its (internal) flux variables forced: A forces $v_1 = 1$, and E forces $v_6 = 1$. In fact, all solutions have the form:

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{pmatrix} = \begin{pmatrix} 1 \\ \Delta_{23} \\ \Delta_{23} \\ \Delta_{45} \\ 1 + \Delta_{45} \\ 1 \end{pmatrix},$$

where Δ_{23} is a flow *cycle*, B $\leftrightarrow$ C, and Δ_{45} is a flow cycle, C $\leftrightarrow$ D. Such cycles would not need to be considered in a solution that optimizes some linear function of flow values. Cycles would appear in an *unbounded* objective value; for example, if we tried to maximize v_2, there would be no solution since we can make the cycle flow Δ_{23} as great as we want, using $v_3 = v_2$ to maintain the balance. Any bounded objective in this example has this optimal solution:

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}.$$

The LP model does not typically have just one solution that is optimal for all bounded objectives, especially if we allow exchange fluxes (b) to be variable.

Each constraint in an LP has an associated *shadow price*, which can be useful in measuring the marginal rate of change in the optimal objective value under perturbation of the right-side. While it is not exactly true under all solution conditions, Savinell and Palsson [32, 33] used the following relation to measure sensitivity of the solution to right-hand side perturbations:

$$\frac{\mathrm{dOPT}}{\mathrm{d}b_i} = \pi_i,$$

where OPT is the optimal objective value and π_i is the (imputed) shadow price associated with the equation, $\sum_j A_{ij} v_j = b_i$.

Similarly, there is a reduced cost associated with each variable. Only variables that equal zero can, but not necessarily will, have reduced costs that are nonzero. A reduced cost value is similar to a shadow price in that it indicates by what amount the objective function value will change if the associated variable is forced to increase.

These indicators can help us predict how inhibition of certain reactions may affect cell growth; they give us the ability to identify reactions that are not necessary for cell growth, and which may even inhibit growth.

4 Computational Issues

This section addresses the main focus of the study: computational issues in supporting pathway inference. Up to now we have described the biochemistry background and some mathematical models. Now we consider the state of computational methods.

4.1 Representation and Overall Framework

One of the goals for pathway inference is to be able to predict metabolic pathways based on biological information contained within various databases. A key challenge during development of these databases relates to sorting and categorizing the large amounts of data.

The representation that supports the ODE models is simply a system of equations. No software system that we studied contains any more information about the metabolites. The most significant attention paid to representation is Karp's ontological framework, but it does not contain any information about the kinetics.

The study by McEntire et al. [25] provides a comparative analysis across various ontologies. For our purposes, the XML-based *Systems Biology Markup Language* (SBML) [8, 9] merits further attention. This has the promise of producing a comprehensive representation, from which additional information (albeit knowledge) can be stored (generated).

In building a framework, one starts at the atomic level of data, such as what we know about proteins. Then, we add relations among data objects, and call this *information*. Adding rules for building relations gives us *knowlege*.

Then, *intelligence* is a knowledgebase system plus the capacity to acquire new knowledge — that is, *learn*. This hierarchy is shown in Figure 7.

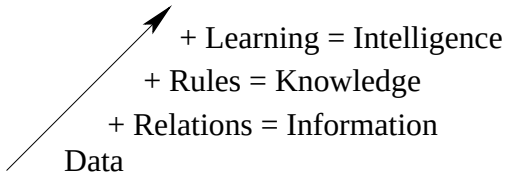


Figure 7: From Data to Intelligence

A simplified framework is shown in Figure 8. At the center are the *knowlegebases* (note the plural), which require associated ontologies to store information about genes and proteins. It is not clear how to represent interactions, but whatever that framework is, it must be designed to support network inference. The *User Interfaces* (also plural) must be able to serve different cognitive skills:

- Tabular
- Graphic
- Natural language

All of these come in different forms.

The *Algorithms Library* is likely to be partitioned, perhaps by function. It includes rapid retrieval and graph layout at one level, and machine learning algorithms at another level.

The *Knowledge Management Support* module contains all database management functions, and it must go further to deal with both internal and external knowledge. The rules for management could, themselves, be subject to automatic adaptation.

Knowledge discovery includes not only acquisition of new information, but also ongoing assessment of the accumulated knowledge. As more information about pathways becomes available, old information could be deemed incorrect. Detecting obscelescence is complex because most facts are not certain.

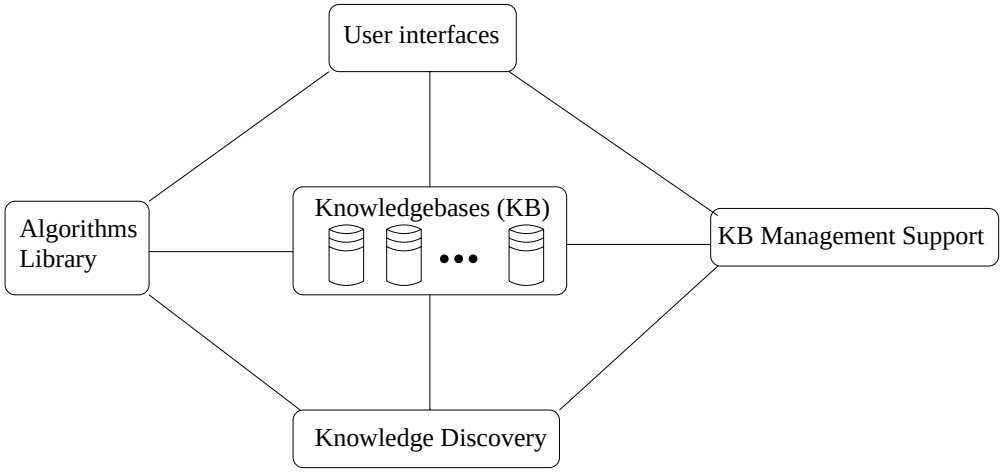


Figure 8: Hypothetical Framework

4.2 Visualization and Alternative Displays

Visualization of pathways is important because it provides information about the interaction effects. The main goal of computer visualization models is to explore ways through which to model, represent, visualize, and enable interpretation of the information associated with complex structures and dynamic processes.

In most database, the pathways were drawn manually and entered as a static object. In MetaCyc [19], pathways are drawn dynamically. In addition to the main reaction from a substrate to a product, there are secondary reactions. MetaCyc displays the secondary reaction as shown in Figure 9.

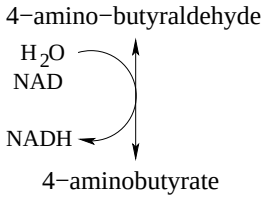


Figure 9: Showing Secondary Reaction (from MetaCyc [19])

The graphical map helps to understand a whole pathway and its interactions. These are drawn by graph layout algorithms and are very efficient [2, 6, 18]. Tabular views are also used, particularly for displaying information about individual proteins. Most systems that are run while on the internet link to the Protein Data Bank [29] (PDB) for individual protein information.

The need of a widely accepted symbol-based notation is at least desirable, and may be necessary for network visualization. Simple networks require schemes, such as shown in 9, to show the multiple inputs and outputs in one process. More generally, we want to visualize interactions among many proteins (or genes). Correlations among the levels of various metabolites can be searched to gain information about the relationships between metabolites. Such correlations are the net result of direct enzymatic conversions and of indirect cellular regulation over transcriptional or biochemical processes. Mathematically, this is the problem of viewing *hypergraphs*, and the number involved could be too much for showing some complex of arcs. Color, shape and size are yet to be considered. (Even sound might be useful to explore.)

In order to understand the main computational issues, it is necessary to understand what databases there are and the form in which they can provide information retrieval. Table 2 shows a list that we visited. (This does not include those that we evaluated, such as EcoCyc and MetaCyc, given in the next section. Also see [40].) Some of these provide drawings of pathways, but they do not necessarily provide information that is easily retrievable.

Name	Contents	Location (http:// omitted)
BIND	Full descriptions of interactions	www.bind.ca/
CSN	Cell signaling networks	geo.nihs.go.jp/csndb/
ENZYME	Enzyme nomenclature	expasy.ch/enzyme/
KEGG	Metabolic and cell signaling pathways (not necessarily from same organism)	www.genome.ad.jp/kegg/
LIGAND	Ligand chemical database for enzyme reactions	www.genome.ad.jp/dbget/ligand.html
PDB	Proteins	www.rcsb.org/pdb/
WIT	Links to pathways posted by others	wit.mcs.anl.gov/WIT2/

Table 2: Databases

Moving from pathways to more general networks presents new challenges. Uetz et al. [37] provide an excellent discussion of this and how they are developing visualization software to support network inference. Of special note is their articulation of three important needs:

1. More complex schemes to integrate data and network visualization — for example, small molecules that influence a gene's expression ought to be represented in the network, but presently they are not

2. Better network layout algorithms — for example, group nodes that represent genes of similar function
3. Broader range of information about the network objects — for example, was some interaction predicted or discovered by experiment, and what was the associated information about error?

4.3 Knowledge Discovery Methods and Concepts

A common source of information about pathways comes from gene expression array data [1, 5, 15, 21, 22]. Results from these studies do provide a powerful means to determine which proteins interact with each other; however, they do not include information on the roles or purpose of the interacting components. The analysis of this volume of data is a complex and challenging problem. Because of this, inferring pathways from interaction data alone is not enough.

Presently, models do not incorporate information about relationships among molecules, compartmentalization, and time dependencies, all of which play a very important role in the dynamics of a pathway. In general, reasoning with biological information requires flexible knowledge representation structures and powerful knowledge-discovery tools.

Datamining and Knowledge Discovery in Databases (KDD) are often used synonymously. Datamining is normally defined as the extraction of implicit, previously unknown, and potentially useful knowledge from data. We can also think of datamining as a variety of techniques used to identify important pieces of information or decision-making knowledge in databases. In datamining, it is the "hidden information" in the data that is useful to uncover.

Datamining and KDD with biological data have been implemented to serve several purposes. These include scientific discovery [17], pattern identification, organization, summarization and description, clustering [14], classifying, associating and predicting, and information extraction. The datamining process can also be used to mine research publications and use it as the primary source for building the desired databases.

Time did not permit a complete analysis of KDD techniques and their applications to support pathway inference. We therefore include this in follow-up study (§6).

5 Current Software Inventory

There are many databases on the web [40], most of which do not have accompanying software to support pathway inference. In this section we present evaluations of some that were selected by the following criteria:

- Must be free, at least for academic use
- Relevance to pathway analysis
- Installs relatively easily

The next section describes our evaluation process and terms used. The subsequent section presents a series of tables, one for each software system evaluated. We elaborate in the text about the specific evaluations.

5.1 Evaluation Process

We evaluated the software in groups, asking questions amongst ourselves in the process. The following basic questions led to the information in the tables shown in the next section.

- What is the purpose of the software, and to what extent is the system fulfilling that purpose?
- On what computing environments does the software run?
- What is the form of input and output?
- How "user friendly" is it?

We found that the software systems support more than one form of output. A particular package may return plots, coefficients, a written report, or a combination. Output in the form of a plot may be represented in two or three dimensions and may be either static or dynamically drawn. For a new user, it is necessary to have good access to help, and this affects our assessment of user friendliness.

If the software is maintained, we expect contact information to be available to someone familiar with the sight for help and further information. To decide if a software package is user friendly, we also asked, "Is it easy to navigate the site? How easy is it to edit equations and variables? Is it necessary to explore to find certain things, or is everything self-explanatory and easily accessible? What is the user interface like? Is it a GUI or is it necessary to type commands to run the software?"

We also considered how difficult it would be to use the system with limited knowledge of biology and/or of computer science. Finally, we compared

software packages, and attempted to find the things that make a particular software package different or better than others.

5.2 Evaluation Results

In this section, we present our evaluations of four systems: Gepasi, Jarnac, PLAS, and PathoLogic/EtaCyc/MetaCyc. We include a summary table for each, plus elaboration in the text, based on the process described in the previous section.

5.2.1 Gepasi

Table 3 summarizes our evaluation of Gepasi 3.0, a Windows-based software system, programmed in C++, used for modeling biochemical pathways. Its primary function is to simulate their kinetics, but it has other tools that make it very versatile.

The number of reactions and metabolites in each model is limited only by available memory. The user supplies information about the structure of the pathway, the kinetics of each reaction, and initial concentration of the substrates. The program then builds the implicit differential equations and solves them. Users can define their own reactions and kinetics in a pathway and define functions that can be used to interpret the data Gepasi gathers after a simulation is complete. Gepasi includes tools to aid modeling, including optimization and a capability to fit model parameters to data.

The report data is output in plain text and can easily be uploaded into other programs. The graphing capabilities are quite vast, allowing for both two and three dimensional plots, with the option of plotting several variables on the same graph or separate graphs.

Source	http://gepasi.dbs.aber.ac.uk/softw/gepasi.html
Function	Modeling biochemical systems
Platforms	MS Windows 95/98/2000/NT
Input	Text file with specifications; user must write personal interface to generate
Output	Text report is automatic; graph options
Documentation	[26]
Technical support	Online help at source site
User interface	GUI
Ease of use	Learning how to solve and see plots takes minutes; learning how to use other features could take a few hours; learning how to generate new input files takes days or weeks
User expertise	Familiarity with biochemical reaction notation; understand time and phase plots
Availability	Free from source site
Special features	Optimazation program, fitting models to data, several graphing options
Reviews	[27]

Table 3: Summary and Evaluation of Gepasi, Version 3.0

The inclusion of several examples that can be modified and the extensive help file make Gepasi easier for those who do not have an extensive biology background. We found Gepasi to be very user friendly, owing to its extensive help and its interface design.

5.2.2 Jarnac

Table 4 summarizes our evaluation of Jarnac, a software program used to study chemical and biochemical models. Jarnac runs in an MS Windows 95/98/2000/NT environment, using a simple, text-based modeling language, which it calls scripts. It calls itself an "interactive" system, but it parses the entire file before you can run it. Many sample scripts are included; here is how *E. coli* reactions are represented:

```
[J1]   Gluc + PEP -> PYR + G6P ; v;
[J2]   G6P => F6P ; v;
  ⋮
[J328] GLAL + ATP -> ADP + G3P ; v;
```

It contains its own editor and split screen features that allow users to view the model and output in their own chosen ratio (including all of one or the other).

There is very little online help, and many things are not intuitive, so it is necessary to read the (pdf) Reference Guide. The learning curve is based on one's advanced knowledge of metabolic systems and the ability to write programs to properly represent such systems. Once past the learning curve, Jarnac has a user-friendly environment.

There are no significant modeling aids, not even an error report for using incorrect syntax. Analysis support is limited to graphs, which have zoom and rotation controls. Jarnac is fast on small problems, but its scalability has yet to be determined. The program lacked a test suite to show correctness, and simple syntax errors went undetected.

A key feature of Jarnac is its ability to manipulate both the structure and kinetics of individual pathways at run time. Jarnac can work with populations of networks to generate pathways based on a variety of interactions as desired. Parameters are easily adjusted in the script files to observe changes in the metabolic system.

Jarnac's author intends to achieve: (1) integration with visual network designers; (2) allow networks to be built up from a library of networks; (3) expose internal functionality so Jarnac could be accessed by other systems; and (4) allow other developers to add enhancements by adding modules from other applications.

Source	http://member.lycos.co.uk/sauro/main.htm
Function	Solves ODE models of chemical and biochemical networks with numerical methods
Platforms	MS Windows 95/98/2000/NT
Input	Simple model specification language in plain text
Output	Time plot is a default, but data files are generated for alternatives
Documentation	[31]
Technical support	Contact author at HSauro@fssc.demon.co.uk
User interface	GUI
Ease of use	Non-intuitive for simple things, like running a new script; takes a few hours to get used to running its basic options and viewing output
User expertise	Knowledge of biochemical reactions and some notation; basic applications programming knowledge (less than matlab); some knowledge of numerical methods useful to understand limitations
Availability	Free from source site
Special features	Multiple graphing formats, including 3-dimensional; pan and zoom functions
Reviews	None in literature

Table 4: Summary and Evaluation of Jarnac, Version 1.19

5.2.3 PLAS

Table 5 summarizes our evaluation of PLAS (Power Law Analysis and Simulation), a tool for modeling *integrative systems* in which the kinetics can be described or approximated by power-law differential equations. Biochemical systems such as metabolic pathways can be modelled in this way.

PLAS is well-designed software to develop the model of integrative systems. It is easy to use by anyone who knows the power law differential equations and learns some rules of PLAS syntax. PLAS produces graphs (with some minor inconvenience) and text files to show the computed trajectories.

Source	http://correio.cc.fc.ul.pt/~aenf/plas.html
Function	Solve S-systems
Platforms	MS Windows 95/98/2000/NT
Input	Simple model specification language in plain text
Output	Visual graph and text delimited file
Documentation	Help file and examples in [39]
Technical support	Contact author, A.E.N. Ferreira, aenf@fc.ul.pt
User interface	GUI
Ease of use	Can be learned in minutes
User expertise	Familiar with power law differential equation
Availability	Free from source site and with [39]
Special features	Text parser
Reviews	[39]

Table 5: Summary and Evaluation of PLAS, Version 1.2 (beta)

5.2.4 PathoLogic, EcoCyc and MetaCyc

Table 6 summarizes our evaluation of PathoLogic with accompanying databases, EcoCyc and MetaCyc. The former is a bioinformatics database that describes the genome and pathways of *E. coli*, and MetaCyc is a database of pathways. The executable is created with the databases put into the PathoLogic LISP code, so a user has PathoLogic, EcoCyc, and MetaCyc in one system.

This system describes molecular activities and functions at a system level. Then, through comparison mapping, finds like pathways for different organisms with the intent to find some common metabolic pathways, applicable and predictable universally.

The source web site is useful for providing online tools and related links, but the downloaded system provides more query options, comparisons of full metabolic maps of two or more organisms, and more compounds, reactions and genes. It also highlights enzymes controlled by a specified transcription factor, it is faster, it has customizable partial gene maps, and it is programable (LISP and Perl-based API's with documentation).

Additionally, the MetaCyc database is located at http://biocyc.org/meta/ with a complete project overview and documentation; it employs the same database schema as does the EcoCyc database. MetaCyc is a review-level database that integrates information from multiple literature sources, it serves as a reference source for prediction of the pathway complement of an organism from its annotated genome. Unfortunately, it does not provide genomic maps or sequences, since the data does not come from one organism, where you can pin it down to one locus.

Source	http://biocyc.org/download.shtml-download databases as flatfiles
Function	Describes *E. coli* genes and metabolic pathways, activators, and inhibitors
Platforms	Sun Solaris, MS Windows 95/98/2000/NT, web access
Input	Select from a menu
Output	Maps, graphs, and lists
Documentation	EcoCyc-http://biocyc.org/ecoli/ and MetaCyc-http://biocyc.org/meta/
Technical support	Contact author, Peter D. Karp, pkarp@ai.sri.com
User interface	GUI
Ease of use	Simple, basic training
User expertise	Biology knowledge and database experience
Availability	Limited license from SRI, contact Peter D. Karp www.ai.sri.com/pkarp/
Special features	Ontological framework, gene/pathway visual layout is dynamic, comparison of organism maps
Reviews	[19]

Table 6: Summary and Evaluation of PathoLogic, Version 6.0

6 Strategies for Follow-up

This section is concerned with how the project might proceed and suggests avenues for further study, following the original outline.

- *What is needed (from a bio-scientist's view)?*

  While getting an overview from Dr. Imran Shah and a concrete example from Dr. Katheleen Gardiner, we do not have enough information

about this. The queries a medical researcher might have could differ significantly from a clinical researcher, even though their ultimate objectives are the same: cure or control diseases without harming the patient.

- *What can be done during the next year (from a compu-technical view)?*

  There are several possibilities, depending upon resources:

  1. Experiment more extensively with existing systems and report the results. Go more deeply into comparative analysis and obtain more relevant software.
  2. Build a framework in which existing systems can be called upon. From this determine a schema that would enable access to many different kinds of databases.
  3. Build a prototype system, drawing from what is already there, but make it separate. Consider using XML, but carefully consider the alternatives, paying special attention to the study by McEntire et al. [25]. Also consider bioPython and bioPerl and their special capabilities for accessing online databases.

- *What should the priorities be?*

  This refers to how to order the follow up, particularly if only a few people are involved.

  1. Gaining a much better understanding of the need. This includes form as well as content — who are the primary users, and what expertise do they have?
  2. Gaining a much better understanding of the current systems. This includes knowing the rationale behind their design and their strengths and weaknesses. Further, the weaknesses need to be put into the context of need. For example, how important is the absence of stochastic properties in the underlying model? What about dynamics?
  3. Gaining a much better understanding of the data needed to perform realistic pathway inference.
  4. Gaining a much better understanding of what kinds of mathematical models are realistic, given the scope of the inferences and the availability of data.
  5. Gaining a much better understanding of how knowledge discovery methods can be used to compensate for paucity of data.

- *What should be planned for the long-term, and what is that horizon?*

  This cannot yet be answered with any reasonable reliability.

- *How is biochemical pathway analysis currently used, and how will that change with technology advances?*

  This was addressed above: more study is needed to articulate this. One thing we did learn is that systems biology is the real key to the future of biology information technology. The systems approach is more recent, and has already had some important successes [13].

Overall, the follow up strategy should be one of more study, done by a group, integrating all of the above comments.

7 Glossaries

This section provides glossaries for terms found in our readings, divided into biology, computer science and mathematics.

7.1 Biology

Student contributors: Ola Adesola, Rico Argentati, Andrew Been, Felemon Belay, Jennifer Dai, Min Hong, Lance Lana, Xuan Le, Cary Miller, Tod Morrison, Dung Tien Nguyen, Adolfo Perez-Duran, Ben Perrone, Jennifer Phillips, Amy Rulo, Kimberly Somers, Jon Stranske, Xuan Tam, Christiaan van Woudenberg, Tessa Weinstein, and Rob Wilburn.

Only a fraction of the terms submitted are included here. If the all students' glossaries were combined, this section would be a large book!

activator is a substance that makes another substance active or that renders an inactive enzyme capable of exerting its proper effect.

active site is a specific region of a molecule (enzyme) where another molecule (substrate) binds and some reaction (catalysis) takes place. (It differs from just a *binding site* in that the amino acid side chains that line the active site participate in the catalytic process.)

active transport is the movement of molecules against a concentration gradient (from low to high) with the aid of proteins in the cell membrane and energy from ATP.

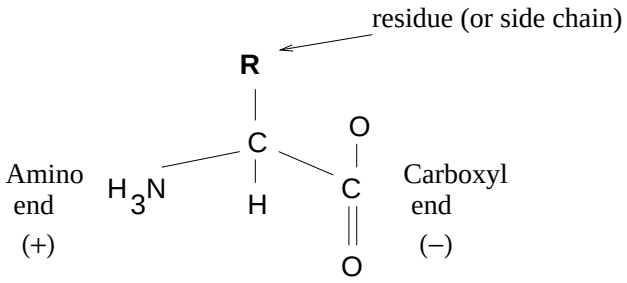
adenosine diphosphate (ADP) is a lower energy form of ATP, having two phosphate groups.

adenosine triphosphate (ATP) has an adenosine base with three phosphate groups attached to it. It is a major energy source for metabolism/cell activity, including antibody production.

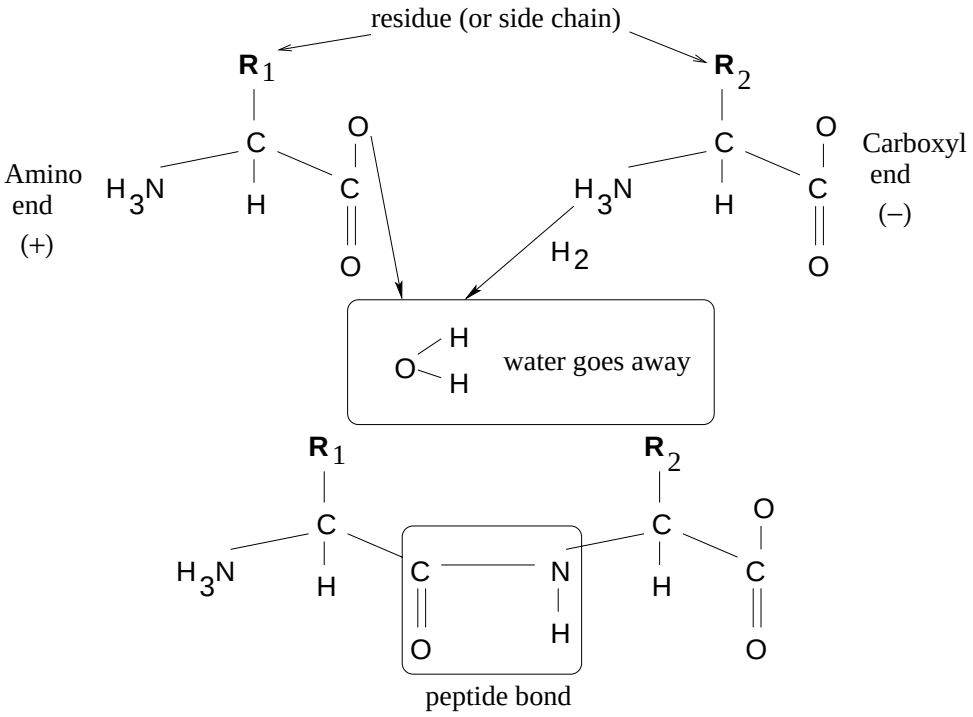
allosteric reaction is a reaction where the enzyme undergoes a conformation change due to the binding of a small molecule or ligand to a secondary site, activating the enzyme.

alternative splicing is a different way of combining a gene's exons to make variants of the complete protein.

amino acid is a molecule with the following structure:



Amino acids bond in the following manner to form a protein:



anabolism is the biosynthesis of pathways that require energy.

anticodon is a sequence of three ribonucleotides on a tRNA molecule that is complementary to a codon on the mRNA molecule.

apoenzyme is the protein component of an enzyme that lacks an essential cofactor for its activation.

apoptosis is a form of cell death necessary to make way for new cells and to remove cells whose DNA has been damaged to the point at which cancerous change is liable to occur.

binding site is an area of a molecule where an enzyme can attach itself to a compound and react with it. Binding sites are also present in antibodies, which have been genetically coded specific to a certain antigen so that the binding site can attach to the antigen in a way that depends on its structure.

catabolism is the degradation of large complex molecules into smaller, simpler molecules for the production of energy for cellular functions.

cell signaling is a method by which extracellular signals are received and converted into a response.

central dogma explains the process by which genes make proteins:

$$\text{DNA} \overset{\text{transcription}}{\Longrightarrow} \text{mRNA} \overset{\text{translation}}{\Longrightarrow} \text{proteins.}$$

These mechanisms ensure that the DNA can stay protected in the cell nucleus, allow many mRNA transcripts to be made from the same DNA strand, and perhaps most importantly, show that this process is one-way.

chromosome is the structure, composed of DNA and some proteins, that contains the genes of an organism.

***cis*-regulatory** is the regulation of a gene by an element that is from the same piece of DNA.

codon is a sequence of three nucleotides in mRNA that codes for an amino acid during protein synthesis or acts as a start or stop signal.

coenzyme is a small organic molecule required for the activation of an enzyme.

cofactor is an inorganic ion or a coenzyme that is required for the activation of an enzyme.

cooperative binding a feature in which the binding of one ligand to a target molecule facilitates the binding of other ligands.

E. coli (*Escherichia coli*) is an archetypal bacterium for biologists to study. Since many pathways are conserved between *E. coli*and other organisms, it is useful to study these pathways in a simple organism such as *E. coli*.

electron transport pathway is a mechanism by which electrons are passed along a series of carrier molecules, releasing energy for the synthesis of ATP.

EC classification is the Enzyme Commission's classification scheme of enzymes. It has four levels, with six classes at the first level:

1 Oxidoreductases

2.1-5 Transferases **2.6-8** Transaminases

3 Hydrolases

4 Lyases

5 Isomerases

6 Ligases

Some examples:

EC classification	Enzyme
1.1.1.1	Alcohol dehydrogenase
1.13.11.20	Cysteine dioxygenase
1.13.11.20	Cysteine dioxygenase
2.4.99.7	Sialyltransferase
2.6.1.1	Aspartate transaminase
3.1.1.3	Triacylglycerol lipase
4.1.1.1	Pyruvate decarboxylase
5.1.3.1	Ribulose-phosphate 3-epimerase
6.2.1.3	Long-chain-fatty-acid-CoA ligase

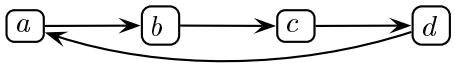
EGF is an epidermal growth factor.

enzyme is a molecule that enhances the rate of a chemical reaction. Enzymes are typically proteins, but catalytic RNA molecules are also known to exist. Each reaction in a metabolic pathway is catalyzed by a protein enzyme.

ERK is an extracellular signal-regulated kinase.

eukaryote is a cell or organism that has a cellular membrane, nucleus and other compartment structures such as mitochondrion. Most organisms fit into this class, except bacteria, virii and some types of algae.

feedback is a relation (arc) that affects an object (node) that precedes it in a time-ordering of a process. An example is



An ordering is indicated by the left-to-right arcs; the feedback is the arc from d to a. A *positive feedback* is reinforcing; a *negative feedback* is inhibiting. This applies to all types of networks or pathways: a metabolite's production could be reinforced or inhibited; a signal could could reinforce or inhibit its recipient; genes could be regulated with increased (reinforced) or decreased (inhibited) expression.

fragility is the state of being easily broken, or failed in some way. In biology, it is often the susceptibility of an organism to be harmed by an attack, such as by a parasite, that will destroy or alter it. In pathway analysis, it is a structure that is susceptible to failure. This can happen when complexity is needed for stability. An example is a feedback loop that inhibits an enzymes production, but can easily be made to do the opposite.

gene is a DNA sequence that codes for an RNA molecule, peptide, or polypeptide.

gene expression is the process by which a gene's coded information is converted into the structures present and operating in the cell. Expressed genes include those that are transcribed into mRNA, then translated into protein, and those that are transcribed into RNA but not translated into protein (there is post transcriptional modification). A gene in a diseased tissue can be *over-expressed* or *underexpressed*, which means it produces too many or too few of the protein for which it is coded.

gene expression array is a 2-dimensional collection of cDNA molecules from a large number of genes that is used to detect gene expression for each gene in the array.

gene regulation is the turning on or off of a gene or the enhancement or repression of the rate of transcription by elements that promote or repress gene expression.

genome is all DNA in the chromosomes of an organism. Its size is usually counted in base pairs.

homeostasis describes the collection of automated responses most organisms exhibit to maintain certain steady states such as temperature and oxygen uptake, most often through negative feedback.

hybridoma is a cell arising from chemical fusion of two parental cell lines. One parent is an antibody secreting cell (usually from the spleen) isolated from an immunized animal. The other parent is a myeloma cell (a type of B cell tumor). Hybridomas are immortal somatic cell hybrids that secrete antibodies.

in silico is the use of a computer to model and analyze a biological experiment.

in vitro is an experiment done outside of an organism or living cell — literally in glass (a test tube).

in vivo is an experiment done in an organism or living cell.

kinase is an enzyme that adds a phosphate group to a protein. The particular group depends upon the protein and the site.

ligand is a small molecule that binds to a particular site on a large molecule.

mitogen activated protein kinase (MAPK) pathway consists of three kinases acting consecutively, stimulated by extracellular signals.

mass spectrometry is an analytical method by which molecules are analyzed according to their mass-to-charge ratio. Mass spectrometry is often combined with other methods such as liquid chromatography to identify the type and quantity of compounds in a sample.

messenger RNA (mRNA) is the RNA product of transcription that codes for the amino acid sequence of a protein.

metabolic pathway is a series of chemical reactions in an organism where the product of one reaction in the pathway is usually the substrate for another reaction in the pathway. Some reactions can have more than one substrate as input and some have by-products.

metabolism is the total of all metabolic reactions in an organism.

metabolite is any substrate or product of a metabolic reaction.

negative feedback is the inhibition of a metabolic pathway by a downstream product, usually by competitive inhibition of an enzyme binding site.

Nuclear Magnetic Resonance (NMR) is a process that gives the structure of a section of a protein.

nuclear receptor is a protein on the nuclear membrane that binds to a specific extranuclear molecule and transfers signals thereby directing the cell to respond appropriately.

peptide is an amino acid polymer with fewer than 50 amino acid residues.

phosphorylation is a process that converts an organic compound into an organic phosphate.

polypeptide is an amino acid polymer with more than 50 amino acid residues.

prokaryote is a living cell that lacks a nucleus, such as bacteria and other types of unicellular organisms.

protein is a macromolecule that is made up of one or more polypeptides. Its *primary structure* is a sequence of amino acid residues; its *secondary structure* has *motifs*, such as α-helices, β-sheets and coils. Its *tertiary structure* is its complete 3-dimensional structure; proteins often join in homo- and hetero-dimers to form a *quaternary* structure.

promoter is an upstream sequence of DNA that signals the starting point of transcription. The TATA box is an example.

ribosome is the functional unit of translation. The two subunits composed of RNA and proteins facilitate the assembly of polypeptides from the mRNA transcript.

signal transduction is the cascade of processes by which an extracellular signal, like a hormone or neurotransmitter, interacts with a receptor at the cell surface, causing a change in the level of a second messenger. This usually effects a change in DNA transcription.

steady state is when the "state" of a system no longer changes. In an ODE model, of the form $\frac{\mathrm{d}S}{\mathrm{d}t} = f(x, S, t)$, the steady state (S) is reached at time T if $f(x, S, t) = 0$ for all $t > T$. In biochemical reactions, the steady state is when the rate of the anabolic and the catabolic processes are [approximately] equal.

stoichiometry is the mole/mass relationships between reactants and products, and the representation it provides is based on conservation of matter. For example, for the reaction

$$2A + B \rightharpoonup C,$$

the stoichiometry is that when combining two molecules of A with one molecule of B, the result is one molecule of C. The word "stoichiometry" derives from two Greek words: *stoicheion*, which means element,

and *metron*, which means measure. In pathway, analysis we have the *stoichiometric equations*:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = Sv,$$

where s is the concentration, v is the flux, and S is the *stoichiometric matrix*. It is generally held that each coefficient S_{ij} is integer-valued.

transferase is a class of enzymes that transfers some grouping from one molecule to another. For example, acyl transferases transfer acyl groups.

transcription is the process by which an mRNA strand is synthesized from the DNA coding strand in the nucleus. The mRNA transcript is then transported into the cytoplasm where it is translated into protein.

transcription factor is a protein that controls the expression of other genes.

translation is the process by which an amino acid peptide is synthesized from the mRNA transcript generated in the nucleus and transported into the cytoplasm.

transfer RNA (tRNA) is a small RNA molecule that binds to an amino acid and takes it to a ribosome for protein synthesis.

trans-regulatory is the regulation of a gene at a distance by an element that is from a different piece of DNA.

X-ray crystallography is a process that gives the structure of a crystallized protein. It gives larger scale structure than NMR, so it is not used to visualize ligand binding.

7.2 Computer Science

Student contributors: Min Hong, Xuan Le, Dung Tien Nguyen, Adolfo Perez-Duran, Ben Perrone, and Kimberly Somers. Related terms can be found in the *Mathematical Programming Glossary* [11].

algorithm is a step-by-step process of solving a problem. One example is the algorithm to do long division. The design and analysis of algorithms is a core subject in computer science; they are the instructions we give to a computer to solve a problem.

artificial intelligence (AI) is concerned with how to make computers function in a manner that requires reasoning, similar in appearance to human intelligence. This includes not only a database of facts and relations among those facts, but also a mechanism by which new knowledge can be learned and false statements can be detected and corrected.

complexity is a measure of computer time or space to execute an algorithm to solve some problem in a well defined set. This measure is defined as a function of the problem's size. Suppose $T_A(n)$ is the most time it takes for algorithm A to find a shortest path in a network with n nodes (over all possible data values). If $kn \leq T_A(n) \leq Kn$ for all $n > N$, for some constants, k, K, and N, we say that A has *linear* time complexity. More generally, if there exists p such that $kn^p \leq T_A(n) \leq Kn^p$ for all $n > N$, we say that A has *polynomial* time complexity.

A problem in a set of problems (like finding shortest paths in any network), of size n, has worst-case time complexity of $\tau(n)$ if $T_A(n) \geq \tau(n)$ for all algorithms, A, that apply to that set of problems. A problem is said to be *tractable* if there exists a polynomial time algorithm to solve it. If it is not tractable, it could be known to take an exponential amount of time to solve it, or it could be that we do not know whether there exists a polynomial algorithm to solve it, but such an algorithm could exist.

Complexity analysis typically reveals important structures that render problems tractable. It also helps us to identify *hard* problems as those that are not tractable. Finding shortest paths is a tractable problem; finding a shortest cycle that visits each node exactly once is not tractable.

data mining is the process of seeking knowledge in a database, combining statistical inference with machine learning.

deep web consists of those parts of the web that are inaccessible to a search engine. An example is PubMed's Medline.

distributed computing is when programs or databases are located on different computers but can operate together.

entity-relationship model is a representation of data by their entities and relationships. For example, entities might be *genes* and *proteins*; relationships might be *regulates* and *phosphorylates*, respectively.

frame is a collection of *slots* about an object. Slots can be specific to the object, or its value can be inherited from its *parent*. For example, the object enzyme can have a slot, isa, with value protein. This conveys

that an enzyme is a protein and will inherit properties of proteins. For example, the protein frame would have one slot isdefinedby with value amino acid sequence to convey that every protein is defined by a sequence of amino acids. Another slot might be belongsto with value pathway to convey that every protein belongs to some pathway. The inference, every enzyme must belong to some pathway, must be used with caution.

GUI stands for Graphical User Interface.

high performance computing is focused on developing supercomputers and associated software to deal with *high throughput* or complex problem-solving. Often these use a parallel architecture.

high throughput is when the volume of data is enormous, such as being generated by gene expression array chips.

inheritance is the assignment of properties held by one class to another class. For example, suppose we say that every metabolite is a protein. Then, any property we assign to a protein, such as the fact that it is composed of amino acid residues, is inherited by the metabolite: *every metabolite is composed of amino acid residues.* A common form of inheritance is in a hierarchical system, where each child inherits [some of] the properties of its parent.

machine learning is the ability for a computer to learn from experience. This stems from early works in artificial intelligence, but it is now more of an independent field, which applies to pathway inference by enabling improved performance upon acquisition of new knowledge. In particular, a missing portion of a pathway might be learned by reactions in other pathways having some similarities, or by specific experiments designed to provide information associated with various hypotheses about the missing portion of the pathway.

meta*something* is *something* about *something.* For example, *metalanguage* is a language about language, *metadata* is data about data, and *metarule* is a rule about rules.

Moore's Law is the observation made by Gordon Moore, co-founder of Intel, that data density doubles about every 18 months.

object oriented programming (OOP) is an approach that combines function and data into self-contained packages, called *classes.* These classes are organized into a *type* hierarchy. Programming languages C++ and LISP are examples where OOP is intrinsic.

ontology is a classification methodology for formalizing a subject's knowledge or belief system in a structured way. Dictionaries, encyclopedias and thesauri are examples of ontologies, but our interest is for knowledgebase design for a computer. A more elaborate definition, with types of ontologies, is given by John F. Sowa at http://www-ksl.stanford.edu/onto-std/mailarchive/0136.html.

reverse engineering is determining a original structure by observing properties from experiments. For example, metabolic pathway discovery is sometimes done by perturbation of gene expression; the inferred pathway is the product of *reverse engineering*.

semantic network is a representation of semantics in a language. One popular method is the use of frames with a slot-and-filler structure.

semantics of a language is context-dependent meanings given to phrases. An individual word could be interpreted in different syntactically correct ways, but its semantics is what determines its meaning. This is a difficult, essential part of *natural language processing*.

syntax of a language is its rules of grammar. Formally, a language is defined by an alphabet and a grammar. The syntax is how valid sentences are formed within the language, and generation or recognition of sentences is called *parsing*.

XML is an e**X**tensible **M**arkup **L**anguage that is become one a de facto standard in bioinformatics. Initially developed by the World Wide Web Consortium (W3C), XML describes a class of data objects, called "XML documents," which partially describes the behavior of computer programs that process them. More information is at http://www.w3.org/XML/.

7.3 Mathematics

Student contributors: Min Hong, Cary Miller, Dung Tien Nguyen, Ben Perrone, and Kimberly Somers. Related terms can be found in the *Mathematical Programming Glossary* [11].

Bayesian model is a mathematical representation of uncertainty with associated methods for calculating probability of an event in the presence of new evidence from a *prior* distribution One place where this is used is in determining the carrier state in Mendelian disorders by combining several independent likelihoods.

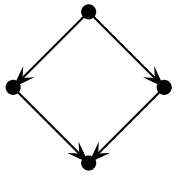
bifurcation is a division in an ODE into two *branches* where the system state could go.

boolean network is a network with a state associated with each node that is 0 or 1. One example arises in gene regulation, where a node represents a gene, and state=0 means the gene is not expressed; state=1 means it is. The arcs of such a network represents the effect one gene has on another.

clique a subgraph that is complete (every node is adjacent to every other node). A clique is *maximal* if it is not a proper subset of another clique. Every edge in a graph is a 2-clique. A triangle is a 3-clique. An m-clique is a clique with m nodes.

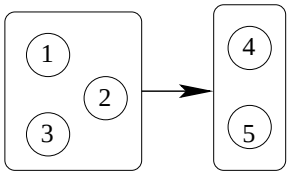
cycle in a network is a path whose first and last nodes are the same. A cycle is *simple* if the removal of any arc makes it a simple path.

directed acyclic graph (DAG) is a network without directed cycles. The following is a DAG:



hidden Markov model (HMM) is a Markov model with two states: one is observed and one is hidden. For example, consider a sequence of observed dice rolls, but the specific pair of dice could be one of two: fair or loaded. HMMs emerged in the 1970's in application to speech recognition; now they have wide application in molecular biology.

hypernetwork is like a network, except that an arc's tail or head is more generally a set of nodes, rather than a single node. The following figure shows a hypernetwork with $V = \{1, 2, 3, 4, 5\}$ and one arc with $T = \{1, 2, 3\}$ and $H = \{4, 5\}$.



Markov chain is a stochastic process such the the probability of a transition from state $s(t)$ at time t to state $s(t+1)$ at the next time period

depends only upon $s(t)$, not on the history of the process. Symbolically,

$$P\big(s(t+1) = k | s(0), \ldots, s(t)\big) = P\big(s(t+1) = k | s(t)\big).$$

This condition is known as the *Markov property* in any context involving states and their transitions. A *Markov model* is a dynamic model with the Markov property.

network is composed of a finite set of *vertices* (or *nodes*), V, a finite set of *arcs*, A, and real-valued functions, b and (H, T, C, L, U), with domains V and A, respectively. The *topology*, or *connectedness*, of the network is denoted $[V, A]$. The functions are:

$|b(v)|$ is the demand or supply at vertex v, according to whether $b(v) > 0$ or $b(v) < 0$, respectively.

$H(a), T(a) \in V$ = head and tail of arc a, respectively.

$$T(a) \xrightarrow{a} H(a)$$

It is possible to have $T(a) = \emptyset$ or $H(a) = \emptyset$, we can add dummy nodes to have the mathematically equivalent network such that $|T(a)| = |H(a)| = 1$ for all $a \in A$.

$C(a)$ = unit cost

$L(a), U(a)$ = lower and upper bounds on flow across arc a.

ordinary differential equation (ODE) describes the rate of change of a variable (x) over time:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(x, t),$$

where f is some function. Sometimes the *dot notation* is used: $\dot{x} = f(x, t)$. Usually, there is an initial (or side) condition. A *system* of ODEs consists of m such equations for n variables. The equations appear the same, but x is then an n-vector and f is an m-vector. (Usually, $m = n$, but this could be part of a larger model, such as optimal control.)

path in a network is a sequence of nodes that are successively adjacent. A path is *simple* if it does not re-visit a node.

principal components analysis is a technique for reducing the dimensionality of a dataset by using eigenvectors.

S-system is composed of the ordinary differential equations that arise from the Michaelis-Menten law for a substrate, S, and product, P:

$$\frac{\mathrm{d}S}{\mathrm{d}t} = -\frac{V_{max} S}{K + S} = -\frac{\mathrm{d}P}{\mathrm{d}t},$$

where V_{max} is the maximum rate possible, and K is a parameter.

Bibliography

The following bibliography is partially annotated. A larger bibliography appears as a supplement to this report.

[1] T. Akutsu, S. Miyano, and S. Kuhara. Identification of genetic networks from a small number of gene expression patterns under the boolean network model. In R.B. Altman, A.K. Dunker, L. Hunter, and T.E. Klein, editors, *Proceedings of the Pacific Symposium on Biocomputing* (PSB), volume 4, pages 17–28, http://www-smi.stanford.edu/projects/helix/psb01/, 1999. World Wide Web.

> The *boolean network model* infers genetic network architectures from state transition tables, which correspond to time series of gene expression patterns. This article reports the results of computational experiments, suggest that a small number of stage transition (INPUT/OUTPUT) pairs are sufficient in order to infer the original Boolean network correctly. It goes on to discuss the practical usefulness of this model.
> — Dung T. Ngyuen
>
> This paper proposes an algorithm for inferring genetic network architectures from state transition tables which correspond to time series of gene expression patterns, using the Boolean network model. It is argued that if the indegree of each node is bounded by a constant, only $O(\log n)$ state transition pairs are necessary and suficient to identify, with high probability, the original Boolean network of n nodes correctly. The paper describes the computational experiments executed in order to expose the constant factor involved in $O(\log n)$ notation. The computational results are used to show that a Boolean network of size 100,000 can be identified by their algorithm from about 100 INPUT/OUTPUT pairs if the maximum indegree is bounded by 2. The paper claims that the algorithm is conceptually so simple that it is extensible for more realistic network models.
> — Adolfo Perez-Duran

[2] M.Y. Becker and I. Rojas. A graph layout algorithm for drawing metabolic pathways. *Bioinformatics*, 17(5):461–467, 2001.

> In this article, the authors present a dynamically generated graph layout algorithm that is designed to handle cyclic, partially cyclic, linear, and branched metabolic pathways. The

authors propose an algorithm that specifically deals with the main nodes of complex pathways without graphing the side reactions. By using a recursive algorithm, the graph representing the metabolic pathway to be displayed is partitioned into subgraphs, which have simple display methods. The authors also describe the use of a spring embedding algorithm to position the primary parts of the graph relative to one another. While the algorithm was only tested on five pathways the results were promising.
Lance Lana et al.

[3] BioCarta pathways database and software. World Wide Web, http://www.biocarta.com/genes/.

[4] P. D'haeseleer, S. Liang, and R. Somogy. Tutorial: Gene expression data analysis and modeling. In R.B. Altman, A.K. Dunker, L. Hunter, and T.E. Klein, editors, *Proceedings of the Pacific Symposium on Biocomputing* (PSB), volume 4, http://psb.stanford.edu/psb99/genetutorial.pdf, 1999. World Wide Web.

This is a succinct introduction to the entitled topics. the color figures are very helpful in giving insight into gene expression data and the associated issues with the high dimensionality.

[5] P. D'haeseleer, S. Liang, and R. Somogyi. Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics*, 16(8):707–726, 2000.

This paper focuses on generating models that allow them to systematically derive predictions about important biological processes in disease, development and metabolic control. They use clustering of co-expression profiles, which allow them to infer shared regulatory inputs and functional pathways. The reverse engineering has the goal of identifying the causal relationships among gene products that determine important phenotypic parameters. Using the network inference, the goal of this project is to construct a coarse-scale model of the network of regulatory interactions among the genes.
— Xuan Le

[6] A. Enright and C. Ouzunis. BioLayout — an automatic graph layout for similarity visualization. *Bioinformatics*, 17(9):853–854, 2001.

The authors present a force-based graph layout algorithm solved with simulated annealing. They employ this rather standard algorithm to visualize protein families. Nodes represent proteins and repel each other. However, proteins that have sequence similarities have an edge between them that exerts an attractive force. After about 50 iterations of the algorithm, similar proteins end up clustered together. They also implement a 3-dimensional version that uses OpenGL. However, it appears this version lacks labeling facilities

[7] A.E.N. Ferreira. PLAS, Version 1.2. with Voit [39] and at http://correio.cc.fc.ul.pt/~aenf/plas.html, 2000.

[8] A. Finney, V. Gor, B. Bornstein, E. Mjolsness, and H. Bolouri. Systems biology markup language (SBML) level 2 proposal: Array features. Technical report, California Institute of Technology, http://www.cdb.caltech.edu/erato/, 2002.

[9] A. Finney, V. Gor, B. Bornstein, E. Mjolsness, and H. Bolouri. Systems biology markup language (SBML) level 2 proposal: Miscellaneoous features. Technical report, California Institute of Technology, http://www.cdb.caltech.edu/erato/, 2002.

[10] M.A. Gibson and E. Mjolsness. Modeling the activity of single genes. In J.M. Bower and H. Bolouri, editors, *Computational Modeling of Genetic and Biochemical Networks*, pages 1–48, Cambridge, MA, 2001. MIT Press.

This is a thorough and clear introduction to what genes are and how they function. It is not light reading, but it is especially useful in beginning to gain a non-superficial understanding of the underlying biology and biochemistry of such pathway inference issues as cell signaling.

[11] H.J. Greenberg. *Mathematical Programming Glossary*. World Wide Web, http://www.cudenver.edu/~hgreenbe/glossary/, 1996–2002.

[12] L. Hunter, editor. *Artificial Intelligence and Molecular Biology*, Cambridge, MA, 1993. MIT Press.

This is now available at http://www.aaai.org//Library/Books/Hunter/hunter.html. Chapter 1, by the editor, provides a good introduction to biology for computer scientists and mathematicians. This is highly recommended for a gentle, informative introduction. Other chapters of direct relevance are by Karp [16] and Mavrovouniotis [24].

[13] T. Ideker, T. Galitski, and L. Hood. A new approach to decoding life: Systems biology. *Annual Reviews of Genomics and Human Genetics*, 2:343–372, 2001.

[14] T. Ideker, V. Thorsson, J. Ranish, R. Christmas, and J. Buhler. Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Science*, 292:929–934, 2001.

[15] T.E. Ideker, V. Thorsson, and R.M. Karp. Discovery of regulatory interactions through perturbation: Inference and experimental design. Technical report, Institute for Systems Biology, Seattle, WA, 2000.

[16] P.D. Karp. A qualitative biochemistry and its application to the regulation of the tryptophan operon. In L.E. Hunter, editor, *Artificial Intelligence and Molecular Biology*, pages 289–324, Cambridge, MA, 1993. MIT Press.

This paper discusses the representation and simulation of biological information so that it can be used in various situations. This is done through the particular example of a bacterial gene regulation system, the tryptophan operon of *E. coli*. The possibility of improving simulation programs that predict the outcome of a gene regulation experiment is explored. The GENISM simulator is held up as a simulator that will efficiently do this. Karp explores its functions, possibilities and limitations. Some results of simulations run through GENISM are presented.
— Jennifer Phillips

The focus of this chapter is on the issues of representation and simulation of the gene regulation system of the trytophan operon of *E. coli*. The author presents a model, GENSIM, which describes the biochemical reactions that determine

the expression of the genes, the reactions by which the genes direct the synthesis of enzymes, and the reactions catalyzed by these enzymes. He then presents a detailed discussion of the implementation of this model.
— Tod Morrison

[17] P.D. Karp. Pathway databases: A case study in computational symbolic theories. *Science*, 293:2040–2044, 2001.

This article introduces the concept of pathway genome database (PGDB) as a method of describing biochemical pathways and their component reactions, enzymes and substrates. A PGDB includes pathway information as well as information about the complete genome of the organism. The EcoCyc project is provided as an example of a PGDB. EcoCyc is structured using an ontology of about 1000 classes. The PGDB consists of a network of interconnected frames. Each frame represents a biological object. The labeled connections between the frames represent semantic relationships between the objects. The key to this representation is devising an ontology that clearly defines the meaning of the different PGDB fields and provides ease of extension when new domain concepts are discovered. The authors include a discussion of a program called "PathoLogic," which was developed to predict the metabolic network from the genome of an organism. Prediction of pathway flux rates for the entire metabolic network of an organism is also discussed. The authors stress the importance of using database content in solving these computational problems. There are no known algorithms that can solve these problems without being coupled with an accurate and well-designed pathway DB.
— Lance Lana et al.

[18] P.D. Karp and S.M. Paley. Representations of metabolic knowledge: Pathways. In R. Altman, D. Brutlag, P. Karp, R. Lathrop, and D. Searls, editors, *Proceedings of the Second Intelligent Systems for Molecular Biology Conference* (ISMB), pages 203–211, Menlo Park, CA, 1994. AAAI Press.

Karp and Paley discuss the formulation of a database schema and a set of accompanying software tools which form the knowledge base known as EcoCyc. The core problem is one of path representation; how can one most efficiently store pathway data and still construct complete pathways? The authors

discuss the problems encountered during their formulation of the knowledge base and propose solutions to achieve a complete reconstruction of the known pathways in *E. coli*. Relationships between reactions is stored in a predecessor list, containing tuples of a reaction and its predecessor compound in the pathway. Main versus side compounds and direction in individual reactions is determined by the predecessor list along with a list of heuristics.
— Raphael Bar-Or et al.

In this article Karp and Paley tackle the difficult problem of using information in the EcoCyc knowledge base (KB) to obtain a pathway graph, with the ultimate goal of using the pathway graph to create a pathway map. The EcoCyc knowledge base is a database (DB) that has information pertaining to the genes and intermediary metabolism of *E. coli*. Specifically, it contains information about the genes, enzymes, reactions and chemical compounds that participate in the metabolic pathways in *E. coli*. Their goal was to use a minimal amount of information from the KB to accomplish this so that the KB would be easy to maintain and update. The primary contribution they made to this end was what the authors call a predecessor list, which effectively gives reactions in a pathway order. Using the predecessor list and information stored in the KB about reactions, the enzymes that catalyze those reactions and the chemical compounds that are active in those reactions, Karp and Paley develop an algorithm based on production rules and heuristics to turn a predecessor list into a pathway graph. To this end they are somewhat successful.
— Tessa F. Weinstein

In this article the authors present an automated graph layout algorithm that dynamically draws given pathways present in the EcoCyc database. The algorithm determines the topology of the pathway as being cyclic, linear, or branched. Larger groupings of such pathways are handled by predefined layouts that are applied to the subgraphs within the pathway. Facilities for navigating, expanding and collapsing pathways within the user interface are discussed. Complex junctions and superpathway algorithms are also discussed. It is apparent that one of the goals of the algorithm is depth of representation, but not necessarily breadth.
Lance Lana et al.

[19] P.D. Karp, M. Riley, M. Saier, I.T. Paulsen, S.M. Paley, and A. Pellegrini-Toole. The EcoCyc and MetaCyc databases. *Nucleic Acids Research*, 28(1):56–59, 2000.

The authors provide an update on the EcoCyc and MetCyc knowledge bases, now unified under a common software toolbox called "The Pathway Tools," released as version 5.0 at the time of publication. While EcoCyc attempts to completely document the metabolic map of *E. coli*, MetaCyc is designed as a reference for metabolic pathways in various organisms, without the detailed genetic information provided in EcoCyc. The breadth of information contained in EcoCyc has been expanded to include membrane transport systems with the same level of detail afforded to the metabolic pathways represented in EcoCyc already. This allows researchers to query the relationships between metabolic pathways and transport systems at the cellular level. MetaCyc is built on the framework of EcoCyc, but it is expanded to contain species information for each pathway reaction. It does not contain any genetic map information.
— Raphael Bar-Or et al.

This article focuses on describing the information contained in, and the available forms of query to access, the EcoCyc and MetaCyc databases. EcoCyc is a database (DB) that contains biochemical information about *E. coli*, such as signal transduction pathways, transports, and its genes. A recent addition to this DB is information regarding membrane transport systems. MetaCyc, on the other hand, aims to describe metabolic pathways from a variety of different species. The information it contains about pathways includes reactions, enzymes and substrate components. However, it does not include information about transport processes like the EcoCyc database does.
— Tessa F. Weinstein

[20] Kyoto encyclopedia of genes and genomes (KEGG). World Wide Web, http://www.genome.ad.jp/kegg/.

This database contains pictures of pathways, hyperlinked to give information about the parts. Basic references to the literature are cited, and some are available as pdf or postscript files.

[21] S. Liang, S. Fuhrman, and R. Somogyi. REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. In R.B.

Altman, A.K. Dunker, L. Hunter, and T.E. Klein, editors, *Proceedings of the Pacific Symposium on Biocomputing* (PSB), volume 3, pages 18–29, http://www-smi.stanford.edu/projects/helix/psb98/, 1998. World Wide Web.

The REVerse Engineering ALgorithm (REVEAL) was implemented as a C program to infer a complex regulatory network architecture from input/output patterns of its variables. The algorithm was generalized to include multi-state models, essentially allowing direct application to realistic biological data sets. A genetic network analysis tools can be designed based on generating model systems on which the performance of the tools can be tested on. An example of such model system is the Boolean network. In this network, the wiring of the elements to one another correspond to functional links between genes, and the rules determine the result of a signaling interaction given a set of input values. Genes are expressed as either off or on, which results in binary elements interacting according to Boolean rules.
— Olasumbo Olufunke Adesola

This article attempts to answer the question of whether it is possible in principle to completely infer a complex regulatory network architecture from the input/output patterns of its variables. Using state transition tables to represent gene expression patterns, Information Theory (Shannon Entropy) is used to determine the wiring relationships of the extended genetic network. A program called REVEAL does this, enabling the inference of inputs which control genes in the network.
— Rob Wilburn

[22] Y. Maki, D. Tominaga, M. Okamoto, S. Watanabe, and Y. Eguchi. Development of a system for the inference of large scale genetic networks. In R.B. Altman, A.K. Dunker, L. Hunter, and T.E. Klein, editors, *Proceedings of the Pacific Symposium on Biocomputing* (PSB), volume 6, pages 446–458, http://www-smi.stanford.edu/projects/helix/psb01/, 2001. World Wide Web.

This paper presents an approach to the inference of interrelated mechanisms among genes in a genetic network based on the analysis of gene expression patterns. The approach developed in this paper actually uses a combination of two previously developed methods. The first analysis is performed by a static Boolean network model based on a multi-level di-

graph approach. The second uses a dynamic network model, the S-system that relies on the analysis of temporal responses of gene expression patterns against perturbations or internal changes. The weakness in the Boolean network model is that relations between genes that affect each other cannot be determined. These genes are assigned to equivalence classes and then the dynamic network model is used to determine the relations within these equivalence classes. The Boolean network model can infer large genetic networks (10,000+ genes) in less than a second, but it cannot determine the relations between genes that belong to the same equivalence classes. The dynamic model based on the S-system can infer the network even if there are equivalence classes, but the run time is $O(n^2)$ so this model is not practical to use with large networks. The authors approach uses the Boolean network model to reduce the size of the network into functional units that the dynamic model can solve in a "reasonable" amount of time.
— Lance Lana et al.

[23] E. Martz. *Beginner's Guide to Molecular Biology.* World Wide Web, http://www.iacr.bbsrc.ac.uk/notebook/courses/guide/, 2001.

This is a simplified introduction, with online animations to help understand mainstream topics in molecular biology. (The animations are with RasMol, which is free software that runs in an MS Windows environment.)

[24] M.L. Mavrovouniotis. Identification of qualitatively feasible metabolic pathways. In L.E. Hunter, editor, *Artificial Intelligence and Molecular Biology*, pages 325–364, Cambridge, MA, 1993. MIT Press.

This chapter describes an algorithm for the synthesis of biochemical pathways. Biochemical pathway synthesis is the construction of pathways which produce certain target bioproducts, under partial constraints on the available reactants, allowed by-products, etc. Given a set of stoichiometric constraints and a database of biochemical reactions, this algorithm transforms an initial set of available bioreactions into a final set of pathways by and *iterative* satisfaction of constraints. After explaining the design of the algorithm, the author presents a case study of its application to study of the synthesis of biochemical pathways for the production of lysine from glucose and ammonia.
— Tod Morrison

This article discusses the use of an AI method for finding quantitatively feasible metabolic pathways. In order to quantify a pathway's feasibility, the method uses information on the types and amounts of enzymes, ratios of metabolites, and the likelihood of a reactions occurrence in a particular direction within the pathway. The chapter discusses how the AI algorithm works and gives an abstract problem as an example.
— Rob Wilburn

[25] R. McEntire, P. Karp, N. Abernethy, D. Benton, G. Helt, M. DeJongh, R. Kent, A. Kosky, S. Lewis, D. Hodnett, E. Neumann, F. Olken, D. Pathak, P. Tarczy-Hornoch, L. Toldo, and T. Topaloglou. An evaluation of ontology exchange language for bioinformatics. In P. Bourne, M. Gribskov, R. Altman, N. Jensen, D. Hope, T. Lengauer, J. Mitchell, E. Scheeff, C. Smith, S. Strande, and H. Weissig, editors, *Proceedings of the Eighth Intelligent Systems for Molecular Biology Conference* (ISMB), pages 239–250, Menlo Park, CA, 2000. AAAI Press.

This paper compares some candidate ontology-exchange languages to find the one that best deals with a variety of issues. Ontology exchange languages should exchange using a standardized form that has well-described syntax and semantics to make the sharing of information effective. If the database uses a well-defined ontology, then it can convey more accurate nuances of purpose. On the other hand, coarsely-defined ontologies will convey only superficial facets of information.

This paper compares the following ontology exchange languages: Ontolingua, CycL, OML/CKML, OPM, XML/RDF, UML, ASN.1, and ODL with the following ideal criteria: Language support and standardization, data model/capabilities, performance, pragmatics, and connectivity. It turns out Ontolingua and OML/CKML have enough expressivity, however, Ontolingua does not have XML expressions and OML is not a framed-based system, the author recommends a new language XOL (XML Ontology Language) that has frame-based semantics with XML expressions the author feels that XML is important due to the proliferation of the web and the widespread availability of parsers.
— Min Hong

[26] P. Mendes. *Computer Simulation of the dynamics of biochemical pathways*. PhD thesis, University of Wales, Institute of Biological Sciences,

Aberystwyth, Wales, 1994.

[27] P. Mendes and D. Kell. Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. *Bioinformatics*, 14(10):869–883, 1998.

This article discusses the suitability of various optimization methods, used as part of simulation software, to study the kinetics of biochemical reactions. The authors focus on analyzing the ability of these algorithms to find global minima. They recommend that a suite of diverse optimization methods should be available in simulation software, as no single one performs best for all problems. They discuss how they have implemented such a simulation-optimization strategy in the biochemical kinetics simulator Gepasi (http://gepisi.dbs.aber.as.uk/softw/Gepasi.html). They provide an overview of optimization methods and discuss the importance of finding the global minimum. They also discuss computational issues that arise from the nonlinearity. Applications and numerical results are discussed for two areas: simulation of a hypothetical branched biochemical pathway with conserved cofactor and feedback, and parameter estimation. The following methods are compared: L-BFGS-B, Levenberg-Marquardt, Steepest descent, Simulated annealing, Multistart, Random search, Truncated Newton, Evolutionary programming and a genetic algorithm. Many of these methods can be implemented without explicit calculation of derivatives.
— Rico Argentati

[28] *MIT Biology Hypertextbook*. Massachusetts Institute of Technology, http://esg-www.mit.edu:8001/esgbio/7001main.html, latest edition, 2002.

This has undergone maturation since its first posting, and it is a fairly complete introductory resource. The chapters of most direct benefit to understanding pathways are *Chemistry Review, Large Molecules, Cell Biology, Enzyme Biochemistry, Glycolysis and the Krebs Cycle,* and *Prokaryotic Genetics and Gene Expression.* (Of course, the other chapters are also important.) You can stretch your knowledge by working their "practice problems."

[29] Protein data bank (PDB). World Wide Web, http://www.rcsb.org/pdb/.

This database contains protein data, including annotations. It highlights a "Molecule of the Month."

[30] T.J. Rothenberg. Simultaneous equations models. In J. Eatwell, M. Milgate, and P. Newman, editors, *The New Palgrave: Econmetrics*, pages 229–237, New York, NY, 1990. W.W. Norton & Company.

The book is a series of invited articles by noted econometricians about subjects basic to the theory and doctrine of the field of econometrics. The discussion on this chapter on simultaneous equations gives a clearer view as to how econometricians use this method and the limitations of this method in modelling real economic phenomena. The author conisiders exogenous and endogenous factors to be important in this analysis as tools for building the model. This can be compared to external and internal factors in the behavior of the cell, or some other biological systems.
— Andrew Been

[31] H.M. Sauro. Jarnac: A system for interactive metabolic analysis. In J-H. S. Hofmeyr, J. M. Rohwer, and J. L. Snoep, editors, *Animating the Cellular Map*, 9th International BioThermoKinetics Meeting, pages 221–228, http://www.sun.ac.za/biochem/btk/book/Sauro.pdf/, 2000. Stellenbosch University Press.

[32] J.M. Savinell and B. Ø. Palsson. Network analysis of intermediary metabolism using linear optimization I: development of mathematical formalism. *Journal of Theoretical Biology*, 154:421–454, 1992.

The article addresses the benefits of using a stoichiometric matrix as a tool for understanding metabolic behavior of a cell its relationship with external compounds. The metabolic pathways considered in the matrix were the mass and energy factors. Shadow prices acted as the parameters on the matrix, which ultimately helped in identifying any growth limitations. Attempts at linearly optimizing the matrix revealed important cell behavior in its processing of elements, such as carbon, for growth factors. Glucose, glutamine, and glutamate factors were the primary focus upon linear optimization.

The linear optimization returned information relating to energy use, growth, and ratios under given conditions such as limited oxygen uptake or minimized production of NADH. Using linear optimization techniques proved successful when attempting to minimize certain productions, which then offered insight to inhibiting some cell functions or deleting genes altogether. Overall, the stoichiometric matrix for analyzing the cell behavior related to metabolic pathways returned good results and offered great promise for further study.
— Kimberly A. Somers

[33] J.M. Savinell and B. Ø. Palsson. Optimal selection of metabolic fluxes for *in vivo* experimental determination I: development of mathematical methods. *Journal of Theoretical Biology*, 155:201–214, 1992.

This paper presents a framework for determining internal fluxes for highly branched metabolic networks. The problem is formulated as a linear system which contains both measured and calculated fluxes. The goal of this particular work is to determine an optimal set of fluxes to measure so that the calculated values are as accurate as possible. This is done by anayzing the sensitivity of the system (bounded by a condition number) to the choice of measured vs. calculated fluxes. The authors find an estimation to the optimal set of measured fluxes using a shotgun approach and observing the shape of the scatter plot produced.
— Raphael Bar-Or et al.

[34] F. Schacherer. *An object-oriented database for the compilation of signal transduction pathways*. PhD thesis, Technical Universität Braunschweig, Braunschweig, FRG, 2001.

[35] C.H. Schilling, D. Letscher, and B. Ø. Palsson. Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. *Journal of Theoretical Biology*, 203:229–248, 2000.

Uses the well-established, super-powerful techniques of linear algebra to analyze metabolic pathways. Makes use of a subset of all possible metabolic pathways available in a system to define the 'metabolic flux cone'. This is the set of all conceivable metabolic phenotypes. To make the analysis tractable to linear algebraic methods makes the simplifying assumption

that reactions and pathways are at a steady-state. This assumption is not totally justifiable but is common in modeling of metabolic pathways. The paper does not address pathway regulation. Some of the descriptive linear "equations" are actually inequalities so concepts of linear programming or convex analysis are used.
— Cary Miller et al.

[36] Z. Szallasi and S. Liang. Modeling the normal and neoplastic cell cycle with "realistic boolean genetic networks": Their application for understanding carcinogenesis and assessing therapeutic strategies. In R.B. Altman, A.K. Dunker, L. Hunter, and T.E. Klein, editors, *Proceedings of the Pacific Symposium on Biocomputing* (PSB), volume 3, pages 66–76, http://www-smi.stanford.edu/projects/helix/psb98/, 1998. World Wide Web.

[37] P. Uetz, T. Ideker, and B. Schwikowski. Visualization and integration of protein-protein interactions. In E. Golemis, editor, *The Study of Protein-Protein Interactions — An Advanced Manual*, Cold Spring Harbor, 2002 (to appear). Cold Spring Harbor Laboratory Press.

The introduction compares protein interaction maps with metabolic pathways and describes the kinds of information in need of visual displays. Besides the usual zoom and pan functions, they point out the need for graph condensation and expansion, raising issues of recognizable protein classifications. They proceed to descibe the symbolic syntax of their displays and handling supplemental data. In their "Future Directions" the authors point to three important needs:

1. More complex schemes to integrate data and network visualization — for example, small molecules that influence a gene's expression ought to be represented in the network, but presently they are not
2. Better network layout algorithms — for example, group nodes that represent genes of similar function
3. Broader range of information about the network objects — for example, was some interaction predicted or discovered by experiment, and what was the associated information about error?

[38] C. van Gend and U. Kummer. STODE — automatic stochastic simulation of systems described by differential equations. Technical report,

Bioinformatics and Computational Chemistry Group, European Media Laboratory, Schloss-Wolfsbrunnenweg 33, D-69118 Heidelberg, FRG, 2001.

[39] E.O. Voit. *Computational Analysis of Biochemical Systems.* Cambridge University Press, Cambridge, UK, 2000.

[40] U. Wittig and A. De Beuckelaer. Analysis and comparison of metabolic pathway databases. *Briefings in Bioinformatics*, 2(2):126–142, 2001.

This is a useful, albeit very brief, description of current pathway databases (most of which are static and have no computational methods to support inference): CSNDB, EcoCyc/MetaCyc, ExPASy — Biochemcial Pathways, KEGG, PATHDB, SPAD UM-BBD. (It does not include others, such as BioCarta [3].)