Modeling the *E. coli* cell: The need for computing, cooperation, and consortia

Barry L. Wanner, Andrew Finney, Michael Hucka

Abstract

Escherichia coli K-12 is an ideal test bed for pushing forward the limits of our ability to understand cellular systems through computational modeling. A complete understanding will require arrays of mathematical models, a wealth of data from measurements of various life processes, and readily accessible databases that can be interrogated for testing our understanding. Accomplishing this will require improved approaches for mathematical modeling, unprecedented standardization for experimentation and data collection, completeness of data sets, and improved methods of accessing and linking information. Solving the whole cell problem, even for a simple *E. coli* model cell, will require the concerted efforts of many scientists with different expertise. In this chapter, we review advances in (i) computing for modeling cells, (ii) creating a common language for representing computational models (the Systems Biology Markup Language), and (iii) developing the International *E. coli* Alliance, which has been created to tackle the whole cell problem.

1 Introduction

Biology has come a long way since Robert Hooke first used the term "cell" to describe the basic structural unit of cork in 1665. The tiny, room-like structures he saw under his microscope had solid walls, but they were empty because the cork was dead. Today, we can describe in exquisite detail many of the molecular parts and processes that furnish biological rooms. The complete genetic blueprints are available for common and deadly microbes, for economically important animals and plants, and even for human beings. And yet, we still lack a comprehensive understanding of any living cell.

One of the most striking successes of twentieth-century biology has been the identification and characterization of the molecules of life. This has been brought about through the development of disciplines such as biochemistry, biophysics, cell biology, molecular biology, molecular genetics, structural biology, and others. A major challenge of the twenty-first century is to describe the dynamic interactions of these molecules of life in the complex processes that are the essence of a living cell. Meeting this challenge requires that we enhance the highly successful, but limiting, reductionist approaches of the last several decades by revisiting

Topics in Current Genetics
L. Alberghina, H.V. Westerhoff (Eds.): Systems Biology
DOI 10.1007/b138743 / Published online: 25 May 2005

themes first articulated early in the twentieth century by systems-oriented thinkers such as Bogdanov and Bertalanffy (Capra 1996), and somewhat later by Wiener, Kacser, Mesarovic, and others (Wiener 1961; Kacser 1957; Mesarovic 1968). Fundamentally, systems biology strives to augment reductionist molecule-by-molecule accounts of cells by embedding the components within accounts of the broader context of cells and cell systems. This approach is simply an acknowledgment that dynamical interactions between components give rise to new functional properties at all levels from the genome, to molecular modules and networks, up through entire cell systems and beyond, and that these interactions are measurable and quantifiable.

These themes are the core of the definition of systems biology by Alberghia and Westerhoff (this volume) and others (Hood 1998; Ideker et al. 2001; Kitano 2001, 2002). The contemporary resurgence of interest in systems biology can be attributed to at least three major factors. First, there is the explosion of data brought about by modern molecular techniques and the commensurate realization by many researchers that future progress in understanding biological function rests inescapably in the development and application of computational methods (Alm and Arkin 2003; Arkin 2001; Fraser and Harland 2000; Hartwell et al. 1999; Noble 2002; Tyson et al. 2001; Zerhouni 2003). Second, there is the vastly greater power afforded by modern information technology (Butler 1999), beckoning us to reattempt solutions to problems that were beyond reach in the mid-twentieth century. And third, only in recent decades has the mathematical theory of nonlinear systems and stochastic systems advanced sufficiently to allow us to handle the classes of systems that emerge naturally when describing complex biological processes in a detailed, mathematical fashion (Burns 1971; Gillespie 1977; Gillespie and Petzold 2003; Kacser and Burns 1967; Savageau 1969, 1970).

The contrast between our tremendously increased computational and biological powers and technologies on the one hand, and our continuing lack of understanding of any whole cell on the other, has been the major inspiration for the formation of the IECA—the International *E. coli* Alliance (Holden 2002). The mission of this alliance has been to coordinate global efforts to understand a living bacterial cell, the K-12 strain of *Escherichia coli* that has laid so many of the golden eggs of basic biochemistry, genetics, and molecular biology in the last half of the twentieth century (Kornberg 2003). Scientists around the world are working together to create a computer model of *E. coli*, integrating all of the dynamic molecular interactions required for the life of a simple, self-replicating cell. A whole cell model of *E. coli* would not only significantly advance the field of biology; it would also have immediate practical benefits as well, for everything from drug discovery to bioengineering.

The IECA effort is emblematic of systems biology as a whole. The ambitious goal of the IECA is beyond the means of any single investigator or laboratory (Crick 1973). It requires an integrative research program and collaborations between scientists with expertise in biology, chemistry, computer sciences, engineering, mathematics, and physics. Success will depend crucially on bringing to bear both social and technological tools: namely, consortia that help forge collaborations and common understanding, computational tools that permit analysis of vast and complex data, and agreed-upon standards and tools that enable researchers to communicate, integrate, and use their results in practical and unambiguous ways.

In this chapter, we discuss these topics in the context of the IECA effort. We begin in Section 2 by describing the kind of models ultimately sought by Systems Biologists and provide an overview of computational modeling. In Section 3, we survey some of the software tools available today to help with computing in systems biology and follow this in Section 4 with a discussion of the Systems Biology Markup Language (SBML) and its role as an enabling technology for modelers to share their models. Section 5 briefly describes the kinds of experimental standards envisioned by IECA that will be required for successful whole cell modeling. One way of carrying out such standardized experiments as a community is also given in Section 5. It is unrealistic and probably unwise to develop a single database encompassing all information, even for a single cell. Section 6 describes an alternative approach for creation of an accessible and interoperable database that would not only store massive amounts of data in different formats but would also have the capability of interrogating other meaningful databases. Consortia such as the International *E. coli* Alliance have been created as one way to meet this challenge (see Section 7). We close by bringing the discussion back to the IECA effort itself. Several contributors to this book are also participating in the IECA.

2 Quantitative, formal models are essential instruments in systems biology

Models, as abstractions representing observed or hypothesized phenomena, are nothing new to the life sciences, having long been used by life scientists as tools for organizing and communicating conceptual and factual information. However, the majority of models in biology traditionally have been expressed in natural language narratives, sometimes augmented with block-and-arrow diagrams (Bower and Bolouri 2001a). These certainly can be useful and important for describing hypotheses about a system's components and their interactions, but these types of models also have crucial limitations that make them inadequate as vehicles for describing and understanding large and complex systems (Bialek and Botstein 2004). A block-and-arrow diagram combined with verbal explanations and statements about observed effects, quantities of substances involved, and so forth, may appear detailed and precise, but in practice it leaves too much room for ambiguity, misinterpretation, and hidden complexity. More importantly, as Phair and Misteli wrote:

“... it often remains difficult to make quantitative predictions for a given experimental protocol with the use of diagrams alone. Scientific intuition has been successful when systems are limited to a few molecules and processes, but today’s summary diagrams generally have many more molecules and arrows than this, and even simple systems often behave in surprising ways. What is needed is a way to know what the diagram predicts in a given experiment” (Phair and Misteli 2001).

This is not to say that narrative descriptions and diagrams should be abandoned; rather, they should be used as stepping-stones, not stopping points. Scientists must go further and express their models in such a way that each molecular entity is knowable and quantifiable in terms of empirical evidence and each process is expressed step-by-step in a formal, mathematical language. It is by systematizing how entities and processes are defined, represented, manipulated and interpreted, that *formal, quantitative models* can enable “meaningful comparison between the consequences of basic assumptions and the empirical facts” (May 2004).

2.1 Computational modeling is an extension of the scientific method

Computational models are simply formal models expressed in a form that can be manipulated by a computer. The resulting descriptions are more likely to be coherent and internally consistent, because computable representations must be precise and detailed—vague or incomplete elements will not do, else it will not be possible to simulate or analyze the model. While frustrating at times, this is exactly the reason why computational models are an invaluable tool in helping us understand phenomena. Only if one can express every step of a process in such detail that it can be simulated in a computer program can one justifiably claim to understand it very well. This is the fundamental premise for doing computational modeling in biology, and other fields.

Computational models also allow quantitative calculations to be done on a model, allowing researchers not only to test their understanding, but also to explore “what-if” scenarios and make testable predictions about the behavior of the system being studied. This is an essential requirement for being able to understand complicated systems that are replete with feedback mechanisms (the hallmark of biological systems), where the resulting behaviors are rarely predictable through intuitive reasoning alone. Even for the simplest components and systems, it can be impossible to predict such characteristics as sensitivity to exact parameter values without constructing and analyzing a model. Such analyses have shown that some systems are insensitive (e.g. Yi et al. 2000) whereas others are exquisitely sensitive (e.g. McAdams and Arkin 1999). Computational modeling is thus an extension of the scientific method (Phair and Misteli 2001; Fall et al. 2002; Slepchenko et al. 2002), providing the means to create precise, unambiguous, quantitative descriptions of biological phenomena that can be used to evaluate hypotheses systematically and to explore non-obvious dynamical behavior of a biological system (Hartwell et al. 1999; Csete and Doyle 2002; Endy and Brent 2001). For all of these reasons, the emphasis on developing and using models for quantitative predictions is one of the foundations of systems biology.

Life scientists sometimes object to the idea of modeling by arguing, “If you understand something so well that you can simulate it, why bother? You already understand it!” But, this argument misses the point of modeling. One does not begin creating models with understanding; indeed, the opposite is often the case—one begins with ignorance. It is the exercise of developing a model(s) that leads to understanding. Developing a computer model requires a greater degree of intellectual honesty than writing down an informal verbal model or drawing a block-and-arrow diagram. It is all too easy to imagine that one understands something, but it is quite another to make a computer model work.

Many sometimes feel that they cannot create a model because they do not have enough data. Here again we reiterate a basic premise of modeling that developing the model can be an extremely useful exercise for discovering what data are missing. “[The] complexity of biological systems makes it increasingly difficult to identify the next best experiment without such a tool” (Bower and Bolouri 2001b).

Finally, biologists often express the concern that computational modeling is a lot of work and that it requires an entirely different training than, e.g., “wet-bench” experimentation. Unfortunately, this is to a large extent still true. However, modern software tools can provide considerable assistance in developing, verifying, analyzing and sharing computational models (see Section 3).

2.2 Mechanistic models can serve as frameworks for organizing data and hypotheses

A spectrum of types of formal models exists (Gershenfeld 1998; Phair and Misteli 2001). On one end of the spectrum lie observational models: ones that characterize and quantify patterns in data, but in such a way that the elements and processes in the model are not directly related to the components of the underlying system. Curve-fitting models fall into this category. On the other end of the spectrum lie *mechanistic* models: ones in which the entities and processes correspond directly to hypothesized structures and processes in the biological system being modeled. While mechanistic models are much more difficult to develop, they are also more valuable for making predictions that can be related to empirical data.

Detailed, mechanistic models are designed to capture essential structural, biochemical, and genetic aspects of a biological system, staying faithful to chemical and physical laws. Many scientists have been developing such models for decades, and have long recognized the utility of computers for assisting with model simulation and analysis—in fact, the first simulations of biochemical reaction models were made before the advent of digital computers (Chance et al. 1940, 1952; Chance 1943, 1960; Garfinkel 1965). However, the power afforded by modern computers has made possible new levels of model detail and analysis.

What is sometimes lost in the excitement over the power of simulation and analysis is the value of computational models to serve as focal points for research in ways that databases of experimental data cannot. Mechanistic, computational models are specifically constructed to illuminate the functional implications of the data upon which they are built. A realistic computational model represents a modeler's understanding of the structure and function of part of a biological system. Models thus can serve not only as the point of entry for data; they can also serve as dynamic tools that can be used to understand its significance (Bailey 1998). As the number of researchers constructing realistic models continues to grow, and as the models become ever more sophisticated, they collectively represent a significant accumulation of knowledge about the structural and functional organization of the system. Moreover, the assimilation of new hypotheses and data into existing models can be done in a more systematic fashion because the additions must be fitted into the existing constructs using the same rules as for the models themselves. Computational models can thus be far more useful than just encapsulating one modeler's abstraction of a particular system: once properly constructed, the models become *a dynamic representation of our current state of understanding of a system* in a form that can facilitate communication between research groups and help to direct further experimental investigations.

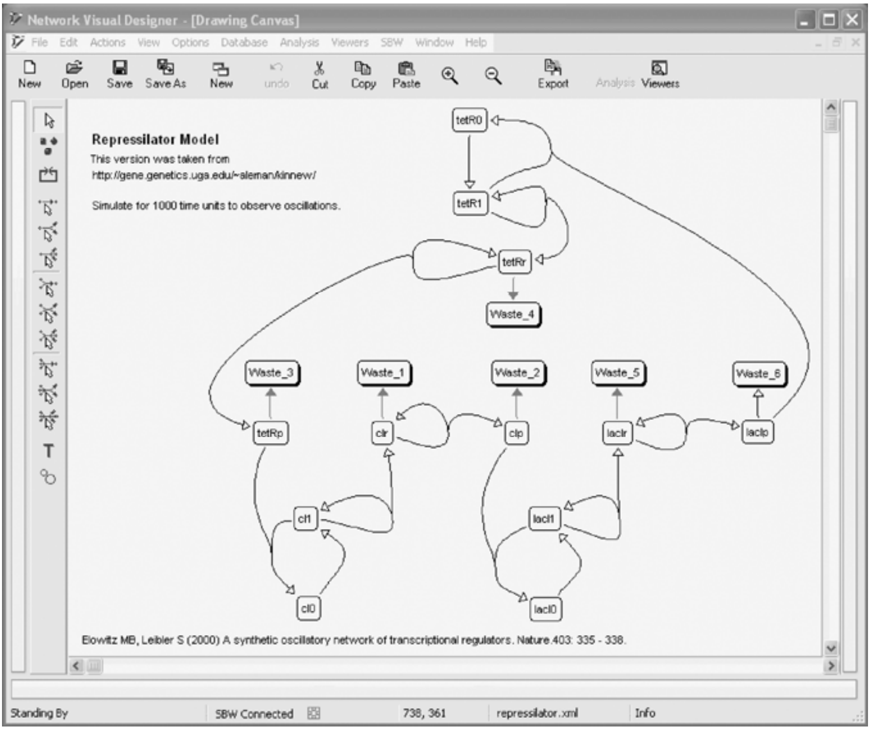
3 A variety of software resources are available today for computational modeling

One of the great advantages of modern software packages for biological modeling is that they allow users to avoid having to work with formal mathematics directly. Although one can certainly use a general-purpose mathematical package for developing and working with computational models, specialized tools can offer dedicated user interfaces and functionality for model development, simulation, and analysis, as well as other capabilities designed to simplify the work of biological modeling. In this section, we survey some of the capabilities provided by different tools as a way of informing prospective modelers of the choices available.

The user-interface paradigm used by a software system is one major dimension along which biological modeling tools can differ. The four most popular types of interfaces offered by modeling tools today are the following:

Diagrammatic: the tool enables users to express models visually by placing or drawing elements, structures, and relationships on a digital canvas. Often this takes the form of a graph resembling the block-and-arrow diagrams commonly presented by biologists as depictions of metabolic or signaling pathways. Additional quantitative information about the model is usually obtained from the user using a small number of fill-in-the-blank forms. Examples of tools implementing this kind of interface include JDesigner (Sauro et al. 2003; Sauro 2001),

Fig. 1 (overleaf). (Top) Screen image of JDesigner (Sauro 2001), a program that provides a graph-based interface allowing users to “draw” models. Nodes represent chemical species and arcs represent chemical reactions. Assignments of chemical rate laws to the arcs and chemical values for concentrations and other parameters are made using pop-up dialogue boxes. (Bottom) Screen image of the JigCell Model Builder (Vass et al. 2004), a program that provides a spreadsheet-style interface. Users input chemical equations, but different parts go into separate columns; moreover, the program performs consistency checking on the user’s input, helping to eliminate some common errors.

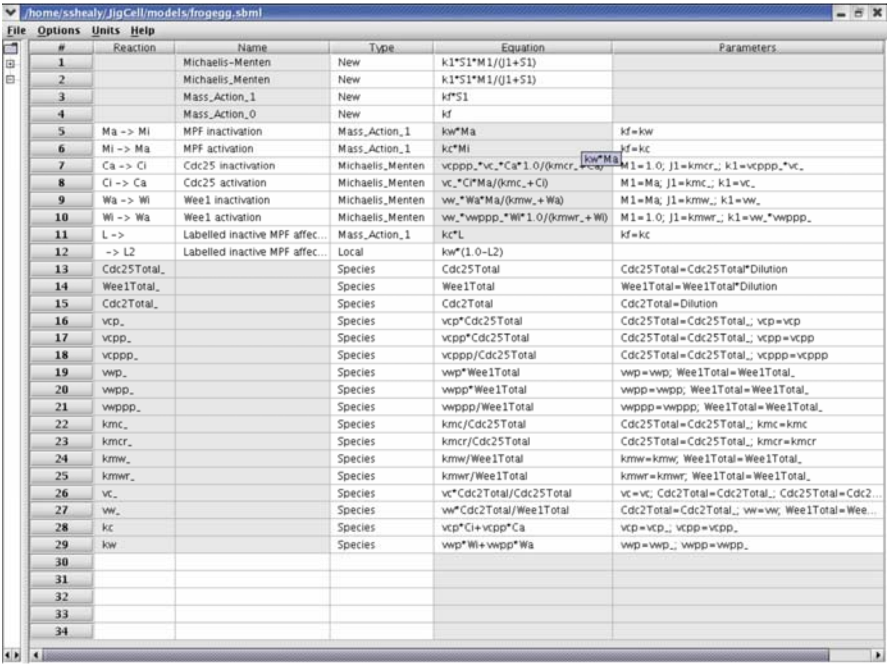

Network Visual Designer - [Drawing Canvas]
File Edit Actions View Options Database Analysis Viewers SBW Window Help
New Open Save Save As New undo Cut Copy Paste Export Analysis Viewers
tetR0
tetR1
tetRr
Waste_4
Waste_3
Waste_1
Waste_2
Waste_5
Waste_6
tetRp
clr
clp
laclr
laclp
cl1
lacl1
cl0
lacl0
Standing By
SBW Connected
738, 361
repressilator.xml
Info

Repressilator Model

This version was taken from
http://gene.genetics.uga.edu/~aleman/kinnew/

Simulate for 1000 time units to observe oscillations.

Elowitz MB, Leibler S (2000) A synthetic oscillatory network of transcriptional regulators. Nature 403: 335 - 338.



#	Reaction	Name	Type	Equation	Parameters
1		Michaelis-Menten	New	k1*S1*M1/(J1+S1)	
2		Michaelis_Menten	New	k1*S1*M1/(J1+S1)	
3		Mass_Action_1	New	kf*S1	
4		Mass_Action_0	New	kf	
5	Ma -> Mi	MPF inactivation	Mass_Action_1	kw*Ma	kf=kw
6	Mi -> Ma	MPF activation	Mass_Action_1	kc*Mi	kf=kc
7	Ca -> Ci	Cdc25 inactivation	Michaelis_Menten	vcppp_*vc_*Ca*1.0/(kmcr_+Ca)	M1=1.0; J1=kmcr_; k1=vcppp_*vc_
8	Ci -> Ca	Cdc25 activation	Michaelis_Menten	vc_*Ci*Ma/(kmc_+Ci)	M1=Ma; J1=kmc_; k1=vc_
9	Wa -> Wi	Wee1 inactivation	Michaelis_Menten	vw_*Wa*Ma/(kmw_+Wa)	M1=Ma; J1=kmw_; k1=vw_
10	Wi -> Wa	Wee1 activation	Michaelis_Menten	vw_*vwppp_*Wi*1.0/(kmwr_+Wi)	M1=1.0; J1=kmwr_; k1=vw_*vwppp_
11	L ->	Labelled inactive MPF affec...	Mass_Action_1	kc*L	kf=kc
12	-> L2	Labelled inactive MPF affec...	Local	kw*(1.0-L2)	
13	Cdc25Total_		Species	Cdc25Total	Cdc25Total=Cdc25Total*Dilution
14	Wee1Total_		Species	Wee1Total	Wee1Total=Wee1Total*Dilution
15	Cdc2Total_		Species	Cdc2Total	Cdc2Total=Dilution
16	vcp_		Species	vcp*Cdc25Total	Cdc25Total=Cdc25Total_; vcp=vcp
17	vcpp_		Species	vcpp*Cdc25Total	Cdc25Total=Cdc25Total_; vcpp=vcpp
18	vcppp_		Species	vcppp/Cdc25Total	Cdc25Total=Cdc25Total_; vcppp=vcppp
19	vwp_		Species	vwp*Wee1Total	vwp=vwp; Wee1Total=Wee1Total_
20	vwpp_		Species	vwpp*Wee1Total	vwpp=vwpp; Wee1Total=Wee1Total_
21	vwppp_		Species	vwppp/Wee1Total	vwppp=vwppp; Wee1Total=Wee1Total_
22	kmc_		Species	kmc/Cdc25Total	Cdc25Total=Cdc25Total_; kmc=kmc
23	kmcr_		Species	kmcr/Cdc25Total	Cdc25Total=Cdc25Total_; kmcr=kmcr
24	kmw_		Species	kmw/Wee1Total	kmw=kmw; Wee1Total=Wee1Total_
25	kmwr_		Species	kmwr/Wee1Total	kmwr=kmwr; Wee1Total=Wee1Total_
26	vc_		Species	vc*Cdc2Total/Cdc25Total	vc=vc; Cdc2Total=Cdc2Total_; Cdc25Total=Cdc2...
27	vw_		Species	vw*Cdc2Total/Wee1Total	Cdc2Total=Cdc2Total_; vw=vw; Wee1Total=Wee...
28	kc		Species	vcp*Ci+vcpp*Ca	vcp=vcp_; vcpp=vcpp_
29	kw		Species	vwp*Wi+vwpp*Wa	vwp=vwp_; vwpp=vwpp_
30					
31					
32					
33					
34					

CellDesigner (Funahashi et al. 2003; Kirkwood et al. 2003b), TERANODE Design Suite (Duncan et al. 2004; Teranode Inc. 2004), and the Virtual Cell (Schaff et al. 1997, 2001). The top half of Figure 1 shows an example screenshot from JDesigner.

Spreadsheet: the tool provides a multicolumn grid interface reminiscent of spreadsheet programs commonly offered in contemporary office productivity software suites. Information about reactions, species, and compartments typically are entered in separate spreadsheet areas, each having separate columns for different characteristics of the elements being entered. An example of a package providing this kind of interface is the JigCell Model Builder (Allen et al. 2003; Vass et al. 2004), a screenshot of which is shown in the bottom portion of Figure 1.

Forms-based: the tool prompts the user for information about a model using fill-in-the-blank forms or dialog boxes. An example of a tool implementing this kind of interface is COPASI (Mendes 2003) and its predecessor, Gepasi (Mendes 1993, 2001). Note that some tools take the information so gathered and display the resulting model using a diagram or a spreadsheet view but do not allow the user to edit the model directly using the diagram or spreadsheet, blurring the distinction somewhat.

Text-based: the tool enables users to define models using a formalized textual language and notation meant to be read and written by a human. Some of these languages mix constructs for defining models with directives for controlling simulations or other actions on the model. Some of the software packages provide a notation based on traditional chemical reaction style notation (e.g. A + B ↔ C), while others explore different notations. Examples of tools using this general user-interface paradigm include Cellerator (Shapiro et al. 2004b, 2003), Dizzy (Ramsey and Bolouri 2004), Jarnac (Sauro 2000b, 2000a), MathSBML (Shapiro 2004; Shapiro et al. 2004a), and WinSCAMP (Sauro and Fell 1991; Sauro et al. 2003).

Some packages provide more than one of these interface paradigms simultaneously, allowing users to switch between interface styles. An example in this category is TERANODE Design Suite.

Of course, the primary purpose of modeling tools is to allow users to perform analysis on the models created by users. Some software tools are dedicated model editors lacking built-in simulation and analysis capabilities; in these, users are expected to transfer the model to a separate analysis package. For this purpose, the Systems Biology Markup Language (SBML; see Section 4) is a popular model export format. Other tools provide built-in analysis capabilities.

In the context of simulation and analysis, software tools differ in the type of model representation *framework* they employ. The following are among the most popular types of frameworks in use today:

Logical: the tool converts the model description into a Boolean or extended logical representation (de Jong 2002). Certain classes of models, such as abstract models of regulatory networks, are more conveniently cast into this form than into, for example, differential-algebraic equations. An example of a tool in this category is NetBuilder (Brown et al. 2002; Schilstra and Bolouri 2002).

Ordinary differential equations (ODE): the tool converts the model description into a system of ordinary differential equations. This commonly involves one differential equation for each chemical species in the model. The ODE framework is the most popular one in use today for biochemical systems simulation. Representative examples include COPASI (Mendes 2003), Gepasi (Mendes 1993, 2001) and Jarnac (Sauro 2000b, 2000a).

Differential-algebraic equations (DAE): the tool converts the model into a system of ordinary differential equations with algebraic constraints. ODE representations are a popular framework, but complex models often include algebraic constraints and require the use of DAE representations. An example is a model that imposes constraints on species concentrations. The DAE framework subsumes the ODE framework. Because the DAE framework supports more of the constructs that modelers often want to express, it is a better match for modelers' needs. However, DAE solvers are more difficult to implement than ODE solvers, and fewer software packages provide full DAE support. An example of a tool that provides limited DAE support is Jarnac (Sauro 2000b, 2000a); an example of a tool providing a full DAE solver is MathSBML (Shapiro 2004; Shapiro et al. 2004a).

Partial differential equations (PDE): the tool converts the model into a system of partial differential equations. These arise when there is more than one independent variable in the system. For example, modeling spatial diffusion requires both time and space as independent variables. PDE solvers are much more difficult than ODE or DAE solvers to implement and use properly, which is why so few software tools use a PDE framework. One that does is the Virtual Cell (Schaff et al. 1997, 2001). (We note in passing that SBML does not currently have support to represent PDE-level models or chemical diffusion.)

Hybrid: the tool converts the model to a (continuous) differential equation framework that also supports time-dependent discontinuous events. Discontinuities can cause abrupt changes in the system of equations and the behavior of the system, and require specialised support in the model interpretation system. Hybrid modeling frameworks are necessary for properly handling such things as cell cycle models. Some packages providing hybrid simulators include E-Cell (Tomita et al. 1999; Tomita 2001), MathSBML (Shapiro 2004; Shapiro et al. 2004a), and TERANODE Design Suite (Duncan et al. 2004; Teranode Inc. 2004).

Stochastic: the tool casts the model as a set of discrete quantities (molecules or chemical species) and associated probabilities for interactions (reactions). Most such software uses the stochastic simulation algorithm by Gillespie (1977) or the Gibson-Bruck variant of Gillespie's algorithm (Gibson and Bruck 2000). Unlike differential-equation frameworks, stochastic frameworks do not approximate the model as a continuous, deterministic system. Instead, a stochastic framework treats the underlying biochemical reactions as random discrete processes in accordance with the chemical and physical properties of the component parts. In essence, stochastic frameworks more accurately represent true molecular interactions. However, the greater accuracy of stochastic frameworks comes at a high cost. Because the behavior of each chemical entity is individually modeled as a stochastic process, simulations are extremely demanding of computational resources. Some examples of systems implementing stochastic simulation capability include BASIS (Kirkwood et al. 2003a, 2003b), Dizzy (Ramsey and Bolouri

2004), E-Cell (Tomita et al. 1999; Tomita 2001), SigTran (DiValentin 2004) and StochSim (Morton-Firth and Bray 1998; Le Novere and Shimizu 2001).

Most of the packages discussed above are standalone applications (i.e. they can be installed and run locally on a computer), while a few are web-based, offering a service located on the Internet which users access remotely using a web browser. BASIS (Kirkwood et al. 2003a, 2003b) and the Virtual Cell (Schaff et al. 1997, 2001) are examples in the latter category.

A few software systems also provide database functionality. Some have an integrated database used to store models and model components in a form more organized than simply a collection of files. These systems sometimes also offer a means to share the database among different users. Examples in this category include Monod, TERANODE Design Suite (Duncan et al. 2004; Teranode Inc. 2004), and the Virtual Cell (Schaff et al. 1997, 2001). A few other systems provide a means to access third-party external repositories data, models or other information. An example in this category is E-Cell (Tomita et al. 1999; Tomita 2001).

Finally, most of the tools mentioned in this section are free for personal and/or educational use, although there may be costs for other users. Other packages, such as TERANODE Design Suite (Duncan et al. 2004; Teranode Inc. 2004), are commercial products.

4 Exchanging models between software tools: The Systems Biology Markup Language

To be useful as formal embodiments for understanding biological systems, computational models must be put into a format that can be communicated effectively between different software tools that work with them. The Systems Biology Markup Language (SBML) project is an effort to create a machine-readable format for representing computational models at the biochemical reaction level (Finney and Hucka 2003; Hucka et al. 2003). By supporting SBML as input and output formats, different software tools can operate on the identical representation of a model, removing chance for errors in translation and assuring a common starting point for analyses and simulations.

The SBML project is not an attempt to define a standard universal language for representing quantitative models; the fluid and rapidly evolving views of biological function, and the vigorous rate at which new computational techniques and individual tools are being developed today are incompatible with a one-size-fits-all concept of a universal language. Instead of trying to define how software tools should represent their models *internally*, the goal of the SBML project is to reach agreement on a format on how the tools communicate models *externally*. The SBML language allows software developers the freedom to explore different representations within their tools while still allowing some degree of interoperability between the tools. Such a format can serve as a *lingua franca* enabling communication of the most essential aspects of models between software systems in much the same way as “contact languages” first enabled human societies to communicate in the Mediterranean during the Middle Ages.

4.1 The general form of SBML

Although SBML models are intended to be read and written by software tools and not by humans, it is useful to overview the general characteristics of the representation in order to better understand how it organizes information about biological systems.

SBML is a machine-readable model definition language based upon XML, the eXtensible Markup Language (Bray et al. 2000; Bosak and Bray 1999), which is a simple and portable text-based substrate that has gained widespread acceptance in computational biology (Augen 2001; Achard et al. 2001). SBML can encode models consisting of biochemical entities (species) linked by reactions to form biochemical networks. An important principle in SBML is that models are decomposed into explicitly labeled constituent elements, the set of which resembles a verbose rendition of chemical reaction equations; the representation deliberately does not cast the model directly into a set of differential equations or other specific interpretations of the model. This decomposition makes it easier for a software tool to interpret the model and translate the SBML format into whatever internal form the tool actually uses.

SBML is being developed in levels, where each higher level adds richness to the model definitions that can be represented by the language. Level 2 is currently the highest level defined; it represents an incremental evolution of the language (Finney et al. 2003) resulting from the practical experiences of many users and developers, who have been working with Level 1 (Hucka et al. 2001, 2003). In SBML Level 2, the definition of a model consists of lists of one or more of the following components:

Compartment, a container of finite volume for homogeneously-mixed substances where reactions take place;

Species, a pool of a chemical substance located in a specific compartment, where this represents the concentration or amount of a substance and not a single molecule (example substances that form species are ions such as calcium and molecules such as ATP or DNA);

Reaction, a statement describing some transformation, transport or binding process that can change one or more species (each reaction is characterized by the stoichiometry of its products and reactants and optionally by a rate equation);

Parameter, a quantity that has a symbolic name, such as a frequently-used constant;

Unit definition, a name for a unit used in the expression of quantities in a model;

Rule, a mathematical expression that is added to the model equations constructed from the set of reactions (rules can be used to set parameter values, establish constraints between quantities, etc.);

Function, a named mathematical function that can be used in place of repeated expressions in rate equations and other formulas; and

Event, a mathematical formula evaluated at a specified moment in the time evolution of the system.

This simple formalism allows modeling of a wide range of biological phenomena, including cell signaling, metabolism, gene regulation, and others. Flexibility and power come from the ability to define arbitrary formulae for the rates of change of variables as well as the ability to express other constraints mathematically.

Many kinds of analyses can be applied to models in the elementary SBML format. The tools discussed in Section 3 are representative of the range of applications for which SBML is suitable.

4.2 The continued evolution of SBML

From its inception, SBML has been largely driven by practical needs of researchers interested in exchanging quantitative computational models between different software tools, databases, and other resources. The language reflects this, and in some respects exhibits the results of pragmatic choices more than elegant, top-down design. The development of SBML Level 2 benefited from two years of experience with SBML Level 1 by many modellers and software developers, and distils more effectively the fundamental needs of the biological network simulation community. It represents, in a concrete way, the consensus of a large segment of the modelling community about the intersection of features that should be possessed by a *lingua franca* for communicating models between today’s software tools.

SBML's popularity has led to the formation of an active community of researchers and software developers who are now working together to push SBML in new directions. As a language that is an intersection rather than a union of features needed by all tools, SBML currently cannot support all the representational capabilities that all software systems offer to users. Some packages offer features that have no explicit equivalent in SBML Level 2, and those tools currently can only store those features as annotations in an SBML model. Yet, in many cases, those features could potentially be used by more than one tool, and thus it would be appropriate to have some agreed-upon representation for them in SBML. Using Level 2 as a starting point, the SBML community has been developing proposals and prototype implementations of many new capabilities that will become part of SBML Level 3.

Because of the demand-driven, consensus-oriented approach to SBML evolution, the features currently in SBML and in development for SBML Level 3 are a reflection of the state of computational modeling today. The list of planned features thus serves to foreshadow what is to come in terms of modeling capabilities in the near future:

Composition: The biochemical network models being constructed by modelers are becoming increasingly large and complex. Structuring the models in a modular fashion is an essential approach to managing their complexity. Composition, as its name suggests, involves composing a model out of a set of instances of submodels. The resulting model structure is hierarchical; for example, a model of a cell might be composed from a model of a nucleus, multiple model mitochondria, and various other model structures. The E-CELL (Tomita et al. 1999, 2001) and ProMoT/DIVA (Stelling et al. 2001) systems are examples of simulation tools that support composing models out of submodels. The addition of a modular composition facility into SBML will bring several benefits. First, it will allow a component submodel to be reused multiple times within a single (larger) model. Second, it will allow the creation of libraries of model components. In time, the systems biology field will be able to develop standard, vetted submodels for commonly-needed components, and eventually, modelers will be able to compose models using high-level components taken from libraries rather than have to re-create every piece from scratch themselves. And third, it will enable modelers to incorporate several alternative submodels for a given model instance, in which each alternative could contain a representation at a different level of detail and/or use a representation that is appropriate for a particular type of simulation algorithm.

Multi-component species: SBML Levels 1 and 2 can represent models in which the chemical species are treated as simple, indivisible biochemical entities having only one possible state. However, this approach becomes untenable when modeling systems in which the species have many possible internal states or the species are composed from subcomponents (Goldstein et al. 2002). An example of this situation involves a protein that can be phosphorylated at multiple locations: the possible phosphorylation combinations lead to a combinatorial explosion of states of the protein. Although currently this can be represented in SBML Levels 1 and 2 by treating each state or combination of subcomponents as a separately named chemical species, this approach is an awkward and limited solution. To address this problem, another current area of SBML development is a representation scheme in which the subcomponents of chemical species are the smallest logical entities, rather than whole species being the entities. The research task is to define a representation scheme that is flexible enough to represent all the relevant biochemical phenomena while remaining computationally feasible for simulation and analysis.

Diagram Layout: Biochemical models are often visualized and edited using software in diagrammatic form. Examples of software that enables this include: JDesigner (Sauro 2003, 2001) and CellDesigner (Funahashi et al. 2003, 2004). The diagram layout that the user creates with these programs is especially useful for interpreting models created with this software. Another active area of SBML development is extending SBML so that diagram information can be added to models in a standard form.

Spatial geometry: The spatial distribution and diffusion of chemical species in space can be highly significant (Fink et al. 2000) and often needs to be represented in models. Not all software tools today support the use of spatial information, but it is likely that more will in the near future.

Alternative Mathematical Representations for Reactions: The current definition of SBML is somewhat biased towards on ODE-based representation of biochemical models. While it is possible to transform a subset of models encoded in this representation into a form acceptable to stochastic simulators, this, unfortunately, does not allow expression of the complete range of facilities that are available in stochastic simulators. Similarly, while it is possible to describe deterministic discrete events explicitly in SBML Level 2, it is not possible to define a reaction that operates in this way. Addressing these and other issues are included in development for SBML Level 3.

5 Development of an *E. coli* systems biology project

A wealth of information has been gained from reductionist biology over the past fifty years. Reductionism has been especially rewarding when directed towards understanding highly amenable systems. Studies of *E. coli* and its phages have given birth to early concepts of the fine structure of the gene, co-linearity of gene structure and protein sequence, molecular mechanisms of suppression, gene regulation, transposition, and many other phenomena. *E. coli* is now the source for much of our information on biochemistry, molecular biology, metabolic pathways, and regulation, and it continues to be a source for new insights into how cells work. *E. coli* has served as a model for understanding innumerable fundamental processes like the mechanisms of DNA replication (Kornberg and Baker 1992) and DNA repair (Chen et al. 2001), DNA transcription, gene repression and activation, protein synthesis, protein folding, protein targeting, macromolecular assembly, signal transduction, the catalytic nature of disulfide bond formation, cell division, the function of catalytic and small regulatory RNAs, and other processes.

The decision to focus early studies of cell physiology on *E. coli* has often been credited to a well-known phrase by Jacques Monod dating from 1954 “Anything found to be true of *E. coli* must also be true of elephants.” Early successes from *E. coli* research have also led many, most notably Sydney Brenner, to develop *E. coli*-like models for other processes (behavior, development, the immune response, multigene families, the nervous system, and many more). Many model organisms now exist for eukaryotic molecular biology (like yeast *Saccharomyces cerevisiae* and *S. pombe* and *Dictyostelium discoidium*), development and human disease (e.g. *Drosophila melanogaster*, *Caenorhabditis elegans*, *Fugu rubripes* (pufferfish), *Brachydanio rerio* (zebrafish), and human biology (*Mus musculus* (the laboratory mouse) and primates. *E. coli*-like models also exist for important processes in other bacteria (e.g. sporulation in *Bacillus subtilis*, cell division in *Caulobacter crescentus*, and development in *Myxococcus xanthus*) and for the Archae (e.g. *Haloferax volcani* and *Sulfolobus solfataricus*). Huge successes in pathogenic bacteriology have been more rapid for those bacteria most closely related to *E. coli*. The decision to create a *Shewanella* consortium for studying environmental bioremediation was based largely on its similarity with *E. coli*. Yet, no similar project exists for *E. coli* systems biology.

To meet this challenge, a small group of mostly *E. coli* biologists and modelers convened an informal workshop at the Intelligent Systems of Molecular Biology

Conference in Edmonton, Canada, in August 2002. Their meeting gave birth to the International *E. coli* Alliance (IECA), which was announced a few weeks later (Holden 2002). IECA was organized to help with the development of highly integrated and interdisciplinary research in bioinformatics, experimental, and modeling sciences that will be required to gain deeper understanding of cellular subsystems (gene regulatory, metabolic, and signaling networks), work that will contribute towards the development of a rudimentary whole cell model.

Subsequent meetings included discussions on how to organize a worldwide *E. coli* systems biology project. These were held in November 2002 at North Mymms, nearby London, UK, in February 2003 in San Diego, USA, and in March 2003 in Magdeburg, Germany. There was consensus that much work was needed. A standard strain would have to be selected, preferably based on data from rigorously controlled experiments. What kind? How many? Who would do them? New technologies would have to be developed. Metadata generated would be enormous. These data would need to be stored, disseminated, and modeled. We would need to reach agreements on data sharing and many other issues. Modelers were in a quandary about data formats and modeling languages, because modeling uses different kinds of data depending upon the approach, as described above. Committees were formed on strain and experimental standards, metabolic measurement and nomenclature, and modeling.

If our objective were modeling of *E. coli*, then experimentalists and modelers would need to work together from the start. This would require cooperation and collaborations among scientists with diverse interests and expertise. Experimentalists and modelers would need to be equally represented. To further promote *E. coli* systems biology research, the First IECA Conference was held in June 2003 at the Institute for Advanced Biosciences, Tsuruoka, Japan. The Second IECA Conference was held in June 2004 in Banff, Canada. More than one hundred international scientists have attended. Plans are now underway to hold the Third IECA Conference in September 2006 in Korea.

A major modeling problem is biological variability, even for experiments with the “same” strain by various investigators in the same or different laboratories. One way to overcome this hurdle would be to grow cells for modeling at a central location and to provide samples from standardized cultures to other researchers for an assortment of measurements. Predictive quantitative modeling is also often beyond the comprehension and belief of many biologists. Indeed, it is difficult to find examples in which modeling has given predictive outcomes where the results had not been known beforehand or could not have been inferred solely on the basis of prior experimental knowledge. It will be necessary to coordinate new experimentation with mathematical modeling as a means to validate or refute the predictive value of different modeling approaches for understanding new features of *E. coli* biology.

Foremost, a standard strain must be chosen that conforms as close to wild type as possible. This will probably be the *E. coli* K-12 sequenced strain MG1655 (Blattner et al. 1997). The finding of discrepancies for the “same” standard strain, e.g. Corbin et al. (2003), in different labs gives impetus to the concept for

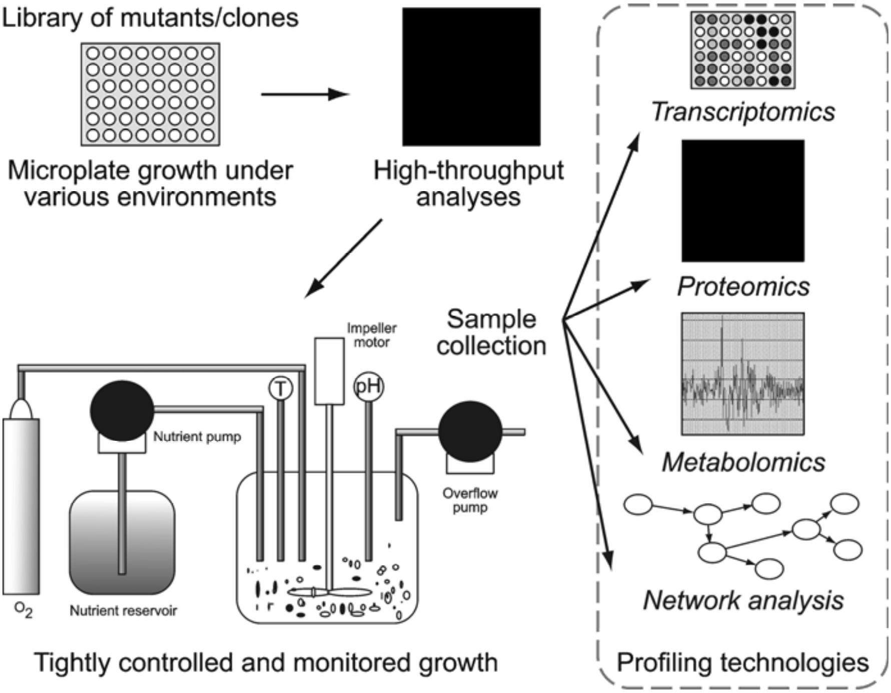


Fig. 2. Schematic of a centralized microbial growth facility. Normal *E. coli*, specific mutant *E. coli*, or *E. coli* cells identified from screening mutant libraries in microplates and characterized by high throughput techniques would be examined. Strains possessing an interesting phenotype would be selected for growth under standardized, rigorously controlled conditions. The fermentation would be continuous mode and samples would be collected and immediately frozen for further analysis by collaborators.

development of a standardized growth facility. To be sure, others had found discrepancies between east and west coast variants of *E. coli* K-12 AB1157 (Verma and Egan 1985). Comparisons of RNA polymerase sigma factor subunits of *E. coli* K-12 W3110 samples revealed multiple variants existed between labs in Japan (Jishage and Ishihama 1997).

Accordingly, a consortium may grow standard cells at a community microbial growth facility (MGF), collect samples, and distribute them to researchers with special expertise in conducting measurements. This should permit doing “community experiments” that capture the interest and expertise of many talented investigators in different fields, regardless of their affiliation with consortia. This should also foster an open data-sharing policy between members and rapid release to the entire scientific community. Numerous kinds of measurements (e.g. transcriptome, proteome, metabolome, and interactome analyses) require diverse expertise that seldom can be found at one location (Fig. 2). These new technologies are also rapidly evolving. Ideally, all measurements to develop, verify or refute quantitative models should be made on the same culture. Thus, a central source for generation of samples under rigorously optimized and standardized growth and harvesting procedures may be a key to success of a whole cell *E. coli* systems biology project.

6 An integrated *E. coli* database for community research and systems biology

One of the requirements of an *E. coli* systems biology project is the establishment of an information center where all data on *E. coli* and related cells are integrated. Several gene, protein, or function-specific *E. coli* databases now contain vast information on gene structure, metabolic pathways, gene regulation, protein function, and other processes, e.g., ASAP (Glasner et al. 2003), ColiBase (Chaudhuri et al. 2004), Colibri (Medigue et al. 1993), EcoCyc (Karp et al. 2002), EcoGene (Rudd 2000), Ecoli Genome (www.genome.wisc.edu), Genobase (http://ecoli.aist-nara.ac.jp/GB5/), GenProtEC (Serres et al. 2004), RegulonDB (Salgado et al. 2004), and others. Links to these and other databases can be found at www.EcoliCommunity.org. Yet, none of these is comprehensive and substantial gaps exist. Also, many contain redundant information that has often been acquired from other databases, sometimes without proper attribution. Considerable biological resources (e.g. mutants, clones, fusions, etc.) now exist for systematic, genome-wide studies of *E. coli* (Mori et al. 2000; Baba et al. 2005; Kang et al. 2004), however, access to information about them is often unavailable or hard to find.

Whole cell modeling will require the application of new systems approaches as well as continual reductionist experimentation of the *E. coli* cell, especially for processes that are still poorly understood. New computational and experimental resources are needed. These resources should support both the development of an *E. coli* systems biology project and enhancement of the highly successful biochemistry, biophysics, molecular biology, molecular genetics, and physiology research now being done by the *E. coli* community. One way to strengthen both community and consortia research would be to develop a federated *E. coli* database, which for the purpose of discussion we will call EcoliBase.

A model organism database such as the envisioned EcoliBase should contain all available information on *E. coli*, a repository of computational and modeling tools, database(s) of all experimental resources for studying *E. coli* and their availability, and a data warehouse for storage, manipulation, and analysis of diverse kinds of high-throughput data.

The development of a new experimental resources database would be of value to *E. coli* systems biology, as well as the *E. coli* community, including both experimentalists and computational scientists. This database should be an integral component of EcoliBase (Fig. 3). EcoliBase should be accessible via a web browser, so that researchers can easily view, retrieve, and exchange data. It should be designed so that it can be queried by typing or clicking on a scrollable genome map, as well as being accessible by modeling software. Various kinds of data

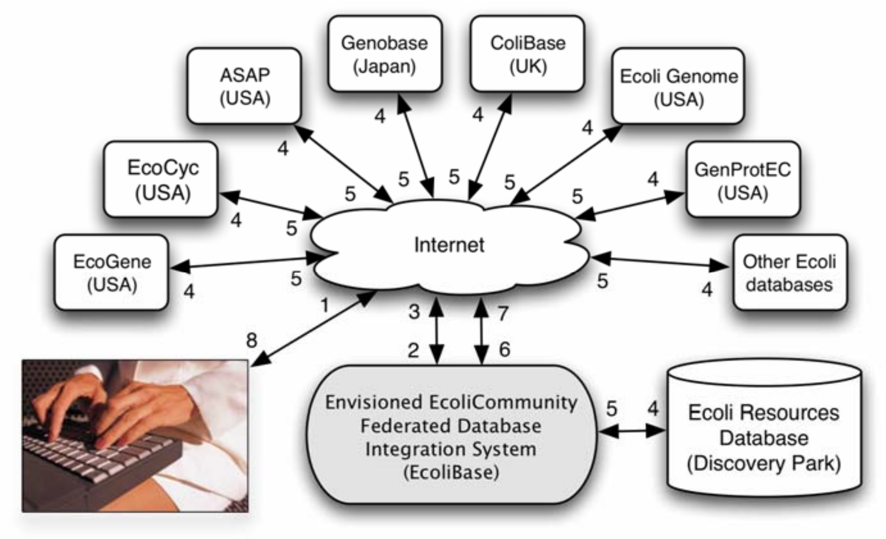


Fig. 3. Steps envisioned when a user submits a request to EcoliBase. A user would submit a request over the web to EcoliBase (Steps 1 and 2). EcoliBase would compile the request, decompose it into multiple sub-queries, and then submit the sub-queries to interoperating, participant databases including the *E. coli* Resources Database (Steps 3 and 4). Each participating database would evaluate the query and submit the answer back to EcoliBase (Steps 5 and 6). EcoliBase would compile and integrate the results, possibly addressing conflicts, and submit the compiled answer to the initiating user (Steps 7 and 8).

should be visualized for integration. An important aim of centralized databases is to set standards for the format of various data. Many kinds of data are essential to bring *E. coli* research to the goals of the next level, the foundation of systems biology and cell simulation of this organism. A few categories that would be stored in the database are discussed below.

The core and most important basis of a database is the list of parts determined by the genomic sequence, accurately annotated. A serious problem with current sequence annotation databases, including the new UniProt database (a combination of the ExPASy (Swiss-Prot), TrEMBL, and PIR databases, which were most commonly used) is that the source of annotation information is sometimes unclear. Since a new genome is usually annotated by homology searching against existing sequence databases, once wrong information is contaminated in the sequence database, the error can be propagated to another gene. Unfortunately, this error propagation is frequently observed in the current databases (Galperin and Koonin 1998; Gilks et al. 2002). In the database that we envision, the annotation of genes will clearly indicate the source of the information (history tracking), by indicating whether it is from experimental evidence or prediction by sequence similarity. In the former case, minimally a link pointing to the relevant literature should be added. In the latter case, a gene(s) with high sequence similarity to the gene of interest should be shown together with the score and homologous regions between the two to clarify where the predicted function originates so that one can trace the annotation history of genes to a set of genes with experimental evidence. To implement this annotation chain management system, we would first need to select only proteins annotated by experimental evidence from the UniProt database then repeat the sequence annotation procedure again. Once a new experiment provides new function information of a gene, the updated information would be passed to "downstream" genes that are annotated from the gene by tracking the annotation chain. These clean annotation data would be valuable not only for the *E. coli* research community but also for all bioinformatics research dealing with gene function.

Any cis-acting regulatory sites associated with a sequence, the boundaries of protein coding and structural RNA genes would also be included in the annotation with information as to how they were determined. Also included should be all other sequence features within the genome: replication origins, repeat elements, non-coding and structural RNAs, prophages (as intact elements as well as component parts). Transcriptional and non-transcriptional regulatory information, at the level of the gene, operon, regulon, and other regulatory circuits would also be described together with experimental information. Non-transcriptional regulation would include allostery and feedback inhibition, translational regulation, modifications, and protein degradation.

Annotations should aim to describe what is known about the gene and encoded protein, as well as any known interactions with other functions, defined genetically, biochemically, and by regulatory patterns. Some of this information would appear in other forms in other parts of the set of interoperating databases, but cross-annotation to the particular gene would be important as well.

As high throughput experiments continue to accumulate, an accessible and searchable repository for these data would be critical for allowing researchers to make correlations and do preliminary tests of hypotheses. These data would be deposited from collaborators around the world. In all cases, clear indications of the strains and growth conditions used and how the data were collected would need to be available, to allow the user to have sense of the reliability of the data. In many cases, the information would be linked to publications describing it. It is expected that this category of information would grow at the greatest rate and thus would require attention to simplify the access to new data as it becomes available, including the discussion of possible templates for experimental protocols and data analysis to allow comparisons.

Many groups have undertaken computational methods for predicting not only genes and the families of the predicted proteins, but sites, non-coding RNAs and secondary structure elements such as terminators. Such studies, with information about the nature of the predictions, would provide investigators with the ability to incorporate these predictions into their work. Combined with some large-scale experimental data, these analyses will give a system-wide view of the organism. Several groups have already begun collaborations to identify all probable regulatory motifs in the *E. coli* genome by using a variety of approaches. These would be shown with a confidence score in the database. Predicted protein tertiary structure (Kihara and Skolnick 2004) and protein localization would be included.

Table 1. Features of an envisioned *E. coli* federated database integrated system.

EcoliBase Integrated Tools	EcoliBase Federated Database Engine
Data-mining	Data importing/exporting
Bioinformatics & statistics	Metadata/version management
Microarray analysis	Schema mapping, evolution & integration
Data visualization	Multi-DB query translation & integration
	Access control & backup
	Annotation management
	Federated DBMS engine
	Web manager and user interfaces

Comparative information allows one to leverage the whole genome information available for *E. coli* to the understanding of other organisms. Orthologous and paralogous genes in other organisms would be listed from an *E. coli* gene. BLAST/FASTA methods, inference of phylogenetic trees and studies of within-species variability are powerful methods of DNA and protein sequence analysis that allow predicting functions of genes and proteins based upon experimentally determined functions in *E. coli* and tracing the evolutionary transformations of functions (gene duplications, genome organization, pseudogenes, etc.). Bioinformatics analyses would be made to the other organisms to allow a comparative study.

Standard sequence analysis tools, such as homology search, motif search, protein secondary structure prediction, should be available by simple manipulation from each gene. Experimental data, such as microarray data, would be linked to analysis tools so that it can be analyzed instantly and in a standard way. Some pathway simulators (Mendes and Kell 2001; Shapiro et al. 2003; Takahashi et al. 2003) would be made available on the web, or if not, at least downloadable.

Not only public domain databases, such as PDB (protein structure), UniProt (proteins), PROSITE (motifs), EcoCyc and KEGG (pathways), but also other existing *E. coli* databases would be integrated as much as possible by collaboration. We would need a unified *E. coli* database (EcoliBase) that would be designed for interoperability so that it can be linked transparently to a larger database structure in the future. We envision EcoliBase to be a web-based interoperable federated database integration system.

All interoperating participant *E. coli* databases including the *E. coli* Resources Database would be registered with EcoliBase. Users would have a web interface to access EcoliBase (Fig. 3). A user may issue a request to EcoliBase via the web. EcoliBase translates and decomposes the submitted requests into sub-queries, then submits the sub-queries to the corresponding and interoperating participant databases. The results of each sub-query are integrated inside EcoliBase and are returned to the user. It is expected that EcoliBase will have the functional components shown in Table 1.

7 Putting models to work: The International *E. coli* Alliance

From the dawn of modern biology, the intestinal bacterium *E. coli* has been the most intensively studied organism. Many basic molecular events, best understood in *E. coli*, are universal throughout the natural world. *E. coli* has laid so many of the golden eggs of basic biochemistry, genetics, and molecular biology that no doubt it will lay even more. Our present day level of basic understanding of natural phenomenon far exceeds the imagination of even the most creative scientists a few decades ago. New tools for gaining even more biological information ensure future revelations will continue to be uncovered at an ever-increasing pace. Although creating a truly virtual cell may be far in the future, the place to start is with a well understood system for which there are tools for deepening our knowledge. Systems biology approaches are needed for conceptualizing and testing our interpretations of these data.

It was with these concepts in mind that IECA was formed as a worldwide alliance for the purpose of constructing a large-scale model of a simple, self-replicating cell. Bringing such a dream to fruition requires not only computational and experimental tools, but also changes in how we do science – the human factor. Many impediments must be overcome. Large-scale experimentation is new to biologists. Other fields of science, most notably areas of physics requiring huge and expensive resources, have dealt with issues now facing systems biology. Much more time is spent planning and designing major experiments in physics than seems to be the norm in systems biology. As in many present day physics projects, systems biology projects of the future will depend more and more on large numbers of researchers working together in distantly located teams. How to achieve this through collaboration and building consortia will be challenging. Funding agencies must also find creative ways of encouraging scientists with diverse expertise to work together in teams to reach a common goal.

Like the physicists' goal for a complete understanding of the world from the inner workings of an atom to the motion and expansion of the universe, the goal of IECA is the complete modeling of a whole cell. Perhaps, modeling a cell is itself a bit too ambitious. However, the time to start is now. A practical way to do this would be to begin by studying modules, like regulatory systems or metabolic or signaling pathways, then to build these into networks that can then be joined together at an ever higher level. Surely, a computerized *E. coli* virtual cell will add powerful new tools to our existing arsenal of discovery, including virtual experimentation and mathematical simulation. These biological and computational tools promise to be useful for everything from drug discovery to bioengineering.

Acknowledgements

Authors are supported by the National Institutes of Health GM62662 (B.L.W.) and GM70923 (A.F. and M.H.).

References

Achard F, Vaysseix G, Barillot E (2001) XML, bioinformatics and data integration. Bioinformatics 17:115-125

Allen NN, Calzone L, Chen KC, Ciliberto A, Ramakrishnan N, Shaffer CA, Sible JC, Tyson JJ, Vass MT, Watson LT, Zwolak JW (2003) Modeling regulatory networks at Virginia Tech. OMICS 7:285-299

Alm E, Arkin AP (2003) Biological networks. Curr Opin Struct Biol 13:193-202

Arkin AP (2001) Simulac and deduce. http://gobi.lbl.gov/~aparkin/Stuff/Software.html.

Augen J (2001) Information technology to the rescue! Nat Biotechnol 19:BE39-BE40

Baba T, Ara T, Okumura Y, Hasegawa M, Takai Y, Baba M, Oshima T, Datsenko KA, Tomita M, Wanner BL, Mori H (2005) Systematic construction of single gene deletions mutants in *Escherichia coli* K-12, submitted

Bailey JE (1998) Mathematical modeling and analysis in biochemical engineering: Past accomplishments and future opportunities. Biotechnol Prog 14:8-20

Bialek W, Botstein D (2004) Introductory science and mathematics education for 21st-century biologists. Science 303:788-790

Blattner FR, Plunkett G III, Bloch CA, Perna NT, Burland V, Riley M, Collado-Vides J, Glasner JD, Rode CK, Mayhew GF, Gregor J, Davis NW, Kirkpatrick HA, Goeden MA, Rose DJ, Mau B, Shao Y (1997) The complete genome sequence of *Escherichia coli* K-12. Science 277:1453-1462

Bosak J, Bray T (1999) XML and the second-generation web. Sci Am May

Bower JM, Bolouri H (2001a) Computational modeling of genetic and biochemical networks. MIT Press, Cambridge, Mass

Bower JM, Bolouri H (2001b) Introduction: understanding living systems. In: Bower, James M and Bolouri H (eds) Computational modeling of genetic and biochemical networks. MIT Press, Cambridge, Mass., p xiii-xx

Bray T, Paoli J, Sperberg-McQueen CM, Maler E (2000) Extensible markup language (XML) 1.0 Second Edition: http://www.w3.org/TR/1998/REC-xml-19980210

Brown CT, Rust AG, Clarke PJC, Pan Z, Schilstra MJ, De Buysscher T, Griffin G, Wold BJ, Cameron RA, Davidson EH, Bolouri H (2002) New computational approaches for analysis of cis-regulatory networks. Dev Biol 246:86-102

Burns JA (1971) Studies on complex enzyme systems. University of Edinburgh

Butler D (1999) Computing 2010: from black holes to biology. Nature 402:C67-C70

Capra F (1996) The Web of Life: A new scientific understanding of living systems. Anchor Books, New York

Chance B (1960) Analogue and digital representations of enzyme kinetics. J Biol Chem 235:2440-2443

Chance B (1943) The kinetics of the enzyme-substrate compound of peroxidase. J Biol Chem 151:553-577

Chance B, Brainerd JG, Cajori FA, Millikan GA (1940) The kinetics of the enzyme-substrate compound of peroxidase and their relation to the Michaelis theory. Science 92:455

Chance B, Greenstein DS, Higgins J, Yang CC (1952) The mechanism of catalase action. II. Electric analog computer studies. Arch Biochem Biophys 37:322-339

Chaudhuri RR, Khan AM, Pallen MJ (2004) coliBASE: an online database for *Escherichia coli*, *Shigella* and *Salmonella* comparative genomics;

http://colibase.bham.ac.uk/about/index.cgi?help=about&frame=genomechoose. Nucleic Acids Res 32:D296-D299

Chen S, Bigner SH, Modrich P (2001) High rate of CAD gene amplification in human cells deficient in MLH1 or MSH6. Proc Natl Acad Sci USA 98:13802-13807

Corbin RW, Paliy O, Yang F, Shabanowitz J, Platt M, Lyons CE Jr, Root K, McAuliffe J, Jordan MI, Kustu S, Soupene E, Hunt DF (2003) Toward a protein profile of *Escherichia coli*: comparison to its transcription profile. Proc Natl Acad Sci USA 100:9232-9237

Crick FHC (1973) Project K: "The complete solution of *E. coli*". Perspect Biol Med 67-70

Csete ME, Doyle JC (2002) Reverse engineering of biological complexity. Science 295:1664-1669

de Jong H (2002) Modeling and simulation of genetic regulatory systems: a literature review. J Computat Biol 9:67-103

DiValentin, P SigTran (2004) http://csi.washington.edu/teams/modeling/projects/sigtran/

Duncan J, Arnstein L, Li Z (2004) Teranode corporation launches first industrial-strength research design tools for the life sciences at DEMO: http://www.teranode.com/about/pr_2004021601.php

Endy D, Brent R (2001) Modelling cellular behaviour. Nature Suppl 409:391-395

Fall C, Marland ES, Wagner JM, Tyson JJ (2002) Computational cell biology. Springer-Verlag, New York

Fink CC, Slepchenko B, Moraru II, Watras J, Schaff JC, Loew LM (2000) An image-based model of calcium waves in differentiated neuroblastoma cells. Biophys J 79:163-83

Finney A, Hucka M, Sauro H, Bolouri H, Funahashi A, Bornstein B, Kovitz B, Matthews J, Shapiro BE, Keating S, Doyle J, Kitano H (2003) The systems biology workbench (SBW) Version 1.0: Framework and modules. Hawaii, USA. Pacific symposium on biocomputing 2003

Finney AM, Hucka M (2003) Systems Biology Markup Language: Level 2 and beyond. Biochem Soc Trans 31:1472-1473

Fraser SE, Harland RM (2000) The molecular metamorphosis of experimental embryology. Cell 100:41-55

Funahashi A, Tanimura N, Morohashi M, Kitano H (2003) CellDesigner: a process diagram editor for gene-regulatory and biochemical networks. BioSilico 1:159-162

Funahashi A, Tanimura N, Morohashi M, Kitano H (2004) CellDesigner; http://www.systems-biology.org/002/

Galperin MY, Koonin EV (1998) Sources of systematic error in functional annotation of genomes: domain rearrangement, non-orthologous gene displacement and operon disruption. In Silico Biol 1:55-67

Garfinkel D (1965) Simulation of biochemical systems. In: Stacy, Ralph W and Waxman, BD (eds) Computers in biomedical research. Academic Press, New York, pp 111-134

Gershenfeld NA (1998) The nature of mathematical modeling. Cambridge University Press, Cambridge

Gilks WR, Audit B, De Angelis D, Tsoka S, Ouzounis CA (2002) Modeling the percolation of annotation errors in a database of protein sequences. Bioinformatics 18:1641-9

Gillespie DT (1977) Exact stochastic simulation of coupled chemical-reactions. J Phys Chem 81:2340-2361

Gillespie DT, Petzold LR (2003) Improved leap-size selection for accelerated stochastic simulation. J Chem Phys 119:8229-8234

Glasner JD, Liss P, Plunkett G III, Darling A, Prasad T, Rusch M, Byrnes A, Gilson M, Biehl B, Blattner FR, Perna NT (2003) ASAP, a systematic annotation package for community analysis of genomes; https://asap.ahabs.wisc.edu/annotation/php/home.php?formSubmitReturn=1. Nucleic Acids Res 31:147-151

Goldstein B, Faeber JR, Hlavacek WS, Blinov ML, Redondo A, Wolfsy C (2002) Modeling the early signaling events mediated by FceRI. Mol Immunol137:1-7

Hartwell LH, Hopfield JJ, Leibler S, Murray AW (1999) From molecular to modular cell biology. Nature 402:C47-C52

Holden C (2002) Cell biology: Alliance launched to model *E. coli*. Science 297:1459-1460

Hood L (1998) Systems biology: New opportunities arising from genomics, proteomics, and beyond. Exp Hematol 26:681

Hucka M, Finney A, Sauro HM, Bolouri H (2001) Systems Biology Markup Language (SBML) Level 1: Structures and facilities for basic model definitions; http://www.sbml.org/

Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H, Arkin AP, Bornstein BJ, Bray D, Cornish-Bowden A, Cuellar AA, Dronov S, Gilles ED, Ginkel M, Gor V, Goryanin II, Hedley WJ, Hodgman TC, Hofmeyr JH, Hunter PJ, Juty NS, Kasberger JL, Kremling A, Kummer U, Le Novere N, Loew LM, Lucio D, Mendes P, Minch E, Mjolsness ED, Nakayama Y, Nelson MR, Nielsen PF, Sakurada T, Schaff JC, Shapiro BE, Shimizu TS, Spence HD, Stelling J, Takahashi K, Tomita M, Wagner J, Wang J (2003) The Systems Biology Markup Language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 19:524-531

Ideker T, Galitski T, Hood L (2001) A new approach to decoding life: systems biology. Annu Rev Genomics Hum Genet 2:343-372

Jishage M, Ishihama A (1997) Variation in RNA polymerase sigma subunit composition within different stocks of *Escherichia coli* W3110. J Bacteriol 179:959-963

Kacser H (1957) Appendix: Some physico-chemical aspects of biological organisation. In: Waddington CH (ed) The strategy of the genes: A discussion of some aspects of theoretical biology. George Allen and Unwin Ltd, London, pp 191-249

Kacser H, Burns JA (1967) Causality, complexity and computers. In: Locker A (ed) Quantitative biology of metabolism. Springer-Verlag, New York, NY, pp 11-23

Kang Y, Durfee T, Glasner JD, Qiu Y, Frisch D, Winterberg KM, Blattner FR (2004) Systematic mutagenesis of the *Escherichia coli* genome. J Bacteriol 186:4921-4930

Karp PD, Riley M, Saier M, Paulsen IT, Collado-Vides J, Paley SM, Pellegrini-Toole A, Bonavides C, Gama-Castro S (2002) The EcoCyc database; http://ecocyc.org/. Nucleic Acids Res 30:56-58

Kihara D, Skolnick J (2004) Microbial genomes have over 72% structure assignment by the threading algorithm PROSPECTOR_Q. Proteins 55:464-473

Kirkwood TBL, Boys R, Wilkinson D, Gillespie C, Proctor C, Hanley D (2003a) BASIS; http://www.basis.ncl.ac.uk/. 3-19-2004a

Kirkwood TBL, Boys RJ, Gillespie CS, Proctor CJ, Shanley DP, Wilkinson DJ (2003b) Towards an e-biology of ageing: integrating theory and data. Nature Reviews Molecular Cell Biology 4:243-249

Kitano H (2002) Computational systems biology. Nature 420:206-210

Kitano H (2001) Foundations of systems biology. MIT Press, Cambridge, MA

Kornberg A (2003) Ten commandments of enzymology, amended. Trends Biochem Sci 28:515-517

Kornberg A, Baker TA (1992) DNA replication. WH Freeman and Company, San Francisco, California

Le Novere N, Shimizu TS (2001) STOCHSIM: modelling of stochastic biomolecular processes. Bioinformatics 17:575-576

May RM (2004) Uses and abuses of mathematics in biology. Science 303:790-793

McAdams HH, Arkin A (1999) It's a noisy business! Genetic regulation at the nanomolar scale. Trends Genet 15:65-69

Medigue C, Viari A, Henaut A, Danchin A (1993) Colibri: a functional data base for the *Escherichia coli* genome. Microbiol Rev 57:623-654

Mendes P (2001) Gepasi 3.21; http://www.gepasi.org

Mendes P (2003) COPASI: Complex pathway simulator; http://mendes.vbi.vt.edu/tiki-index.php?page=COPASI

Mendes P (1993) Gepasi - a software package for modeling the dynamics, steady-states and control of biochemical and other systems. Comput Appl Biosci 9:563-571

Mendes P, Kell DB (2001) MEG (Model Extender for Gepasi): a program for the modelling of complex, heterogeneous, cellular systems. Bioinformatics 17:288-289

Mesarovic MD (1968) Systems theory and biology. Proceedings of the 3rd Systems Symposium at Case Institute of Technology. Springer-Verlag, Berlin, New York

Mori H, Isono K, Horiuchi T, Miki T (2000) Functional genomics of *Escherichia coli* in Japan. Res Microbiol 151:121-128

Morton-Firth CJ, Bray D (1998) Predicting temporal fluctuations in an intracellular signalling pathway. J Theor Biol 192:117-128

Noble D (2002) The rise of computational biology. Nat Rev Mol Cell Biol 3:460-463

Phair RD, Misteli T (2001) Kinetic modelling approaches to *in vivo* imaging. Nat Rev Mol Cell Biol 2:898-907

Ramsey S, Bolouri H (2004) Dizzy; http://labs.systemsbiology.net/bolouri/software/Dizzy/

Rudd KE (2000) EcoGene: a genome sequence database for *Escherichia coli* K-12. Nucleic Acids Res 28:60-64

Salgado H, Gama-Castro S, Martinez-Antonio A, Diaz-Peredo E, Sanchez-Solano F, Peralta-Gil M, Garcia-Alonso D, Jimenez-Jacinto V, Santos-Zavaleta A, Bonavides-Martinez C, Collado-Vides J (2004) RegulonDB (version 4.0): transcriptional regulation, operon organization and growth conditions in *Escherichia coli* K-12; http://www.cifn.unam.mx/Computational_Genomics/regulondb/. Nucleic Acids Res 32:D303-D306

Sauro HM (2003) WinScamp; http://www.cds.caltech.edu/~hsauro/Scamp/scamp.htm

Sauro HM (2000b) Jarnac; http://www.cds.caltech.edu/~hsauro

Sauro HM (2000a) Jarnac: A system for interactive metabolic analysis. Snoep JL, Hofmeyr JH, and Roywer JM; Animating the Cellular Map: Proceedings of the 9th International Meeting on BioThermoKinetics. Stellenbosch University Press

Sauro HM, Fell DA (1991) SCAMP: A metabolic simulator and control analysis program. Mathl Comput Modelling 15:15-28

Sauro HM, Hucka M, Finney A, Wellock C, Bolouri H, Doyle J, Kitano H (2003) Next generation simulation tools: The systems biology workbench and BioSPICE integration. OMICS 7:355-372

Sauro HS (2001) JDesigner: A simple biochemical network designer; http://members.tripod.co.uk/sauro/biotech.htm

Savageau MA (1969) Biochemical systems analysis.1. Some mathematical properties of rate law for component enzymatic reactions. J Theor Biol 25:365-366

Savageau MA (1970) Biochemical systems analysis .3. Dynamic solutions using a power-law approximation. J Theor Biol 26:215

Schaff J, Fink CC, Slepchenko B, Carson JH, Loew LM (1997) A general computational framework for modeling cellular structure and function. Biophys J 73:1135-1146

Schaff J, Slepchenko B, Morgan F, Wagner J, Resasco D, Shin D, Choi YS, Loew L, Carson J, Cowan A, Moraru I, Watras J, Teraski M, Fink C (2001) Virtual Cell; http://www.nrcam.uchc.edu

Schilstra M, Bolouri H (2002) NetBuilder; http://strc.herts.ac.uk/bio/maria/NetBuilder/index.html

Serres MH, Goswami S, Riley M (2004) GenProtEC: an updated and improved analysis of functions of *Escherichia coli* K-12 proteins; http://www.genprotec.mbl.edu/. Nucleic Acids Res 32:D300-D302

Shapiro BE (2004) MathSBML; http://sbml.org/mathsbml.html

Shapiro BE, Hucka M, Finney A, Doyle JC (2004a) MathSBML: A package for manipulating SBML-based biological models. Bioinformatics 20:2829-2831

Shapiro BE, Levchenko A, Meyerowitz EM, Wold BJ, Mjolsness ED (2003) Cellerator: extending a computer algebra system to include biochemical arrows for signal transduction simulations. Bioinformatics 19:677-678

Shapiro BE, Mjolsness E, Levchenko A (2004b) Cellerator; http://www-aig.jpl.nasa.gov/public/mls/cellerator/

Slepchenko BM, Schaff JC, Carson JH, Loew LM (2002) Computational cell biology: Spatiotemporal simulation of cellular events. Annu Rev Biophys Biomol Struct 31:423-441

Stelling J, Kremling A, Ginkel M, Bettenbrock K, Gilles E (2001) Towards a virtual biological laboratory. In: Kitano H (ed) Foundations of systems biology. MIT Press, Cambridge, MA, pp 189-212

Takahashi K, Ishikawa N, Sadamoto Y, Sasamoto H, Ohta S, Shiozawa A, Miyoshi F, Naito Y, Nakayama Y, Tomita M (2003) E-Cell 2: Multi-platform E-Cell simulation system. Bioinformatics 19:1727-1729

Teranode Inc. (2004)VLX Design Suite

Tomita M (2001) Towards computer aided design (CAD) of useful microorganisms. Bioinformatics 17:1091-1092

Tomita M, Hashimoto K, Takahashi K, Shimizu TS, Matsuzaki Y, Miyoshi F, Saito K, Tanida S, Yugi K, Venter JC, Hutchison CA, III (1999) E-CELL: software environment for whole-cell simulation. Bioinformatics 15:72-84

Tomita M, Nakayama Y, Naito Y, Shimizu T, Hashimoto K, Takahashi K, Matsuzaki Y, Yugi K, Miyoshi F, Saito Y, Kuroki A, Ishida T, Iwata T, Yoneda M, Kita M, Yamada Y, Wang E, Seno S, Okayama M, Kinoshita A, Fujita Y, Matsuo R, Yanagihara T, Watari D, Ishinabe S, Miyamoto S (2001) E-CELL; http://www.e-cell.org/

Tyson JJ, Chen K, Novak B (2001) Network dynamics and cell physiology. Nat Rev Mol Cell Biol 2:908-916

Vass M, Shaffer CA, Tyson JJ, Ramakrishnan N, Watson LT (2004) The JigCell model builder: a tool for modeling intra-cellular regulatory networks. Bioinformatics 20:3680-3681

Verma M, Egan JB (1985) Phenotypic variations in strain AB1157 cultivars of *Escherichia coli* from different sources. J Bacteriol 164:1381-1382

Wiener N (1961) Cybernetics; or, control and communication in the animal and the machine, 2nd edn. MIT Press, New York

Yi TM, Huang Y, Simon MI, Doyle J (2000) Robust perfect adaptation in bacterial chemotaxis through integral feedback control. Proc Natl Acad Sci USA 97:4649-4653

Zerhouni E (2003) The NIH roadmap. Science 302:63-64

Wanner, Barry L.
Department of Biological Sciences, Purdue University, West Lafayette, IN 47907-2054 USA
blwanner@purdue.edu

Andrew Finney
Science and Technology Research Institute, University of Hertfordshire, Hatfield, AL10 9AB, UK

Michael Hucka
Control and Dynamical Systems, California Institute of Technology, Pasadena, CA 91125-8100 USA